

# **SISTEMA DE RECOMENDACIONES BASADO EN CONTEXTO PARA ENTORNOS DE PUBLICIDAD UBICUA**



Trabajo de Grado  
Maestría en Ingeniería Telemática

**Víctor Garzón Marín**

Director: PhD. Ing. Gustavo Adolfo Ramírez González  
Asesor: PhD. Ing. Francisco Orlando Martínez Pabón

**Universidad del Cauca**  
**Facultad de Ingeniería Electrónica y Telecomunicaciones**  
**Departamento de Telemática**  
**Línea de Investigación en Aplicaciones sobre Internet**  
**Popayán, 2016**



# Agradecimientos

*Al PhD. Gustavo Ramírez, director,  
por estar siempre al tanto,  
su valioso tiempo dedicado y exigir que se entregue lo mejor para obtener excelentes resultados.*

*Al PhD. Francisco Martínez, asesor y autor original de SMARTA,  
base inicial de este trabajo.  
Gracias por alentarme a iniciar esta fase y estar allí para dar siempre su opinión y consejo.*

*A mis padres y hermanos, base fundamental en mi formación.  
Gracias por estar siempre allí,  
por querer escucharme aún cuando no me comprendieran,  
por exigir siempre más y ser parte de este logro.*

*A Ángela, por su apoyo incondicional,  
ser compañía de traspasos y madrugadas,  
alentarme a seguir aún cuando todo parecía estar mal.  
Gracias por ser un motivo para continuar  
y por comprender cuando el tiempo parecía no ser suficiente.*

*A la corporación Cluster CREATIC,  
por apoyar el talento TI del departamento.*

*A todas las personas que sacaron unos minutos de su tiempo para ser voluntarios en los experimentos  
desarrollados durante este proyecto,  
su información fue valiosa para conseguir los resultados.*

*Gracias!*



# CONTENIDO

	<b>Pág.</b>
<b>1 Introducción</b>	<b>1</b>
1.1 Escenario de motivación . . . . .	2
1.2 Planteamiento del problema . . . . .	3
1.3 Objetivos . . . . .	4
1.3.1 Objetivo general . . . . .	4
1.3.2 Objetivos específicos . . . . .	5
1.4 Actividades . . . . .	5
1.4.1 Generación de la base inicial de conocimiento . . . . .	5
1.4.2 Caracterización del Contexto . . . . .	5
1.4.3 Diseño del Sistema de Recomendaciones . . . . .	6
1.4.4 Construcción del piloto de pruebas y evaluación del Sistema de Recomendaciones . . . . .	6
1.4.5 Publicación . . . . .	6
1.5 Contribuciones . . . . .	6
1.5.1 Mecanismos para la predicción de calificaciones usando contexto . . . . .	6
1.5.2 Implementación de referencia . . . . .	6
1.5.3 Dataset para publicidad ubicua con información contextual . . . . .	7
1.5.4 Librería Mahout-Context-Recommender . . . . .	7
1.5.5 Adaptación de un modelo de contexto para entornos de publicidad Ubicua . . . . .	7
1.5.6 Publicaciones . . . . .	7

1.6	Contenido de la monografía . . . . .	8
<b>2 Estado actual del conocimiento</b>		<b>9</b>
2.1	Contexto de la investigación . . . . .	9
2.2	Metodología de la revisión . . . . .	10
2.3	Trabajos relacionados . . . . .	10
2.3.1	Sistemas de Recomendaciones (SR) . . . . .	10
2.3.2	Publicidad Ubicua . . . . .	11
2.3.3	Sistemas de recomendaciones basados en Contexto . . . . .	13
<b>3 Contexto en entornos de publicidad ubicua y su implantación en Sistemas de Recomendaciones</b>		<b>17</b>
3.1	¿Qué es Contexto? . . . . .	18
3.2	Adaptación de un modelo contextual para el caso específico de publicidad ubicua . . . . .	18
3.2.1	Metodología para la definición de un modelo de contexto . . . . .	18
3.2.2	Adaptación del modelo para publicidad ubicua . . . . .	19
3.3	Aproximación a la inclusión de contexto dentro de los sistemas de recomendaciones . . . . .	21
3.3.1	Prefiltrado y Postfiltrado . . . . .	23
3.3.2	Aproximación basada en modelo . . . . .	26
3.3.2.1	Filtrado user-user . . . . .	26
3.3.2.2	Filtrado item-item . . . . .	27
3.3.2.3	Similitud de usuarios e ítems considerando el contexto . . . . .	27
3.4	Similitud de contextos . . . . .	29
3.4.1	Similitud de contextos incluyendo la similitud de los usuarios o ítems . . . . .	31
3.5	Construcción experimental de un Dataset para publicidad ubicua usando información contextual . . . . .	32

## CONTENIDO

---

3.5.1	Diseño del experimento . . . . .	32
3.5.2	Implementación y ejecución del experimento . . . . .	33
3.5.3	Dataset resultante . . . . .	38
3.5.4	Cálculo de pesos para las variables de contexto . . . . .	38
3.5.5	Cálculo de distancias entre los posibles valores de cada variable contextual . . . . .	40
3.6	Conclusiones del capítulo . . . . .	43
<b>4</b>	<b>Construcción de un sistema de recomendaciones soportado en información contextual</b>	<b>45</b>
4.1	Mahout: framework seleccionado para soportar el sistema . . . . .	46
4.2	Recuperación de la Arquitectura . . . . .	47
4.2.1	Modelo de datos (DataModel) . . . . .	48
4.2.2	Mecanismo de similitud (UserSimilarity o ItemSimilarity) . . . . .	48
4.2.3	Vecindad (UserNeighborhood) . . . . .	48
4.2.4	Recomendador (Recommender) . . . . .	49
4.3	Arquitectura de referencia incluyendo información de contexto . . . . .	49
4.4	Descripción de la arquitectura de referencia mediante el modelo de 4+1 vistas de RUP . . . . .	50
4.4.1	Vista lógica . . . . .	50
4.4.1.1	Diagrama de casos de uso del sistema . . . . .	50
4.4.1.2	Diagrama de clases del sistema . . . . .	50
4.4.1.3	Diagrama entidad relación . . . . .	54
4.4.2	Vista de procesos . . . . .	55
4.4.2.1	Diagramas de secuencia: Solicitud de recomendaciones para un recomendador user-user e item-item incluyendo contexto . . . . .	55
4.4.2.2	Diagramas de actividad: Estimación de preferencias en un recomendador user-user e item-item incluyendo contexto . . . . .	58
4.4.3	Vista de desarrollo . . . . .	60
4.4.3.1	Diagrama de componentes del sistema . . . . .	60

4.4.4	Vista de despliegue . . . . .	61
4.4.4.1	Diagrama de despliegue del sistema . . . . .	61
4.5	Conclusiones del capítulo . . . . .	63
<b>5</b>	<b>Experimentación y resultados</b>	<b>65</b>
5.1	Definición de algoritmos a evaluar . . . . .	66
5.2	Métricas de evaluación . . . . .	67
5.2.1	RMSE: Root Mean Square Error . . . . .	68
5.2.2	Precisión . . . . .	68
5.3	Definición del plan de pruebas . . . . .	69
5.3.1	Pruebas Offline . . . . .	69
5.3.2	Pruebas Online . . . . .	69
5.3.2.1	Aplicativo construido para la evaluación online de algoritmos de recomendaciones . . . . .	69
5.4	Resultados y análisis del plan de pruebas . . . . .	73
5.4.1	Resultados de la prueba offline . . . . .	73
5.4.1.1	Métrica RMSE . . . . .	73
5.4.1.2	Métrica de Precisión . . . . .	74
5.4.2	Resultados de la prueba online . . . . .	76
5.4.2.1	Métrica RMSE . . . . .	76
5.4.2.2	Métrica de Precisión . . . . .	78
5.5	Conclusiones del capítulo . . . . .	80
<b>6</b>	<b>Conclusiones y trabajo futuro</b>	<b>81</b>
6.1	Conclusiones del estado actual del conocimiento . . . . .	81
6.2	Conclusiones de contexto en entornos de publicidad ubicua y su implantación en Sistemas de recomendaciones . . . . .	82

## CONTENIDO

---

6.3	Conclusiones de Construcción de un sistema de recomendaciones soportado en información contextual . . . . .	84
6.4	Conclusiones de experimentación y resultados . . . . .	84
6.5	Trabajo Futuro . . . . .	85
	<b>Bibliografía</b>	<b>87</b>
	<b>A Descripción de casos de uso de la librería Mahout-Context-Recommender</b>	<b>95</b>
	<b>B Implementación de un Sistema de Recomendaciones que involucra información contextual utilizando la librería “Mahout Context Recommender”</b>	<b>101</b>
B.1	Agregar las librerías necesarias al proyecto . . . . .	101
B.2	Instanciar un Modelo de datos que incluye información contextual . . . . .	102
B.3	Crear un objeto para medir la similitud entre contextos . . . . .	103
B.4	Crear un objeto para medir similitud entre usuarios o ítems incluyendo contexto . . . . .	103
B.5	Crear el recomendador utilizando información de contexto . . . . .	105
	<b>C Ficha técnica de publicaciones</b>	<b>107</b>
	<b>D Información del Dataset para publicidad ubicua con información contextual generado experimentalmente</b>	<b>111</b>
D.1	Información contenida en el dataset . . . . .	111
D.2	Dataset resultante . . . . .	113



## LISTA DE TABLAS

	<b>Pág.</b>
2.1 Resumen de brechas encontradas . . . . .	15
3.1 Matriz de factorización de ejemplo . . . . .	28
3.2 Valores del coeficiente de correlación de Pearson . . . . .	28
3.3 Similitud entre usuarios considerando el contexto . . . . .	29
3.4 Información del dataset resultante . . . . .	38
3.5 Dataset extraído como ejemplo para el cálculo de distancia de la variable hora . . . . .	40
3.6 Ítems comunes para el cálculo de distancia . . . . .	41
3.7 Cálculo final de la distancia para los posibles valores de la variable hora . . . . .	41
3.8 Distancias entre cada una de las variables de contexto . . . . .	42
4.1 Frameworks para la construcción de SR explorados . . . . .	47
5.1 Clasificación de los posibles resultados de una recomendación de un ítem a un usuario	69
5.2 Resultados de la métrica RMSE en la prueba online . . . . .	77
5.3 Resultados de prueba T-Student para valores RMSE . . . . .	78
5.4 Resultados de prueba T-Student para valores de RMSE de context-based-user . . . . .	79
5.5 Resultados de la métrica de Precisión en la prueba online . . . . .	79
5.6 Resultados de prueba T-Student para valores de Precisión de context-based-user . . . . .	80



## LISTA DE FIGURAS

	<b>Pág.</b>
1.1 Escenario de motivación . . . . .	2
3.1 Modelo de contexto para publicidad . . . . .	20
3.2 Modelo de contexto adaptado . . . . .	21
3.3 Matriz de factorización 2D . . . . .	22
3.4 Matriz de factorización 3D . . . . .	23
3.5 Inclusión de contexto . . . . .	24
3.6 Variables de contexto seleccionadas . . . . .	33
3.7 Diagrama de casos de uso para experimento de Dataset contextual . . . . .	34
3.8 Experimento dataset contextual - registro . . . . .	35
3.9 Interfaz de Login para experimento dataset contextual . . . . .	35
3.10 Interfaz de configuración del contexto para experimento dataset contextual . . . . .	36
3.11 Interfaz de calificación del anuncio para expermineto dataset contextual . . . . .	36
3.12 Interfaz de finalización para expermineto dataset contextual . . . . .	37
3.13 Errores en Interfaces Gráficas iniciales de Experimento de construcción del dataset . . . . .	38
3.14 Pesos porcentuales de cada uno de los ambientes . . . . .	39
3.15 Pesos porcentuales de cada una de las variables del contexto separadas en ambientes . . . . .	39
4.1 Módulos de un SR en Mahout . . . . .	48
4.2 Módulos de un SR incluyendo contexto en Mahout . . . . .	49

## LISTA DE FIGURAS

---

4.3	Diagrama de casos de uso del sistema . . . . .	51
4.4	Diagrama de clases del sistema . . . . .	52
4.5	Diagrama entidad relación del dataset . . . . .	54
4.6	Diagrama de secuencia user-user y contexto . . . . .	56
4.7	Diagrama de secuencia item-item y contexto . . . . .	57
4.8	Diagrama de actividad user-user y contexto . . . . .	58
4.9	Diagrama de actividad item-item y contexto . . . . .	59
4.10	Diagrama de componentes . . . . .	60
4.11	Diagrama de despliegue . . . . .	62
5.1	Diagrama de procesos para evaluación online . . . . .	70
5.2	Interfaz de métrica de precisión . . . . .	72
5.3	Resultados de la prueba RMSE offline para algoritmos user-user . . . . .	73
5.4	Resultados de la prueba RMSE offline para algoritmos item-item . . . . .	74
5.5	Resultados de la prueba de Precisión offline para algoritmos user-user . . . . .	75
5.6	Resultados de la prueba de Precisión offline para algoritmos item-item . . . . .	76
A.1	Diagrama de casos de uso de la librería . . . . .	96
B.1	Creación de la fuente de datos . . . . .	102
B.2	Creación de una piscina de conexiones para la base de datos . . . . .	103
B.3	Creación del modelo de datos con información contextual . . . . .	104
B.4	Instanciación de un objeto para medir similitud entre contextos . . . . .	104
B.5	Instanciación de un objeto para medir similitud entre usuarios e items incluyendo contexto . . . . .	105

**LISTA DE FIGURAS**

---

B.6 Código para la implementación de un SR user-user incluyendo contexto . . . . . 106

D.1 Información del dataset contextual . . . . . 112



# Capítulo 1

## Introducción

¿Cuántas veces ha ocurrido que mientras una persona observa su programa favorito es interrumpida para entregarle algún anuncio publicitario? ¿Cómo podrían describirse sus sentimientos cuando esto sucede? tal vez enojo si es detenida en la parte más emocionante para enseñarle algo que ni siquiera le es de su interés, o tal vez le dé igual porque tampoco estaba tan bueno el programa, pero tal vez emoción porque justo el anuncio publicitario que están presentando es algo que le interesa mucho. Esta forma de reacción ante un anuncio es común en las personas, constantemente existe un bombardeo de publicidad, tanto que muchas veces ni siquiera llega a ser percibida, en la televisión, en la bandeja de correo electrónico, en la calle, en internet, etc, y la forma como un individuo reacciona frente a ella depende en gran medida de sus gustos y preferencias, pues de antemano existe una predisposición a aceptar ciertas cosas y a rechazar algunas otras. Pero, ¿también pasa que algo que normalmente sería rechazado consigue llamar la atención porque se está con otra persona a quien le gusta? seguramente que sí, y es porque las personas que están cerca entre sí logran influenciarse las unas a las otras y logran cambiar sus perspectivas; y no sólo las personas al rededor, sino también el ambiente en general, la temperatura, estado de ánimo, hora del día, día de la semana, son elementos que pueden influir y hacer que alguien acepte cosas que normalmente rechazaría o por el contrario rechace cosas que normalmente aceptaría.

Lo anterior permite comprender que la publicidad puede ser más efectiva en la medida en que se entregue en el lugar justo y en el momento justo, y este momento y lugar vienen acompañado no solo de una fecha y una localización, sino de un estado actual de la temperatura, de un estado de ánimo, de un conjunto de personas, de muchas otras variables que suelen conocerse como contexto. De esta forma, considerar la información de contexto se convierte en un elemento relevante en el momento de pensar en recomendarle publicidad a un usuario, y es por este motivo que este trabajo se enfoca en la construcción de un sistema de recomendaciones para publicidad ubicua que involucra información contextual. Para ello, se definirán el conjunto de variables que serán consideradas como contexto, el o los mecanismos para incluirlas en el sistema y las herramientas para su implementación y evaluación. A lo largo de este documento se presentarán un conjunto de decisiones tomadas a partir del análisis de trabajos relacionados relevantes, paradigmas de diseño de software y evaluación y pruebas del sistema construido.

## 1.1. Escenario de motivación



Figura 1.1: Escenario de motivación Fuente: creación propia

Considérese la siguiente situación. Es el final de la Champions League, Real Madrid contra el Barcelona se enfrentan en un clásico que no es posible dejar de ver, así que un grupo de amigos deciden reunirse para ver el partido en casa de Andrés, la decisión básicamente se tomó porque es él quien cuenta con el televisor mas grande y con el mejor servicio de televisión satelital, sin contar con que su vivienda no es la más grande pero cuenta con el espacio necesario y suficiente para albergar a todo el grupo. Todo está listo para ver el partido, hay suficiente comida y Carlos se ha comprometido a llevar las bebidas, todos han llegado a casa de Andrés y como lo prometió Carlos ha llevado las bebidas, sólo que calculó mal el número de personas y no ha llevado suficiente. La estación del año es verano, así que el calor empieza a invadir el lugar y peor aún con la gran cantidad de personas que se encuentran allí. A sólo dos cuadras de casa de Andrés hay una tienda de barrio donde están estrenando congelador, cuentan con bebidas muy frías y además ofrecen servicio de domicilios. Sin embargo, ninguna de las personas en casa de Andrés saben de la existencia de esta tienda y peor aún, no quieren despegarse ni un minuto del televisor. Afortunadamente, el dueño de la tienda ha suscripto su negocio a un sistema de publicidad, este sistema cuenta en su base de datos con muchos anuncios, incluyendo ropa deportiva, y presenta a sus usuarios una experiencia personalizada acorde a los gustos de cada uno. Para Andrés o Carlos el sistema ha determinado que les gusta mucho los deportes y puede presentarles como publicidad la ropa deportiva, pero este sistema considera además el contexto y al determinar la temperatura recomienda los domicilios de bebidas en su lugar.

El escenario anterior motiva a la construcción de un sistema que incluya lo necesario para hacer precisamente lo que se ha mencionado, pues la publicidad justo dentro del partido podría ser

## 1.2. Planteamiento del problema

---

rechazada pero si es lo que el usuario desea y resuelve una necesidad en el momento, puede ser muy atractiva. Adicionalmente, un sistema de recomendaciones de publicidad podría suplir un problema que normalmente se conoce como publicidad spam y que se da justamente porque al enviar los anuncios se hace de forma indiferente y se termina abrumando al individuo con demasiadas cosas, algunas buenas y otras no, pero al final, tantas que para el usuario es una pérdida de tiempo revisar cada una y termina rechazando todo. La figura 1.1 resume este problema.

## 1.2. Planteamiento del problema

La publicidad es considerada como una relación que permite a los proveedores compartir información con los consumidores buscando un beneficio para ambos, pues educa a estos últimos acerca de los productos y servicios disponibles que pueden satisfacer sus necesidades y ayuda a los vendedores a crear un ambiente agradable para su acercamiento con los clientes [1]. A pesar de esto, cada día las personas son más reacias a aceptar contenidos publicitarios. Esto puede apreciarse por ejemplo en trabajos como [2, 3] donde se realizó un estudio de la eficiencia del mercadeo utilizando blogs y curiosamente se concluye que el incluir contenido publicitario en estos disminuye la confianza de los usuarios sobre el contenido del mismo. Por otro lado, un estudio sobre la postura de los consumidores frente a la publicidad realizado por la Asociación Americana de Agencias de Publicidad (American Association of Advertising Agencies) en abril de 2004 [4] mostró un 69 % de interés en productos y servicios que ayuden a bloquear la publicidad y un 65 % de afirmación que la publicidad es siempre spam. Una explicación para estos resultados puede basarse en la forma como por mucho tiempo ha sido emitida la publicidad, pues es indiferente a todos los consumidores y no tiene en cuenta sus gustos y preferencias [5].

A su vez, una tecnología que se ha hecho popular en la última década es la conocida como “computación ubicua”, la cual presenta un mundo donde las soluciones computacionales y los servicios de comunicación están disponibles en todo momento y en todo lugar [6], permitiendo la construcción de sistemas que pueden adaptarse a los constantes cambios de su entorno y soportando las actividades diarias de sus usuarios [7]. Esta concepto de computación ubicua da paso a lo conocido como “Publicidad ubicua” refiriéndose al uso de la computación ubicua con fines publicitarios [8]. En este orden de ideas, las características esenciales de este paradigma computacional (ubicuidad, interactividad y automatización) son ideales para transformar los esquemas de publicidad tradicionales a través de mecanismos de comunicación simétrica, personalización, medición precisa de audiencia, adaptación al contexto y persuasión; dando a la publicidad las herramientas para bordear su principal falencia, relacionada con la ausencia de personalización de los anuncios que son entregados a los clientes potenciales. No obstante, a pesar de todas las ventajas mencionadas la publicidad ubicua enfrenta una serie de retos relacionados con aspectos como: la entrega del contenido apropiado en el momento adecuado (personalización), privacidad de los usuarios, publicidad invasiva, usabilidad, gestión de perfiles de usuario, entre otros, que han sido abordados a través de diversas aproximaciones en diferentes trabajos de investigación [9, 10, 11, 12, 13, 14].

Como ya se ha mencionado una de las posibles causas por las cuales clientes potenciales pierden interés en la publicidad es la baja personalización de la misma. No obstante, incluir mecanismos de personalización sugiere una necesidad de interacción entre los consumidores y los sistemas encarga-

dos de desplegar el contenido. En consecuencia, la publicidad no puede ser un canal de una sola vía, ésta debe percibir y entender las necesidades y preferencias de los consumidores, lo que es posible aprovechando las bondades de la computación ubicua. Con el objetivo de entender las necesidades del usuario y aprovecharlas en el proceso de compra, se han llevado a cabo investigaciones en temas como diseño centrado en el usuario, minería de datos e incluso sistemas de recomendaciones [15]. Estos últimos como se presenta en [16], son cualquier sistema que produce recomendaciones individualizadas o que tiene el efecto de guiar al usuario en un camino personalizado de acuerdo a sus intereses para tomar una decisión en un gran espacio de posibles opciones. Es decir, personalizan y filtran los contenidos acorde a las preferencias de los usuarios. Es claro entonces que incluir un sistema de recomendaciones en un entorno de publicidad ubicua podría lidiar con el problema de baja personalización de los anuncios y disminuirlo. Sin embargo, en un proceso de compra la personalización acorde a los gustos del usuario no es suficiente, debido a que si bien un anuncio publicitario puede ser atractivo para un cliente potencial, no necesariamente es lo suficientemente impactante para convertir dicha atracción en la adquisición del producto o servicio en cuestión. Es aquí donde la “Información Contextual” toma un papel importante. Como se menciona en [17] la información del contexto del comprador es críticamente valiosa para los sistemas de recomendaciones en entornos comerciales, en tanto que ignorar datos como la posición del consumidor en la tienda, su objetivo de compra o cualquier otra variable contextual pone al sistema de recomendaciones en un grave problema de rendimiento no óptimo si se evalúa con una métrica mercantil. El término “Contexto” cuenta con diferentes definiciones, una de ellas y la que mejor se ajusta a lo considerado como Contexto en este documento es la de Schillit en [18]: “*El contexto se compone de tres aspectos importantes: dónde estás, con quién estás y con qué recursos cuentas*”, que, como puede apreciarse divide el contexto en tres partes, localización, naturaleza social, y las herramientas para percibir datos del usuario (teléfono móvil, sensores, etc.).

De lo anteriormente dicho, es evidente la necesidad de personalizar los anuncios publicitarios que son entregados a los usuarios, ya que como se comentó, estos tienen el objetivo de beneficiar tanto a clientes como a proveedores. Es indiscutible que la publicidad debe ser un canal de doble vía y no un sistema broadcast que se envía indiferentemente a toda la población, lo que es posible aprovechando todas las bondades de la computación ubicua. Es claro también que un sistema de recomendaciones puede incluir una capa de personalización que podría alivianar el problema, pero que dicha personalización no es suficiente si no se considera la información contextual. De esta manera surge la siguiente pregunta de investigación **¿Cómo mejorar las recomendaciones de contenido publicitario teniendo en cuenta la información contextual de los usuarios?**

### 1.3. Objetivos

#### 1.3.1. Objetivo general

Construir un Sistema de Recomendaciones para entornos de Publicidad Ubicua que involucre Información Contextual.

## 1.4. Actividades

---

### 1.3.2. Objetivos específicos

- Realizar una caracterización inicial de variables contextuales que puedan aplicarse a sistemas de recomendaciones en entornos de publicidad ubicua.
- Diseñar un filtro de recomendación que involucre algunas variables contextuales de acuerdo a la caracterización realizada.
- Construir un piloto en el contexto de publicidad ubicua que permita evaluar el filtro de recomendaciones desarrollado.

## 1.4. Actividades

La estructura básica de las actividades propuestas para el desarrollo del presente trabajo, toma como referencia la descomposición jerárquica WBS (Work Breakdown Structure) sugerida por el Project Management Institute (PMI), a través de su metodología PMBOK (Project Management Base of Knowledge), específicamente en el área de gestión del alcance (Scope Management) [19]. Los procesos de ingeniería del software requeridos, serán abordados a través de un extracto y adaptación de la metodología UP (Unified Process) [20]. La metodología que se utilizará para la evaluación del sistema de recomendaciones (paquete de trabajo 4) se basa en los mecanismos de evaluación clásicos de los SR los cuales se realizan en dos fases: la primera fase offline en la cual a partir de herramientas estadísticas es posible calcular la precisión del SR. La segunda fase se realiza de forma online utilizando usuarios, en ella no solo se busca una precisión estadística objetiva, sino una precisión subjetiva desde el punto de vista de los usuarios a partir de encuestas, donde además es posible incluir métricas propias de la publicidad.

A continuación se realiza un resumen de actividades para cada uno de los paquetes de trabajo propuestos.

### 1.4.1. Generación de la base inicial de conocimiento

- Revisión del estado del arte.
- Síntesis.
- Construcción de la base teórica.

### 1.4.2. Caracterización del Contexto

- Exploración de modelos de contexto para entornos de publicidad ubicua .
- Análisis de las variables contextuales a ser consideradas.
- Caracterización de las variables con fines de recomendación.

- Generación de un dataset para publicidad ubicua con información contextual de forma empírica.

### 1.4.3. Diseño del Sistema de Recomendaciones

- Selección de aproximación para reducción de dimensionalidad.
- Formalización matemática del algoritmo de filtrado usando información contextual.
- Diseño e implementación del algoritmo de filtrado.

### 1.4.4. Construcción del piloto de pruebas y evaluación del Sistema de Recomendaciones

- Exploración de frameworks para la construcción de Sistemas de recomendaciones.
- Definición del entorno del piloto.
- Estructuración del plan de pruebas (offline y online).
- Puesta en marcha del piloto.
- Recolección de datos y análisis.

### 1.4.5. Publicación

- Socialización en el seminario.
- Generación de artículos sobre los avances de la investigación.
- Generación del documento final (monografía).

## 1.5. Contribuciones

### 1.5.1. Mecanismos para la predicción de calificaciones usando contexto

Se logró generar algunas aproximaciones para la información contextual utilizando similitud de contextos. Dichas aproximaciones desembocaron en algunas ecuaciones que al implementarse permiten generar recomendaciones basadas en contexto para anuncios publicitarios.

### 1.5.2. Implementación de referencia

Durante del desarrollo de este proyecto logró construirse una implementación de referencia para sistemas de recomendaciones que involucran información contextual en entornos de publicidad ubicua y que puede ser extendida a otros dominios de aplicación.

## 1.5. Contribuciones

---

### 1.5.3. Dataset para publicidad ubicua con información contextual

A partir de un experimento empírico logró generarse un dataset de publicidad ubicua que involucra información contextual. Dicho dataset se encuentra disponible para ser utilizado en trabajos relacionados y la plataforma creada para construirlo se encuentra funcionando en el siguiente link: <http://bitnami-tomcatstack-9d5b.cloudapp.net/ContextualDS/faces/pages/index.xhtml>. En el anexo D se encuentra descrito en detalle.

### 1.5.4. Librería Mahout-Context-Recommender

Esta librería construida a partir de las ecuaciones definidas y apoyada en el framework Mahout, permite la construcción de sistemas de recomendaciones que involucran información contextual de forma simple y ágil. En el anexo A pueden encontrarse descritas todas las funcionalidades que un programador puede utilizar de la librería y en el anexo B se encuentra el paso a paso para la construcción de un SR utilizando la misma.

### 1.5.5. Adaptación de un modelo de contexto para entornos de publicidad Ubicua

Se generó un modelo de información contextual a partir de la adaptación de modelos existentes en la literatura, con el objetivo de permitir su utilización en sistemas de recomendaciones para entornos de publicidad ubicua.

### 1.5.6. Publicaciones

Con el avance de esta investigación se consiguió realizar algunas publicaciones científicas que se listan a continuación:

- **Sistemas de recomendaciones para entornos de Digital Signage soportado en un esquema de cooperación SmartTV - Smartphone**  
Revista de Ingenierías Universidad de Medellín - A2.
  
- **Enriching public displays ads recommendations using an individual - group cooperation model**  
Conferencia: *3rd International Symposium on pervasive displays* - ACM New York.
  
- **Context aware recommender system for pervasive advertising: experimental approach**  
Pendiente de publicación.

El detalle de los artículos se encuentra en el Anexo C (publicaciones).

## 1.6. Contenido de la monografía

### Capítulo 1. Introducción

Definición del problema y la estructura general del desarrollo del proyecto de investigación.

### Capítulo 2. Estado actual del conocimiento

Este capítulo presenta los trabajos relacionados al proyecto los cuales se han dividido en tres grupos temáticos: sistemas de recomendaciones, publicidad ubicua y sistemas de recomendaciones basados en contexto. Para cada uno de dichos grupos se presentan las brechas más importantes halladas.

### Capítulo 3. Contexto en entornos de publicidad ubicua y su implantación en Sistemas de Recomendaciones

En él se realiza una definición de las variables de contexto para entornos de publicidad ubicua y se construyen todas las ecuaciones que soportarán la integración de dichas variables dentro de un sistema de recomendaciones. Además se presenta un experimento cuyo objetivo es la generación de un dataset para publicidad ubicua utilizando el modelo contextual planteado.

### Capítulo 4. Construcción de un sistema de recomendaciones soportado en información contextual

A partir de las ecuaciones planteadas en el capítulo 3, se diseña una arquitectura de referencia que permita la implementación del sistema de recomendaciones en sí.

### Capítulo 5. Experimentación y resultados

Presenta el plan de pruebas diseñado para determinar si existe una mejora en el sistema de recomendaciones al aplicar información contextual. El plan de pruebas se divide en una fase offline donde sólo se trabaja con información contenida en el dataset y una fase online donde se utiliza un prototipo funcional para permitir que usuario reales puedan interactuar con el sistema. Al final del capítulo se presentan los resultados obtenidos de ambas fases y su análisis.

### Capítulo 6. Conclusiones y trabajo futuro

Por último, se analizan los resultados del trabajo realizado, se detallan las principales contribuciones obtenidas durante el ciclo del proyecto y se expone un conjunto de recomendaciones importantes para el desarrollo de trabajos futuros.

## Capítulo 2

# Estado actual del conocimiento

### 2.1. Contexto de la investigación

Sin duda alguna la publicidad se ha convertido en un elemento tan natural en la cotidianidad de las personas que muchas veces simplemente está allí y es percibida casi como parte de la decoración. Aún más, este mecanismo de comunicación se ha hecho tan relevante que muchos avances tecnológicos han surgido con el objetivo de dar soporte a necesidades propias del mismo, e incluso ha permitido la generación de nuevos modelos de negocio tan exitosos que han concedido que compañías como Google y Amazon existan y facturen miles de millones de dólares al año. Dicha publicidad, se ha incrementado con el constante incremento de tecnología y en muchas ocasiones se ha convertido en una molestia, generando lo que comúnmente se conoce como publicidad spam. Sin embargo, considerando que ella ha sido concebida para ser benéfica, es necesario un esfuerzo por mejorar la forma como los usuarios reciben los contenidos. De esta forma, algunos proyectos se han enfocado en cumplir este objetivo mediante la utilización de sistemas de recomendaciones, y para ello, se han apoyado en el concepto de computación ubicua, considerado como la posibilidad de utilizar sistemas de comunicación y computación en todo momento y en todo lugar. De esta manera, se generan ambientes de interacción que pueden alimentar y mejorar la forma como los usuarios reciben y consumen la publicidad.

Ahora bien, aunque este tipo de sistemas pueden suplir un poco este problema, no es suficiente en publicidad considerar solo las pocas variables que un sistema de recomendaciones clásico considera, es decir, no es suficiente con considerar solo los usuarios y los ítems. Una razón es porque la publicidad es principalmente situacional, tanto que [5] afirma que el 75 % de las decisiones de compra se toman en el sitio de venta o cerca de él; otra razón es porque esta naturaleza situacional implica que la publicidad dependa mucho del entorno o como es llamado, del contexto, pues no es lo mismo ofrecer un artículo cuando hace frío que cuando hace calor, cuando la persona se siente enojada que cuando se siente feliz, incluso [17] indica que no considerar variables como la posición del usuario para un sistema de publicidad lo pone en un problema de bajo rendimiento. Es por este motivo, que este trabajo se enfoca en el diseño y construcción de un sistema de recomendaciones que involucre información contextual. Para ello, se investigará sobre las posibles variables que deben considerarse como contexto en un entorno de publicidad, las principales características de los sistemas de recomendaciones, los mecanismos o aproximaciones para incluir las variables de contexto definidas dentro del cálculo de recomendaciones y las herramientas para la implementación

del mismo.

En las siguientes secciones, se describe brevemente la metodología utilizada para la construcción el estado del arte y el conjunto de trabajos relacionados más relevantes para el contexto del trabajo de investigación.

### 2.2. Metodología de la revisión

El enfoque de vigilancia tecnológica, metodología utilizada para abordar la revisión del estado del arte, se define como *“la búsqueda, detección, análisis y comunicación de informaciones orientadas a la toma de decisiones sobre amenazas y oportunidades externas en el ámbito de la ciencia y tecnología”* [21] y se realiza en 3 fases principales: 1) planeación, donde se definió el objetivo de la revisión, se identificaron las fuentes más relevantes para el contexto de investigación (ACM Digital Library, IEEE Xplore Digital Library, Google Scholar, Springer) y se definieron las palabras clave que orientaron la búsqueda (sistemas de recomendaciones, publicidad ubicua, sistemas de recomendaciones basados en contexto, sistemas de recomendaciones multidimensionales); 2) búsqueda y captación, a través de la cual se obtiene un conjunto de los artículos más relevantes relacionados con la presente investigación; y 3) análisis, donde se realiza un estudio de los artículos previamente clasificados en tres grandes grupos temáticos (sistemas de recomendaciones, publicidad ubicua y sistemas de recomendaciones basados en contexto) y, con base en herramientas de análisis de datos como Matheo Analyzer, se obtiene un listado de las fuentes mas relevantes.

Los resultados arrojados por esta vigilancia mostraron que temas como sistemas de recomendaciones y publicidad ubicua han sido ampliamente estudiados. Asimismo, aunque la inclusión de información de contexto en los sistemas de recomendaciones no es un tema nuevo de investigación, los resultados obtenidos de esta búsqueda no son tan amplios como los dos anteriores. Por otro lado los sistemas de recomendaciones de publicidad ubicua son un tema en proceso de investigación, puesto que los resultados parecen ser hasta el momento muy escasos y casi nulos si se considera información de contexto en los mismos. A continuación se presentan los trabajos relacionados clasificados en los grupos temáticos ya definidos, al finalizar la sección se presenta una tabla a modo resumen de las brechas mas importantes encontradas en cada grupo.

### 2.3. Trabajos relacionados

#### 2.3.1. Sistemas de Recomendaciones (SR)

Los sistemas de recomendaciones o SR son herramientas y técnicas software que proporcionan sugerencias basadas en distintos procesos de toma de decisiones sobre items para ser usados por un usuario. Dichos ítems pueden representar cualquier cosa, como por ejemplo: qué comprar, qué música escuchar o qué noticias leer en línea [22]. Este tipo de sistemas no son un tema nuevo de investigación, de hecho varios proyectos han aprovechado sus bondades para llegar de una manera mucho más personalizada a sus usuarios y los han utilizado en un sin número de campos, desde la recomendación de noticias como se hace en [23, 24], hasta en la realización de sistemas para la solución de preguntas de usuarios como Yahoo! Answers presentado en [25]; pasando por

## 2.3. Trabajos relacionados

---

sistemas de recomendaciones para libros, películas, programas de televisión, información para investigadores, e incluso planes vacacionales [26, 27, 28, 29, 30, 31]. Por supuesto sin dejar de lado los sistemas de recomendación de publicidad, [32] es un excelente ejemplo de ello. En él se presenta un sistema de recomendaciones para publicidad desplegado en un gimnasio y según se menciona se enfoca mucho en la obtención de las preferencias del usuario y la combinación de estas preferencias con las de otros usuarios. Sin embargo, no hace un análisis completo del contexto y por el contrario se enfoca más en realizar las recomendaciones grupales. Por otro lado, aunque [33] no es exactamente un sistema para la recomendación de publicidad, se relaciona mucho, pues en él se indica un sistema que intenta encontrar el mejor espacio publicitario para un anuncio basándose en el objetivo del publicista, esto es, si se requiere que el usuario descargue una app o de clics en el anuncio.

En [34] se presenta una conceptualización muy completa de los sistemas de recomendaciones y un estudio del estado del arte de los mismos. Este trabajo clasifica los sistemas de recomendaciones en tres grupos: (1) basados en contenido o aquellos que utilizan información de los ítems que el usuario ha calificado para recomendar ítems similares que puedan interesarle; y (2) filtros colaborativos que, en el caso de ser User-User recomiendan al usuario elementos que fueron atractivos para otros individuos con gustos similares, a su vez, un filtro colaborativo Item-Item sigue la misma idea pero basada en ítems similares. Para calcular dicha similitud entre usuarios, el filtrado colaborativo depende de las calificaciones pasadas de ambos usuarios ya que estos se consideran cercanos si han dado calificaciones similares a los mismos elementos. Finalmente (3) representa los sistemas híbridos que combinan los dos anteriores [35]. Otros trabajos como [36, 37] representan extensiones de estos mecanismos de recomendaciones bajo aproximaciones basadas en conocimiento. Estos últimos se enfocan en entregar recomendaciones basados en especificaciones explícitas del tipo de contenido que el usuario solicita al sistema, y se han clasificado en dos tipos: los basados en limitaciones [38, 39, 40] donde el usuario entrega características que limitan el contenido, por ejemplo, “*rock de la década del 80*”; y los basados en casos [41, 42], en los cuales los usuarios solicitan los ítems al sistema a partir de sus propios objetivos. Adicionalmente, es importante destacar que para la construcción de los SR herramientas como Apache Mahout [43] y Lenskit [44], entre otras, han surgido para ser utilizadas tanto comercial como académicamente.

En síntesis, los trabajos anteriores resumen los conceptos básicos de los sistemas de recomendaciones aplicados a diferentes dominios y como se puede apreciar, existen sistemas de este tipo para entornos de publicidad. Sin embargo, no muchos han incluido información de contexto y los que lo han hecho no han definido un dominio de variables enfocadas directamente a aplicarse dentro de la recomendación.

### 2.3.2. Publicidad Ubicua

En [1] la publicidad ubicua es definida como “el uso de computación ubicua con fines publicitarios”, en este trabajo en particular se presenta un sistema que intenta entregar anuncios mas precisos acorde a las actividades realizadas por el usuario. Adicionalmente [45] y [46] presentan implementaciones similares en un mercado de compras de productos de granja (donde se aplica lo presentado en [1]) y en un ambiente rural o comunidad pequeña respectivamente. Estos trabajos son sólo una muestra de los múltiples entornos en los que puede ser incluido un sistema de publicidad ubicua. Sin embargo, cabe aclarar que se centran específicamente en presentar el concepto y algunas implementaciones, mas no

en sistemas de recomendaciones de publicidad ubicua. De igual forma es importante mencionar que se encuentran incluidos en [8], convirtiendo este último en un punto de partida importante para todos los trabajos relacionados con este tema.

Como puede inferirse, el concepto de publicidad ubicua puede apreciarse muy extenso si se considera que cualquier dispositivo con capacidad de computo que permita hacer uso de las bondades de la computación ubicua (smartphone, computadora portátil, smarttv, etc) puede dar paso a la implementación de ello. En este sentido se han realizado subdivisiones acorde al tipo de mecanismo utilizado para desplegar la publicidad, una de estas subdivisiones es la conocida como “Digital Signage”, que es un concepto desarrollado por Jörg Müller y se refiere a la utilización de pantallas públicas para el despliegue de anuncios [47], mientras que en [48] se estudian los tipos de contenidos que pueden incluirse en este modelo. Por su parte en [49] se presentan los retos y las oportunidades que tiene este tipo de publicidad y en se especifica la necesidad de utilizar nuevas tecnologías para el reconocimiento de usuarios y sistemas de recomendaciones basados en contexto para presentar anuncios relevantes acorde a los usuarios reconocidos. De esta manera, trabajos como [50, 51, 52, 53, 54] han abordado esta concepción y han utilizado tecnologías como bluetooth para acceder a la información del usuario y agregar algo de personalización a la publicidad. Es importante agregar que estos trabajos no se centran en la utilización de sistemas de recomendaciones y no realizan una definición detallada del contexto que será tenido en cuenta.

Al mismo tiempo, trabajos como [55] muestran la necesidad de incluir información contextual en ambientes públicos, donde la publicidad ubicua se presenta como una posible implementación de este paradigma. Asimismo, en [17] se ha estudiado la posibilidad de incluir información de contexto dentro sistemas de recomendaciones para entornos de comercio electrónico (cercano con publicidad ubicua) y se han centrado en el concepto de “Publicidad Contextual” definiéndolo como toda publicidad presentada en un sitio Web que se ajusta al contenido del sitio y a las palabras clave que el usuario introdujo para llegar allí. No obstante, este tipo de publicidad podría definirse como toda aquella que se ajusta a un conjunto de variables contextuales, sea cual sea la definición que se le de al contexto. En este sentido, Bauer en [56] presenta un sistema de despliegue de publicidad en dispositivos móviles basado en la localización del usuario, para ello el sistema conoce previamente anuncios relevantes en ciertas zonas y los utiliza para entregarlos al usuario en el lugar preciso o cercano, sobra decir que se trata de un SR que aunque no necesariamente utiliza un enfoque de filtrado colaborativo, considera una variable de contexto para las recomendaciones. De igual manera en [57] y en [58] se presentan trabajos donde se incluye algo de información contextual utilizando mecanismos de filtrado semántico, en ellos al igual que en trabajos mencionados previamente no se realiza una definición sobre lo que es considerado como contexto, es decir, no se hace un definición de las variables contextuales. Pero en [5] si presenta una aproximación para la utilización de información contextual en entornos de publicidad ubicua. Para ello se define que esta debe ser tratada como un servicio adaptativo, el cual cuenta con cuatro fases definidas al integrar elementos comunes en [59, 60, 61, 62] y se describen de la siguiente manera: **1) Conceptualización:** Identificar las variables que serán tenidas en cuenta como información contextual; **2) Computación:** Medir y almacenar las variables que serán tenidas en cuenta; **3) Adaptación:** esta fase corresponde a utilizar todas las variables que han sido medidas y traducirlas a una acción, en este caso, dicha acción puede corresponder a una recomendación; y **4) Interfaces de Salida:** Corresponde a los dispositivos que permitirán desplegar la información a los usuarios. Además, este artículo, se enfoca principalmente en la fase de conceptualización, y en este sentido tiene como aporte un modelo inicial de lo que

## 2.3. Trabajos relacionados

---

puede ser el contexto para publicidad ubicua basándose en la definición de contexto de Schillit. Por lo tanto, este trabajo proporciona un punto de partida importante para la caracterización del dominio de publicidad ubicua, pero no se ha considerado su adaptación con fines de recomendación ni tampoco se ha realizado una implementación de referencia, así que su aporte es fundamentalmente conceptual.

### 2.3.3. Sistemas de recomendaciones basados en Contexto

En [63] se presenta un paradigma donde además de la información de los usuarios y de los ítems, se utiliza información contextual para alimentar el sistema de recomendaciones, dicha información contextual es organizada jerárquicamente e incluida dentro de una ontología que permita de una forma semántica encontrar la similitud entre los contextos. Cabe destacar que el sistema de recomendaciones propuesto es para hoteles y la información recolectada se centra en determinar la cercanía entre el usuario y sus acompañantes en un viaje, lo que la convierte en insuficiente en entornos de publicidad ubicua.

Dicha inclusión de información contextual en los sistemas de recomendaciones se ha convertido en un tema importante de investigación en los últimos años. Trabajos como [64, 65, 66] indagan sobre la forma como el contexto puede influenciar las decisiones de los usuarios y aunque presentan algunos casos de estudio de sistemas implementados, no se enfocan en la construcción para un ambiente específico. Por otro lado, este tipo de sistemas han sido implementados para la recomendación de viajes, material educativo, música, películas, entre otros, [67, 68, 69, 70, 71]. Ellos muestran la importancia de este campo de investigación y aunque no se enfocan en el dominio específico de publicidad pueden aportar información relevante para la inclusión de datos de contexto al sistema.

El incluir información contextual dentro de los sistemas de recomendaciones sugiere migrar del espacio clásico bidimensional (matriz de factorización bidimensional)  $R : Usuario \times Item$  a un espacio multidimensional  $R : Usuario \times Item \times Contexto$ , donde el contexto puede estar conformado por una o más variables complicando un poco el modelo matemático de los SR clásicos y obligando a adoptar mecanismos que permitan lidiar con el sinnúmero de variables que pueden incluirse en el contexto. El concepto detrás de [72] es que es posible regresar a una matriz bidimensional aplicando un proceso de reducción de dimensionalidad donde básicamente se establece un valor para la variable contextual y se selecciona la matriz de factorización bidimensional que cumpla con dicho valor de contexto. Luego de la reducción de dimensionalidad es posible trabajar con SR clásicos. Cabe aclarar que este trabajo sólo se centra en la presentación del concepto de reducción de dimensionalidad y no en un Sistema de Recomendaciones específico. Como aporte importante de este trabajo se encuentra la posibilidad de usar una reducción de dimensionalidad dentro de la construcción del Sistema de Recomendaciones que incluye contexto. Sin embargo, no es claro cual debe ser el mecanismo utilizado para realizar dicha reducción, trabajos como [73, 74, 75, 76] presentan distintas opciones para efectuar este proceso, entre las que se incluyen árboles de decisión aleatorios, razonamiento basado en casos, entre otros.

Finalmente, en [77] se tiene un enfoque algo diferente, pues hasta el momento todos los trabajos que se han presentado han indicado que el orden para generar recomendaciones usando contexto consiste inicialmente en hacer una reducción de dimensionalidad y luego utilizar la matriz resultante en un sistema de recomendaciones clásico. Por el contrario este trabajo indaga sobre el mejor momento de

realizar la reducción de dimensionalidad y expresa tres posibilidades: 1) hacer un prefiltro: tal cual como se había considerado antes; 2) hacer un post-filtro: en este caso primero se trabaja con todos los datos del dataset sin considerar la información de contexto y al resultado aplicarle una reducción de dimensionalidad y; 3) incluir la información de contexto directamente en el modelo del SR. Este trabajo hace evidente la necesidad de determinar el momento justo para realizar la reducción de dimensionalidad, exponiendo que no existe un punto para hacerlo que funcione bien para todos los SR, y por tanto es necesario hacer pruebas de ensayo y error para identificarlo. Trabajos como [78] han intentado hacer esta comparación, y aunque no logran concluir algo que se cumpla para todos los sistemas, presentan un modo de proceder para evaluar cual de las dos posiciones del filtro es mejor para un dataset específico si el prefiltro o el postfiltro. Adicionalmente, Zheng en [79] indica la posibilidad de utilizar una aproximación basada en modelo y para ello se basa en la similitud de los contextos, sus experimentos sólo utilizan la variable tiempo dentro del contexto y en ambientes de turismo y restaurantes parecen mejorar el valor de precisión del sistema.

Con esto se da por concluido el estado actual del conocimiento revisado, la tabla 2.1 resume las brechas encontradas en cada grupo temático.

## 2.3. Trabajos relacionados

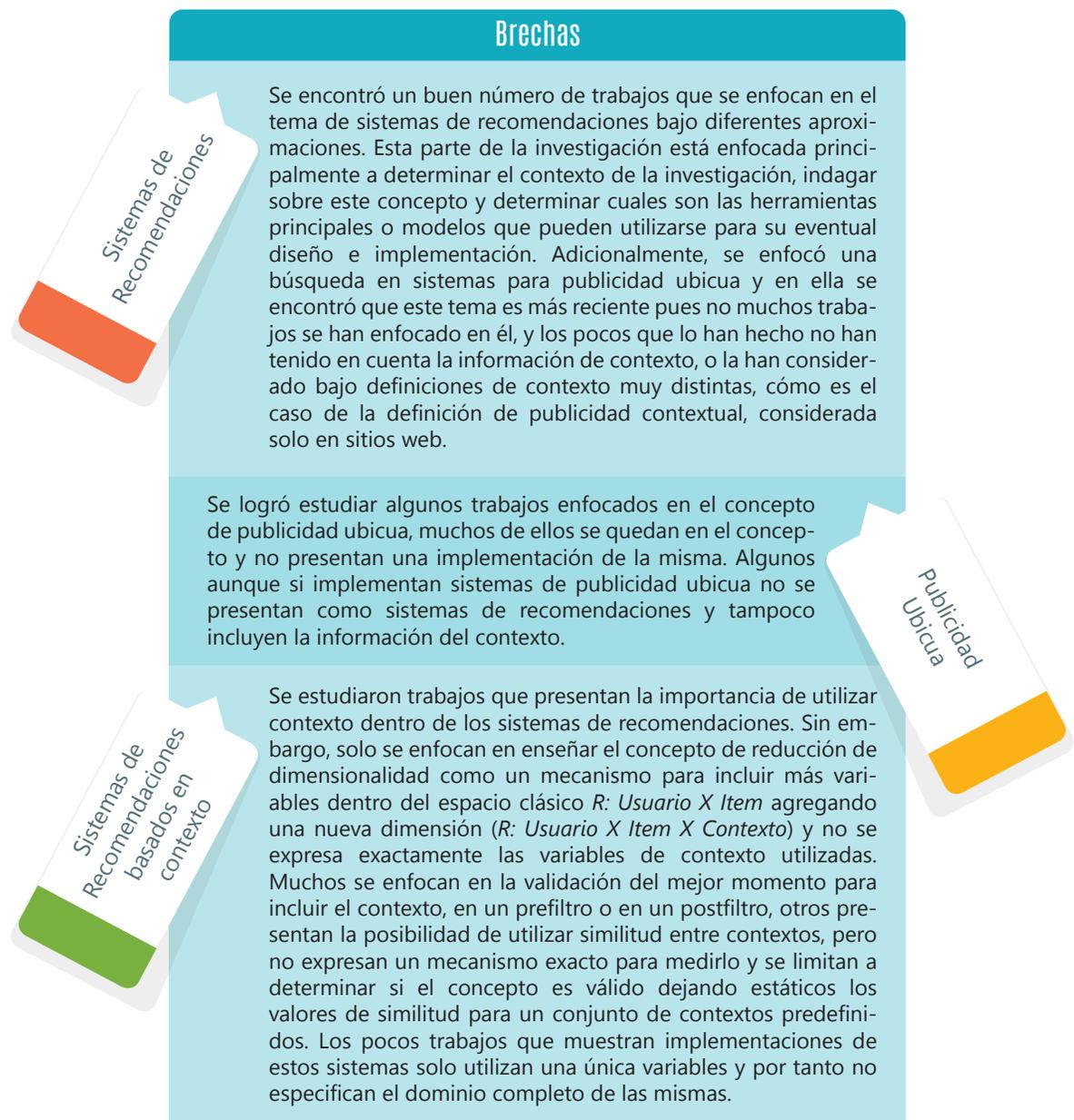


Tabla 2.1: Resumen de brechas encontradas Fuente: creación propia.



## Capítulo 3

# Contexto en entornos de publicidad ubicua y su implantación en Sistemas de Recomendaciones

Día a día las personas son rodeadas de información publicitaria, anuncios en la calle, en la televisión o en su bandeja de correo electrónico ofreciendo productos y servicios que en muchas ocasiones son atractivos y en muchas otras una molestia. Normalmente, este nivel de atracción depende de los gustos de las personas, pues constantemente se crean opiniones sobre cosas que nos gustan, nos disgustan o simplemente no nos interesan. Aunque esto pasa de forma inconsciente y los gustos varían mucho, siguen ciertos patrones; la gente tiende a sentirse atraída por elementos que son similares o están relacionadas con cosas que le han gustado antes; así mismo, individuos con gustos similares tienden a sentirse interesados por los mismos elementos. De esta manera, si eres alguien que odia el futbol no te interesará saber que hay un descuento en las entradas para ver jugar al Real Madrid contra el Barcelona en el Santiago Bernabeu; lo mismo puede pasar con la comida, las películas, los libros, etc. Los sistemas de recomendaciones intentan predecir estos patrones o modelos para seleccionar un listado de sugerencias que el usuario tal vez no conoce. Sin embargo, estos modelos pueden verse afectados por elementos externos, por ejemplo, a muchas personas les encanta el helado, pero lo piensan dos veces antes de comerse una “banana split” en una fría noche de invierno. Dichos elementos externos hacen parte del contexto de la recomendación y como lo muestra el ejemplo, pueden resultar supremamente importantes en el momento de predecir los gustos de los usuarios, y en el caso de la publicidad resulta ser aún más crítico, pues como se menciona en [17] la información del contexto del comprador es críticamente valiosa para los sistemas de recomendaciones en entornos comerciales, en tanto que ignorar datos como la posición del consumidor en la tienda, su objetivo de compra o cualquier otra variable contextual, pone al sistema de recomendaciones en un grave problema si se evalúa con una métrica mercantil.

Después de esta introducción sobre las recomendaciones y un leve vistazo sobre la importancia del contexto, en este capítulo se presentará una definición más completa del mismo. También se presentará el proceso de construcción de un modelo contextual donde se identificaron las variables más interesantes, a partir de trabajos previos, para un entorno de publicidad ubicua. Finalmente, se aportará un mecanismo de inclusión de dicho modelo contextual dentro de los sistemas de recomendaciones, que será evaluado de forma experimental en capítulos posteriores.

### 3.1. ¿Qué es Contexto?

El término contexto cuenta con diferentes definiciones acorde al conjunto de variables que en él están incluidas [80, 81, 18]. Sin embargo, para este trabajo sólo han sido consideradas dos de ellas. La primera viene de Dey y Abowd [81]. Según ellos, el contexto es “*Cualquiera información que pueda utilizarse para caracterizar la situación de una entidad. Donde la entidad puede ser cualquier persona, lugar u objeto importante en la interacción entre un usuario y una aplicación, incluyendo dicho usuario o aplicación*”, que permite inferir que el contexto básicamente está representado por cualquier situación, objeto o personas que se encuentren alrededor de una interacción entre un usuario y un aplicativo, y que dicho usuario o aplicativo también hacen parte del contexto. Aunque si bien esta definición se ajusta mucho a lo que se espera plantear como contexto en este trabajo, deja muy abiertos los límites y esto conlleva a que su posterior modelado no se encuentre lo suficientemente acotado, concediendo la utilización de muchas variables que pueden ser difíciles de medir y además generan demasiado ruido en los cálculos.

Por otro lado, Schilit et al [18] presentan la siguiente definición: “*El contexto se compone de tres aspectos importantes: dónde estás, con quién estás y con qué recursos cuentas*”. Esta definición está un poco más limitada, en ella se hace énfasis en tres aspectos importantes: **el dónde**, que puede entenderse como la posición geográfica en la cual se lleva a cabo la interacción entre el usuario y el aplicativo, y que además, dicha posición viene acompañada de todos los elementos físicos o del medio ambiente que caracterizan ese punto; **el con quién**, habla sobre la naturaleza social del individuo y por tanto, tiene en cuenta la información de personas que se encuentran al rededor del usuario; finalmente, **el con qué**, permite considerar las herramientas, objetos o dispositivos que acompañan al usuario y que además de ser fuentes de información, funcionan como puertas para medir y acceder a datos del usuario, esto es: smartphones, relojes inteligentes, o cualquier otro dispositivo móvil.

Dadas las características de la definición de contexto de Schilit, este trabajo se ha basado en ella para determinar el contexto de la recomendación. Pero más allá de ello, en el dominio de la publicidad es necesario determinar un conjunto más completo de variables que describan el entorno del usuario que recibe el anuncio y además que sean útiles en el momento de recomendarle. En ese sentido, la siguiente sección está dirigida a la adaptación de un modelo contextual existente para el caso específico de publicidad ubicua y para ello, inicialmente presenta la metodología utilizada y luego las adaptaciones.

### 3.2. Adaptación de un modelo contextual para el caso específico de publicidad ubicua

#### 3.2.1. Metodología para la definición de un modelo de contexto

Como ya se ha mencionado, el modelo utilizado para este trabajo parte de la definición dada por Schilit [18]. Sin embargo, la simple definición no es suficiente para modelar todo el espectro de posibles atributos que aportan información relevante a la recomendación y se requiere, por tanto,

### 3.2. Adaptación de un modelo contextual para el caso específico de publicidad ubicua

---

agregar variables que sean medibles y que puedan definir un contexto real. Para ello, se ha partido de la metodología expuesta por [5], que utiliza aproximaciones tanto Bottom-Up como Top-Down. La aproximación Top-Down consiste en una revisión de la literatura que permita identificar las diferentes partes y niveles del modelo, mientras que la aproximación Bottom-Up considera el tipo de información y la capacidad de ser medido. La metodología planteada se divide en dos partes importantes:

#### 1. Primera parte: Definición del modelo

Consiste en definir todas las variables que harán parte del modelo. Se parte de una aproximación Top-Down donde se toman los elementos más grandes o complejos del modelo y se subdividen en partes cada vez más pequeñas, hasta llegar a variables específicas. Para determinar los elementos principales se realiza un análisis de otros modelos existentes, luego a partir de una lluvia de ideas los elementos son desglosados o agrupados.

#### 2. Segunda parte: Verificación del modelo

Utiliza una metodología Bottom-Up en la que se verifica que las variables incluidas en el modelo sean medibles de alguna manera. Para realizar este paso, Bauer utilizó el concepto de expertos, quienes fueron invitados para dar su punto de vista, agregar, eliminar o reorganizar las variables planteadas como ellos lo considerasen más conveniente.

Cabe destacar que el trabajo de Bauer no solo aportó la metodología planteada, sino también un modelo como punto de partida. La siguiente sección presenta dicho modelo con más detalle y las posteriores adaptaciones que se realizaron para utilizarlo en un sistema de recomendaciones de publicidad ubicua.

#### 3.2.2. Adaptación del modelo para publicidad ubicua

Bauer y Spiekerman a partir de la metodología planteada en [5] construyeron un modelo contextual para ambientes de comercio ubicuo. Para su definición se basaron en el planteamiento de Schilit, y a partir de la revisión bibliográfica, dividieron el contexto en tres partes fundamentales:

1. **Ambiente del proveedor:** engloba elementos propios de quien presenta la publicidad y se conforma de la oferta y demanda de productos y servicios, y la campaña publicitaria.
2. **Ambiente físico:** representado por todo lo concerniente a la zona o lugar que rodea al usuario, y consta de la localización, las condiciones manipulables (ruido, luz, etc) y las condiciones no manipulables o atmosféricas (ej: clima).
3. **Ambiente del consumidor:** reúne elementos que describen al cliente y su interacción social, y en ello se incluyen: el perfil del cliente, su ambiente social y su proceso de compra, o en otras palabras su objetivo de compra.

Adicionalmente, el contexto ha sido dividido en tres niveles acorde a la granularidad de la variable, ellos son: nivel macro, que se refiere a valores muy globales y que si se usa la localización como ejemplo podría ser una ciudad o todo un país; nivel micro que es un poco más específico como un barrio dentro de la ciudad, y nivel situacional, que se enfoca en la escena como tal y usando el mismo ejemplo podría ser una tienda o la posición de una pantalla que despliega publicidad

### Capítulo 3. Contexto en entornos de publicidad ubicua y su implantación en Sistemas de Recomendaciones

dentro de la tienda. La figura 3.1 presenta un resumen del modelo. Como se ha mencionado, este esquema de contexto es el más completo encontrado en la literatura para entornos de publicidad. No obstante, para cumplir el objetivo de incluir esta información contextual dentro de los Sistemas de Recomendaciones, es necesario hacer ciertas adaptaciones que simplifiquen el modelo. Dichas adaptaciones dependen en gran medida del caso específico de publicidad ubicua para el cual esté dirigido el SR. En ese sentido, para determinar las variables finales utilizadas en este trabajo se ha puesto el foco en un caso de publicidad llamado “Digital Signage”, el cual consiste en el despliegue de anuncios utilizando pantallas electrónicas públicas y se ha popularizado en los últimos años como medio publicitario en aeropuertos, hospitales, centros comerciales, entre otros [82]. Cabe destacar que en comparación con la publicidad estática tradicional, como carteles o letreros, la publicidad a través de Digital Signage cuenta con la ventaja de poder actualizar oportunamente la información que se despliega en pantalla, y esta característica puede verse complementada y mejorada con la utilización de sistemas de recomendaciones que consideren el contexto del usuario.

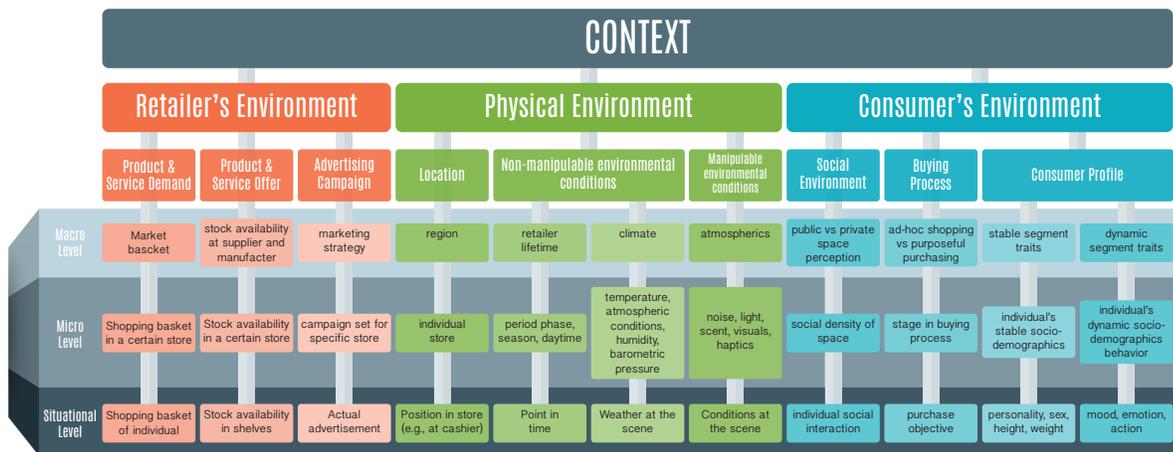


Figura 3.1: Modelo de contexto para publicidad Fuente: Adaptación de [5]

Partiendo del modelo anterior y usando una aproximación Top-Down, se tomaron las siguientes decisiones considerando la información recolectada en la literatura. En [50] se afirma que la localización de la pantalla es una variable críticamente importante en la publicidad pues entre más tráfico de personas, mayor atención atrae. Adicionalmente se hace énfasis en la importancia del contenido para la comunidad local, este par de afirmaciones se relacionan con el hecho de que la publicidad es en un gran porcentaje situacional. De hecho en [5] se afirma que el 75 % de las decisiones de compra se toman en el punto de venta o cerca de él. De esta manera, se ha decidido eliminar el nivel macro y micro del modelo contextual y se ha resuelto trabajar sólo con el nivel situacional. Por otro lado, una de las variables consideradas dentro del perfil del consumidor es llamada “proceso de compra”, esta variable fue incluida tomando como referencia [83]. Sin embargo, analizando con más detenimiento dicho trabajo se ha encontrado que una de las fases del proceso mencionado es “involvement” el cual se define como “lo que el proveedor hace para atraer la atención del cliente publicando en una pantalla o en un sitio web”, a partir de esto y considerando que la publicidad está en este punto del proceso, pues se utiliza precisamente para generar atracción hacia el cliente, se ha decidido eliminar esta variable del modelo.

### 3.3. Aproximación a la inclusión de contexto dentro de los sistemas de recomendaciones

[51] hace énfasis en que el tiempo que una persona pasa frente a una pantalla mientras se despliega un anuncio dice mucho del interés que esta persona tiene sobre el producto ofertado. De esta manera, para un sistema de recomendaciones de publicidad sería muy valioso el conocer que productos han atraído la atención del cliente y por este motivo se ha incluido el tiempo frente al anuncio publicitario como una variable a considerar en el modelo de contexto, ubicándola dentro de un grupo que ha sido denominado “Interacción con el sistema” dentro del ambiente del consumidor. De la misma forma, otros trabajos enfocados en publicidad han reforzado algunas de las variables incluidas en el modelo. [84] le ha dado importancia a la fecha y tiempo, localización y clima, mientras que [85] ha incluido la localización y clima como ambiente físico, y el género y la edad como ambiente del consumidor, también ha tenido en cuenta el estado de ánimo del cliente y para ello se enfoca en variables como su expresión facial, su distancia respecto al dispositivo y la dirección hacia donde enfoca su vista. [50] y [52] han considerado también el tiempo, edad, localización, estado de ánimo y ruido ambiente. De esta manera el modelo contextual resultante es el presentado en la Figura 3.2.

En cuanto a la aproximación Bottom-Up, la mayoría de variables consideradas siguen siendo parte del modelo original de Bauer, por tanto se confía que son perfectamente medibles. Por su parte, la variable de interacción con el sistema puede utilizar información del GPS del dispositivo móvil del usuario o análisis facial para determinar el tiempo que el usuario lleva observando el anuncio, de esta manera se considera que la variable es medible.

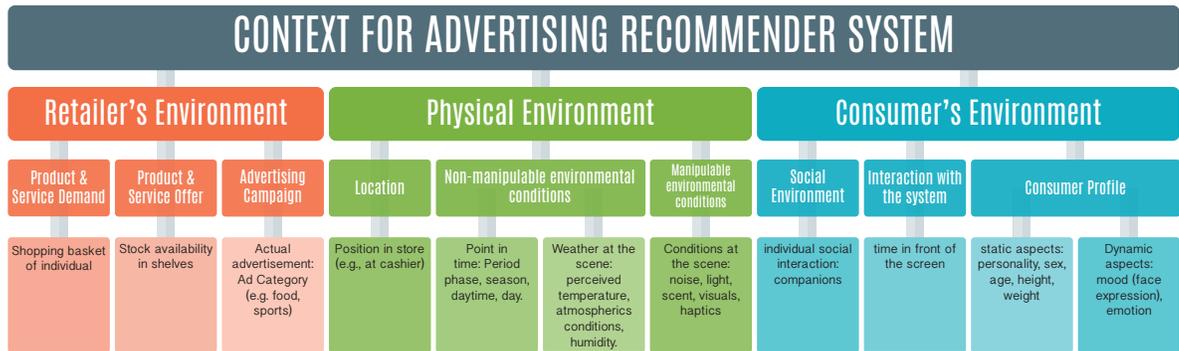


Figura 3.2: Modelo de contexto adaptado para SR de publicidad Fuente: creación propia basado en [5]

### 3.3. Aproximación a la inclusión de contexto dentro de los sistemas de recomendaciones

Una de las tareas de los sistemas de recomendaciones antes de hacer la recomendación es predecir el gusto de un usuario o usuarios por los ítems del sistema. Para ello, comúnmente están soportados en un espacio bi-dimensional conformado por Usuarios e ítems, y relacionados entre sí con las calificaciones o nivel de preferencia que cada usuario ha expresado por los ítems. Particularmente, el filtrado colaborativo realiza la tarea de predicción usando la información o ratings de otros usuarios del sistema, similares al usuario para el cual se desea recomendar. Esta información es traducida a una matriz de factorización como la mostrada en la figura 3.3, desde la cual el análisis de los datos

### Capítulo 3. Contexto en entornos de publicidad ubicua y su implantación en Sistemas de Recomendaciones

se hace más simple en el momento de realizar tareas propias del filtrado. Puesto que dicha matriz se encuentra construida a partir de las calificaciones que los usuarios han expresado por los ítems del sistema, normalmente se encuentra con espacios incompletos que representan ratings desconocidos, es aquí donde entra en juego la tarea de predicción, pues en ella se busca rellenar aquellos espacios vacíos a partir de análisis matemáticos.

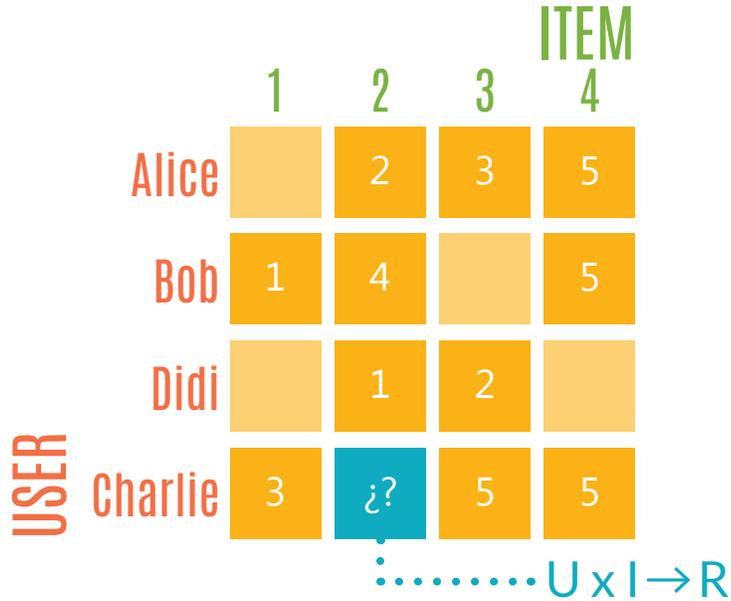


Figura 3.3: Matriz de factorización bidimensional Fuente: creación propia

Agregar una nueva variable al sistema, como en este caso el contexto, sugiere migrar del espacio bi-dimensional a un espacio multidimensional, donde además de tener en cuenta los usuarios y los ítems entran en juego nuevas variables que se convierten en nuevas dimensiones dentro de la matriz de factorización. La figura 3.4 muestra una matriz de factorización en la que además del espacio  $U \times I$  se ha tenido en cuenta la variable tiempo  $t$ .

Adomavicius [72] ha estudiado este problema y ha planteado un mecanismo de reducción de dimensionalidad, donde se parte de un SR multidimensional y se convierte en un SR bi-dimensional que puede ser tratado como filtrado colaborativo clásico. Dicha reducción de dimensionalidad puede llevarse a cabo de tres maneras, diferenciadas por el momento en que se realiza: la primera opción ha sido llamada **“prefiltrado”** y su nombre se debe a que la reducción de dimensión se lleva a cabo antes de hacer las recomendaciones. Esto es, hacer un filtrado previo al dataset acorde a las nuevas variables, y como resultado obtener un nuevo dataset que es utilizado para recomendar. El segundo enfoque es el **“postfiltrado”** y su funcionamiento es similar al prefiltrado, pero tiene lugar después de hacer la recomendación. De esta manera, el listado de ítems que son potencialmente recomendables son filtrados haciendo uso de las nuevas variables. Finalmente, otra opción es hacer una aproximación **“basada en modelo”**, que básicamente requiere una modificación a la ecuación que calcula las predicciones y de esta manera la reducción tiene lugar durante la recomendación. La

### 3.3. Aproximación a la inclusión de contexto dentro de los sistemas de recomendaciones

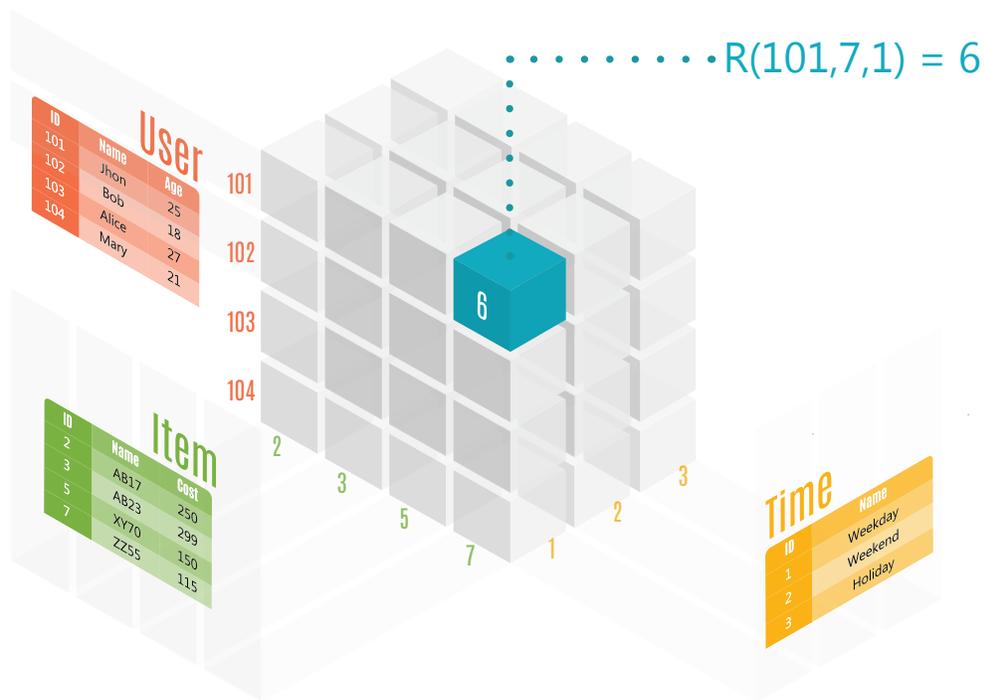


Figura 3.4: Matriz de factorización al agregar una nueva variable de contexto Fuente: Adaptación de [72]

figura 3.5 resume las aproximaciones planteadas.

El enfoque que se utilice para un SR específico depende en gran medida del dataset utilizado. Algunos autores se atreven a afirmar que cuando se trata de datasets muy pequeños no es muy recomendable utilizar un postfiltrado o prefiltrado, pues estos reducirían aún más el tamaño del dataset dando lugar a problemas de dispersión en los datos, y dificultando la posibilidad de encontrar ítems potencialmente recomendables. Durante el desarrollo de este trabajo se estudiaron las tres opciones con el objetivo de encontrar un mecanismo que se ajustará de la mejor manera las necesidades del proyecto.

#### 3.3.1. Prefiltrado y Postfiltrado

Como ya se ha mencionado, el prefiltrado y postfiltrado tienen lugar antes y después de hacer la predicción utilizando un RS clásico. Dado a que se trata de mecanismos similares, las herramientas utilizadas para realizar el proceso de reducción de dimensionalidad pueden usarse también en ambos casos. A continuación se listan cuatro opciones para dar paso a la reducción que fueron estudiadas en el desarrollo de este proyecto.

##### 1. Sistema de recomendaciones basado en contenido:

Los SR basados en contenido a diferencia de los filtrados colaborativos no se enfocan en las interacciones pasadas de los usuarios con el sistema sino en la información que conforma cada

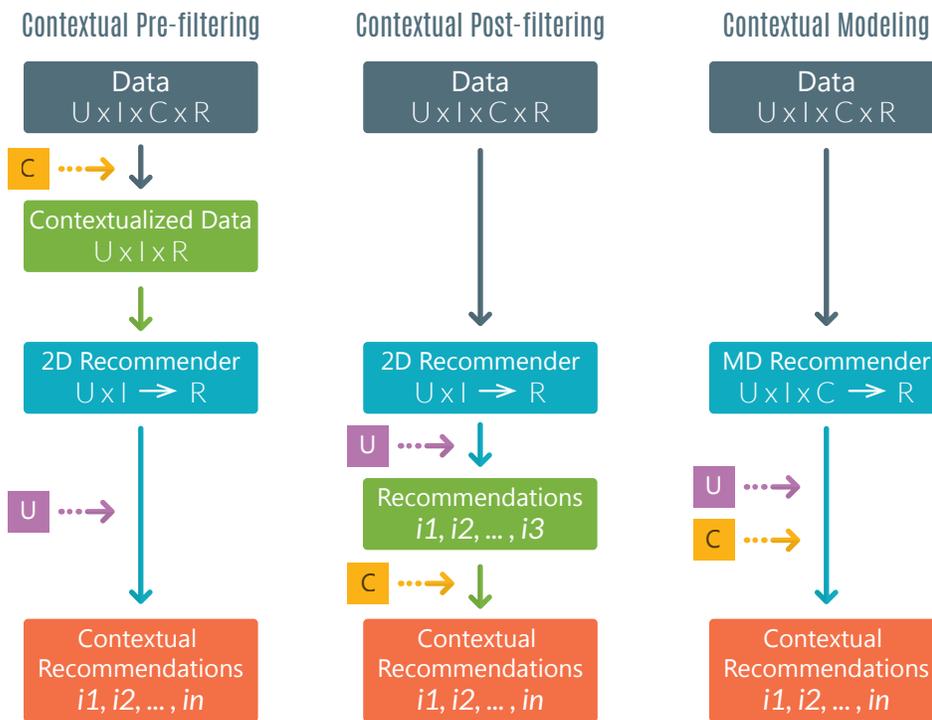


Figura 3.5: Paradigma para la inclusión de información contextual en SR clásicos Fuente: Adaptación de [72]

uno de los ítems y usuarios del mismo. De esta manera, mide la cercanía entre cada uno de estos elementos a partir de la cercanía entre sus características. Si se considera el contexto como una característica que puede ser intrínseca de los ítems o de los usuarios, es viable considerar que un SR basado en contenido podría aplicarse como filtrado inicial para determinar una vecindad de ítems para los cuales el contexto sea similar a aquel en el momento justo en el que se pretende hacer la recomendación. Sobra aclarar que dicha vecindad sería el dataset reducido que se utilizaría como entrada al mecanismo de recomendaciones utilizado en el caso de un pre-filtrado y sería el paso final de la recomendación en el caso del postfiltrado. [74] presenta un trabajo donde es utilizado este mecanismo como reducción de dimensionalidad combinado con recomendaciones basadas en conocimiento y procesos analíticos en línea (por sus iniciales en inglés: OLAP).

## 2. Razonamiento basado en casos CBR:

El razonamiento basado en casos es una aproximación utilizada en diferentes tareas de clasificación y resolución de problemas. Con el objetivo de resolver problemas nuevos este tipo de sistemas utiliza los problemas pasados que han sido resueltos y almacenados en una base de casos. Cada uno de dichos casos cuenta con una parte que describe el problema y otra que describe la solución. El proceso de resolución cuenta con cuatro pasos: el primero es la recuperación, equivalente a buscar todos los problemas similares a aquel al que se le desea dar solución. Utilizando para ello la descripción del problema almacenada junto con cada caso. Si alguno de los problemas hallados es el mismo o similar se pasa al siguiente paso; la reutilización, en el que dicho caso es duplicado temporalmente para determinar si resuelve el

### 3.3. Aproximación a la inclusión de contexto dentro de los sistemas de recomendaciones

---

nuevo problema y envuelve consigo una fase de adaptación; El tercer paso es la revisión, donde se evalúa si la solución propuesta eventualmente resuelve el problema planteado, para ello se pueden utilizar diferentes aproximaciones como por ejemplo el juicio de expertos; finalmente se pasa a la retención, donde se determina si el caso es algo nuevo y debe almacenarse o es completamente igual a otro existente y debe desecharse. [75] presenta un trabajo donde se ha utilizado esta aproximación como método de reducción de dimensionalidad. Por otra parte, el proyecto MyCBR [86] ha desarrollado una implementación open-source de algunas técnicas de Razonamiento basado en casos y se encuentra disponible para su descarga.

#### 3. Árboles de decisión aleatorios:

En [73] se presenta una aproximación para seleccionar la dimensión del contexto en la cual aplicar un filtrado colaborativo clásico a partir de un algoritmo de árboles de decisión aleatorios. El cual, es muy utilizado para crear árboles de decisión múltiples de forma aleatoria. La idea es particionar los ratings del dataset original y agruparlos en ratings con contextos similares. Cuando se construye cada árbol de decisión, en cada nivel se selecciona aleatoriamente una de las variables contextuales para particionar el data set. La partición se hace acorde a la variable seleccionada, si por ejemplo se selecciona el día, entonces se hace una partición para cada una de las opciones (lunes, martes, etc.). Cuando la partición se ha completado, se elimina la variable seleccionada de las variables posibles en el contexto con el objetivo de procesar una variable a la vez. A continuación, se toma cada una de las particiones y se repite el proceso hasta contar con un árbol que se ajuste a las necesidades. Otro trabajo que utiliza la misma aproximación se presenta en [87].

#### 4. ítems virtuales:

En [16] se presenta una aproximación para la reducción de dimensionalidad un poco diferente. En ella cada variable contextual es tratada como un nuevo ítem (ítems virtuales) en el dataset. Es decir que agrega una nueva fila y una nueva columna a la matriz de similitud para cada variable contextual y calcula la similitud correspondiente con los otros ítems del sistema.

Como se expuso con anterioridad, la decisión de utilizar un aproximación prefiltrado-postfiltrado o una basada en modelo puede depender en gran medida del dataset utilizado. Durante el desarrollo de este proyecto se tuvo una gran limitación referente a este tema y de hecho la sección 3.5 está enfocada a la construcción de un dataset con la información de contexto necesaria para los requerimientos propios del proyecto, pues no fue posible encontrar uno existente que se adaptara a nuestras necesidades. Ahora bien, trabajos como [78] han hecho una comparación entre pre y post-filtrado buscando determinar cual de los dos presenta mejores resultados y en qué circunstancias. Su conclusión no logra ser contundente y pone como variable importante el tamaño del dataset. Por su parte [88] incluye la aproximación basada en modelo en un experimento similar y demostró, al menos empíricamente, que esta aproximación genera, en la mayoría de los casos, mejores resultados cuando el RS es evaluado combinando las métricas de diversidad y precisión. Además su experimento también mostró que en muchos casos utilizar un enfoque sin contexto puede ser más efectivo que utilizar un pre o post-filtrado. Por otra parte, algunos experimentos simples que se ejecutaron utilizando árboles de decisión y un dataset que luego fue utilizado como

base para la creación de la fuente final de datos para pruebas, mostraron que no se obtenían resultados relevantes al incluir el contexto, bajo esta aproximación, en la precisión de las recomendaciones.

Por los motivos ya planteados se tomó la decisión de enfocarse en una inclusión de contexto bajo una aproximación basada en modelo.

### 3.3.2. Aproximación basada en modelo

El utilizar un enfoque basado en el modelo sugiere una complejidad mayor para el diseño del SR. A continuación se presentan las adaptaciones realizadas tanto al filtrado basado en usuario como al filtrado basado en ítem para poder conseguir el objetivo.

#### 3.3.2.1. Filtrado user-user

El filtrado user-user es un caso particular del filtrado colaborativo donde la predicción se hace a partir de usuarios similares a aquellos a los cuales desea recomendar. Para ello se confía en la métrica de similitud que se utiliza para generar una vecindad de usuarios parecidos al actual, y que han calificado previamente el ítem que se desea predecir. De esta manera, la predicción del rating que un usuario  $u$  dará a un ítem  $i$  se puede resumir como el promedio ponderado de todos los ratings que los usuarios similares a  $u$  han expresado para  $i$ . La siguiente es la ecuación de filtrado colaborativo basado en usuario.

$$p_{u,i} = \frac{\sum_{u' \in N} s(u', u) r_{u',i}}{\sum_{u' \in N} |s(u', u)|} \quad (3.1)$$

donde  $p_{u,i}$  equivale a la predicción del rating que el usuario  $u$  dará al ítem  $i$ .

$N$  se refiere a la vecindad de usuarios similares a  $u$ .

$s(u', u)$  es la similitud entre el usuario  $u$  y el usuario  $u'$ .

$r_{u,i}$  equivale al rating que el usuario  $u$  ha expresado para el ítem  $i$ .

A partir de ello se plantea una inclusión de contexto considerando además del rating que el usuario expresó para dicho ítem, el contexto en el momento en que expresó dicho rating. La siguiente es la ecuación adaptada.

$$p_{u,i} = \frac{\sum_{u' \in N} s(u', u) \eta(c, c_{r_{u',i}}) r_{u',i}}{\sum_{u' \in N} |s(u', u)| \left| \eta(c, c_{r_{u',i}}) \right|} \quad (3.2)$$

donde  $\eta(c, c')$  representa la similitud entre el contexto  $c$  y el contexto  $c'$ . Debe tenerse en cuenta que dado a que esta ecuación recorre un conjunto de usuarios y deja fijo el ítem, el contexto tenido en cuenta sólo puede considerar el ambiente del consumidor y el ambiente físico, pues el ambiente del proveedor se relaciona directamente con el ítem y no se cuenta con dos elementos diferentes para ser

### 3.3. Aproximación a la inclusión de contexto dentro de los sistemas de recomendaciones

---

comparados.

#### 3.3.2.2. Filtrado item-item

El filtrado item-item al igual que el user-user predice la preferencia de un usuario por cierto ítem a partir de la información previa contenida en el dataset, pero esta vez se buscan ítems similares a los que han sido relevantes para el usuario y se recomiendan aquellos cuya predicción sea mayor. La siguiente es la ecuación del filtrado colaborativo item-item:

$$p_{u,i} = \frac{\sum_{i' \in I} s(i', i) r_{u,i'}}{\sum_{i' \in I} |s(i', i)|} \quad (3.3)$$

donde  $p_{u,i}$  equivale a la predicción del rating que el usuario  $u$  dará al ítem  $i$ .  
 $s(i', i)$  es la similitud entre el ítem  $i$  y el ítem  $i'$ .  
 $r_{u,i'}$  equivale al rating que el usuario  $u$  ha expresado para el ítem  $i'$ .

Nuevamente se plantea una inclusión de contexto de la siguiente manera:

$$p_{u,i} = \frac{\sum_{i' \in I} s(i', i) \eta(c, c_{r_{u,i'}}) r_{u,i'}}{\sum_{i' \in I} |s(i', i)| \left| \eta(c, c_{r_{u,i'}}) \right|} \quad (3.4)$$

Donde al igual que en la anterior ecuación  $\eta(c, c_{r_{u,i'}})$  representa la similitud entre el contexto  $c$  y el contexto en el momento en que el usuario  $u$  calificó el ítem  $i'$ . Debe notarse que esta vez el contexto se encuentra enfocado en el ambiente del proveedor y físico, pues la variable de usuario se deja fija y por tanto no hay una manera de calcular similitudes en el ambiente del consumidor.

#### 3.3.2.3. Similitud de usuarios e ítems considerando el contexto

Las ecuaciones 3.2 y 3.4 representan la inclusión de contexto para los algoritmos de filtrado colaborativo user-user e item-item respectivamente. En ellas es posible notar que no todos los ambientes considerados en el contexto pueden tenerse en cuenta al tiempo. En la ecuación 3.2 el ítem se mantiene constante impidiendo que el ambiente del proveedor sea comparable con otro, mientras que en la ecuación 3.4 el usuario se mantiene constante y en este caso, es el ambiente del consumidor el que no tiene sentido compararse. Ahora bien, ambas ecuaciones parten de afectar el resultado de similitud ya sea entre usuarios o entre ítems con la similitud de los contextos. Sin embargo, hay que tener en cuenta que normalmente la similitud se mide considerando los ratings que los usuarios han expresado para un conjunto igual de ítems, de hecho, una de las métricas de similitud más populares es la correlación de Pearson que puede apreciarse en la ecuación 3.5.

$$s(u, v) = \frac{\sum_{i \in I_u \cap I_v} (r_{u,i} - \bar{r}_u) (r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_u \cap I_v} (r_{v,i} - \bar{r}_v)^2}} \quad (3.5)$$

donde  $s(u, v)$  representa la similitud entre el usuario  $u$  y el usuario  $v$ ,  $r_{u,i}$  y  $r_{v,i}$  representan la calificación dada por el usuario  $u$  y el usuario  $v$  al ítem  $i$  respectivamente, y  $\bar{r}_u$  y  $\bar{r}_v$  representan los valores medios de los ratings expresados por los usuarios  $u$  y  $v$ . Esta ecuación también es usada en recomendaciones basadas en ítems, pero en ese caso, se enfoca en la similitud entre los ítems utilizando un conjunto de usuarios quienes han expresado calificaciones para ambos ítems. Desde este punto de vista, si se tiene en cuenta el contexto de dichos ratings es posible que el calculo de similitud se vea afectado. Considérese la matriz de factorización de la Tabla 3.1 como ejemplo, donde además del rating se incluye el contexto en el momento de calificar ( $c_1$  y  $c_2$ ).

	A	B	C	D
Alice	4/ $c_2$	1/ $c_1$	3/ $c_1$	5/ $c_1$
Bob	3/ $c_1$	5/ $c_1$	4/ $c_1$	1/ $c_2$
Charlie	5/ $c_2$	2/ $c_1$	3/ $c_2$	4/ $c_2$

Tabla 3.1: Matriz de factorización de ejemplo Fuente: creación propia

La Tabla 3.2 presenta los valores de similitud calculados haciendo uso de la correlación de Pearson e indican un gran parecido entre los usuarios Alice y Charlie, a su vez una diferencia entre los gustos de Alice y Bob, e igualmente una diferencia entre Bob y Charlie. Estos resultados son consistentes con los valores de la tabla 3.1, pero ¿cuál es el resultado si se tienen en cuenta los valores de contexto?. Para este caso particular, se han considerado los contextos  $c_1$  y  $c_2$  completamente diferentes el uno del otro  $\eta(c_1, c_2) = 0$ .

	Alice	Bob	Charlie
Alice	1	-0,943	0,832
Bob		1	-0,680
Charlie			1

Tabla 3.2: Valores del coeficiente de correlación de Pearson Fuente: creación propia

### 3.4. Similitud de contextos

Ahora bien, aunque Alice y Charlie parecen ser muy similares según el coeficiente de correlación calculado, los contextos en los cuales se presentó su calificación son muy diferentes en muchos de los casos. Por tanto, sería lógico pensar que la similitud entre los usuarios debería considerar esta información. Para ello, le ha sido agregado el factor de similitud de contextos a la ecuación 3.5, y los valores de similitud han sido calculados nuevamente en la tabla 3.3 haciendo uso de la ecuación 3.6. Estos resultados muestran que aunque Alice y Charlie tienen cierto parecido, la similitud de los contextos afecta significativamente los resultados y su parecido se disminuye en la medida en que la calificación se da en un entorno completamente diferente.

$$s(u, v) = \frac{\sum_{i \in I_u \cap I_v} (r_{u,i} - \bar{r}_u) (r_{v,i} - \bar{r}_v) \eta(c_{r_{u,i}}, c_{r_{v,i}})}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_u \cap I_v} (r_{v,i} - \bar{r}_v)^2}} \quad (3.6)$$

	Alice	Bob	Charlie
Alice	1	-0,021	0,170
Bob		1	-0,567
Charlie			1

Tabla 3.3: Similitud entre usuarios considerando el contexto Fuente: creación propia

Nuevamente, esto puede ser visto también en un SR basado en ítems, puesto que la ecuación de correlación de Pearson es la misma y sólo se cambia el punto de vista. En este caso todo se analiza considerando los ítems y no los usuarios.

En esta sección se han presentado alternativas para incluir contexto dentro de los sistemas de recomendaciones y se ha dejado claro que utilizar una aproximación basada en modelo puede generar mejores resultados que usar algún filtrado ya sea antes o después de aplicar los algoritmos de recomendaciones clásicos. El utilizar esta aproximación basada en modelo sugiere la necesidad de incluir un factor extra a las ecuaciones de predicción de ratings, de tal forma que con él se tenga en cuenta la similitud entre los contextos. De esta manera, la siguiente sección se enfoca en presentar un mecanismo para medir la similitud entre contextos, es decir, calcular el valor de  $\eta$  (eta), y con ello completar las ecuaciones planteadas.

### 3.4. Similitud de contextos

Las métricas de similitud son una parte importante de los sistemas de recomendaciones, pues estos confían su funcionamiento en los componentes encargados de medir la semejanza entre los elementos involucrados en la recomendación. Los sistemas basados en usuarios por ejemplo, requieren encontrar los usuarios más parecidos a aquel a quien se pretende entregar una recomendación, de la misma

### Capítulo 3. Contexto en entornos de publicidad ubicua y su implantación en Sistemas de Recomendaciones

forma aquellos que se soportan en los ítems primero deben encontrar los ítems más parecidos a aquellos que han sido relevantes para el usuario [89]. A partir de esto es posible inferir que las diferentes métricas de similitud son aplicables a cualquier tipo de objeto, así como los RSs pueden usarse para recomendar cualquier tipo de ítem. Algunas de las métricas de similitud más utilizadas son la correlación de Pearson, la correlación de Spearman, la distancia Euclidiana y la medida de cosenos; las dos primeras hacen referencia a correlaciones, por tanto se enfocan en medir la similitud de dos objetos respecto a sus comportamientos, y las dos siguientes son medidas de distancia y evalúan los objetos a partir de la cercanía entre cada una de las partes que lo componen. Las medidas de correlación tienen una ventaja y es que pueden ser aplicadas indistintamente a cualquier objeto y funcionarán sin hacer cambio alguno. No obstante, determinar la correlación entre dos elementos sugiere que se realicen distintas iteraciones en el cálculo, requiriendo un monto importante de procesamiento. Las medidas de distancia suelen depender de los elementos que componen los objetos y por tanto para cada objeto distinto que se compare se requiere una implementación propia. Pese a esto, una vez se cuenta con la información necesaria, el monto de procesamiento requerido para calcular la distancia entre dos objetos es menor que el requerido para determinar su correlación. En el caso particular de las recomendaciones, las medidas de correlación tienen otra desventaja, y es que para poder medirla es necesario contar con datos de entrada que puedan indicar un comportamiento y sea comparable con otros elementos en el sistema, y en muchos casos cuando el elemento apenas empieza a existir se imposibilita el cálculo de su similitud. Este problema comúnmente se le conoce como “arranque en frío”.

El arranque en frío es un problema tan común de los SR tanto basados en usuario como basados en ítem, que trabajos como [90] se han enfocado a presentar formas de bordearlo. Eventualmente, el agregar una nueva variable como en este caso el contexto, puede aumentar los casos en los que este problema se presenta. Pues contextos con nuevas configuraciones o características diferentes son considerados como elementos nuevos imposibles de ser comparados con una métrica de correlación. Este es el principal motivo por el cual para este trabajo se ha decidido trabajar con una métrica de distancia para medir la similitud entre los contextos, siendo en este caso particular la medida de Distancia Euclidiana, dada su popularidad.

Ahora, para medir la distancia euclidiana supóngase  $A$  y  $B$  dos elementos de contexto conformados por los atributos  $\{a_1, a_2, \dots, a_n\}$  y  $\{b_1, b_2, \dots, b_n\}$ , la distancia euclidiana de este par de elementos se encuentra expresada por:

$$D(A, B) = \sqrt{\sum_{i=1}^n d_i^2} \quad (3.7)$$

Donde  $d_i$  es equivalente a la diferencia entre los atributos  $a_i$  y  $b_i$ , es decir,  $d_i = a_i - b_i$ . Ahora bien, es posible que no todas las características de los objetos tengan el mismo grado de importancia en el momento de medir la similitud, por ejemplo, puede darse el caso de un contexto donde interese más la variable de temperatura que la hora del día, por motivos propios del sistema, como que los anuncios están ligados a ítems que son interesantes en momentos calurosos. Para este caso, se ha incluido un tercer vector de variables llamado  $W = \{w_1, w_2, \dots, w_3\}$ , el cual está conformado por todos los pesos de cada una de las variables. Así la ecuación de distancia incluyendo dichos pesos es:

### 3.4. Similitud de contextos

---

$$D(A, B) = \sqrt{\sum_{i=1}^n w_i d_i^2} \quad (3.8)$$

Finalmente, tal como se encuentra expresada la ecuación no es posible considerarla como una medida de similitud, por una razón simple. Las medidas de similitud generalmente se encuentran en el rango  $[0, 1]$  donde 1 significa que los elementos son completamente iguales y 0 completamente diferentes. Por el contrario las medidas de distancia se encuentran en  $[0, \max(a_i, b_i)]$  donde 0 significa que no hay distancia entre los objetos y por tanto son completamente iguales. Para corregir ello se ha recurrido a la siguiente transformación:

$$\eta(A, B) = \frac{1}{1 + \sqrt{\sum_{i=1}^n w_i d_i^2}} \quad (3.9)$$

Que puede ser fácilmente utilizada como ecuación para un cálculo de similitud.

#### 3.4.1. Similitud de contextos incluyendo la similitud de los usuarios o ítems

Se ha visto que el incluir el contexto dentro de la similitud puede cambiar completamente el resultado del parecido de dos usuarios o dos ítems. Pero también se ha visto que la similitud entre contextos se mide mediante la combinación de distintas variables. Ahora, ¿qué pasa si se considera la similitud entre los usuarios como una variable más del contexto? por ejemplo, en vez de utilizar la similitud de las variables del consumidor o el proveedor, se considera la similitud entre los ítems o usuarios midiéndolas como una correlación. La ecuación de similitud de contextos podría extenderse a lo siguiente:

$$\eta(A, B) = \frac{1}{1 + \sqrt{w_s d_{s(u,v)}^2 + \sum_{i=1}^n w_i d_i^2}} \quad (3.10)$$

Donde  $d_{s(u,v)}$  es la distancia equivalente a la similitud entre usuarios calculada con una correlación de Pearson o alguna otra.

Hasta este punto ya está claro cuáles son las variables que puede contener un elemento de contexto para publicidad y un mecanismo para medir su similitud. Sin embargo, no está claro como medir la distancia entre cada una de las variables del contexto. Para ello en [63] se hizo una clasificación de las variables de contexto en tres tipos: escalares, ordinales y categóricas, aunque en dicho trabajo la clasificación se realiza directamente al contexto, es válido pensar que se enfoca más en los atributos del mismo y adicionalmente, considerando el modelo de contexto planteado, se ha agregado un cuarto tipo llamado variables de decisión. Las variables de decisión sólo requieren ser revisadas por su cumplimiento, es decir, son completamente iguales o completamente diferentes, por ejemplo, existen o no elementos en stock; la distancia entre variables escalares y ordinales comúnmente se

mide mediante su diferencia aritmética, las variables categóricas por su parte, son un poco más complejas y comúnmente requieren del uso de ontologías de dominio o algún otro mecanismo de análisis semántico para calcular su distancia.

a lo largo de las secciones 3.3 y 3.4 se han presentado mecanismos para incluir contexto dentro de los sistemas de recomendaciones mediante la adaptación de ecuaciones clásicas. Asimismo, se ha presentado la distancia euclidiana como punto de partida para determinar la similitud entre un par de contextos. Sin embargo, para poder hacer una evaluación de lo planteado es necesario contar con una fuente de información que permita aplicar las ecuaciones y sacar conclusiones, y por dicha razón, la siguiente sección se enfoca en la construcción experimental de un dataset que contenga información contextual que se ajuste a las necesidades propias de este caso.

### **3.5. Construcción experimental de un Dataset para publicidad ubicua usando información contextual**

Generalmente, los sistemas de recomendaciones se componen de tres elementos: el primero es la fuente de datos conocida como “Dataset”, el segundo es el encargado de hacer medidas de similitud y el tercero es el responsable de hacer la recomendación usando para ello los elementos anteriores. Considerando esto, uno de los limitantes para la construcción y evaluación de un SR es encontrar datasets que se ajusten a las necesidades propias del sistema. Este trabajo particular requiere de unas características propias de la fuente de datos que no son fáciles de encontrar. Entre ellas tal vez la más importante es que contenga información de contexto (adaptada al modelo planteado) en el momento de expresar calificaciones. Adicionalmente se requiere que los ítems cuenten con sus datos de descripción, títulos y fotos, pues son necesarios para hacer pruebas del sistema con usuarios reales; y finalmente se requiere contar con una información base de los usuarios, como su género o edad, para alimentar la información de contexto. Dadas estas condiciones, no fue posible encontrar una fuente que incluyese dicha información y por tanto, fue necesario crear un experimento para recolectarla y construir con ella el dataset. Cabe destacar que entre los datasets revisados antes de tomar esta decisión se encuentran Epinions [91] que contiene datos de publicidad incluyendo descripción de los ítems pero no cuenta con información de contexto, nopcommerce [92] que es un aplicativo web para ecommerce y contiene datos de ítems pero no cuenta con calificaciones y mucho menos con contexto, y otros obtenidos de Amazon. Para el experimento se tomó como base un dataset construido en el proyecto SMARTA [93] el cual incluye anuncios publicitarios y calificaciones de varios usuarios para dichos anuncios. Cabe aclarar que estas calificaciones, al igual que en los otros datasets, no cuentan con la información del contexto en el momento de recomendar y por tanto de esta fuente solo se tomó la información de los ítems.

#### **3.5.1. Diseño del experimento**

A continuación se listan una serie de decisiones para el experimento:

1. Es necesario que el experimento pueda ser accesado en línea por varios usuarios al tiempo. Por tanto, se construyó una aplicación web donde varios usuarios pueden acceder y calificar anuncios.

### 3.5. Construcción experimental de un Dataset para publicidad ubicua usando información contextual

- No todas las variables contextuales incluidas en el modelo son fácilmente medibles desde un aplicativo sencillo. De esta manera, se seleccionaron algunas variables contextuales que fueran fácilmente medibles o que el usuario pudiera indicar de forma explícita en el aplicativo web. Cabe destacar que una regla importante para seleccionar las variables fue que todos los ambientes incluidos en el modelo aparecieran con al menos un atributo dentro de los seleccionados para el experimento. Estas variables se encuentran consignadas en la Figura 3.6.

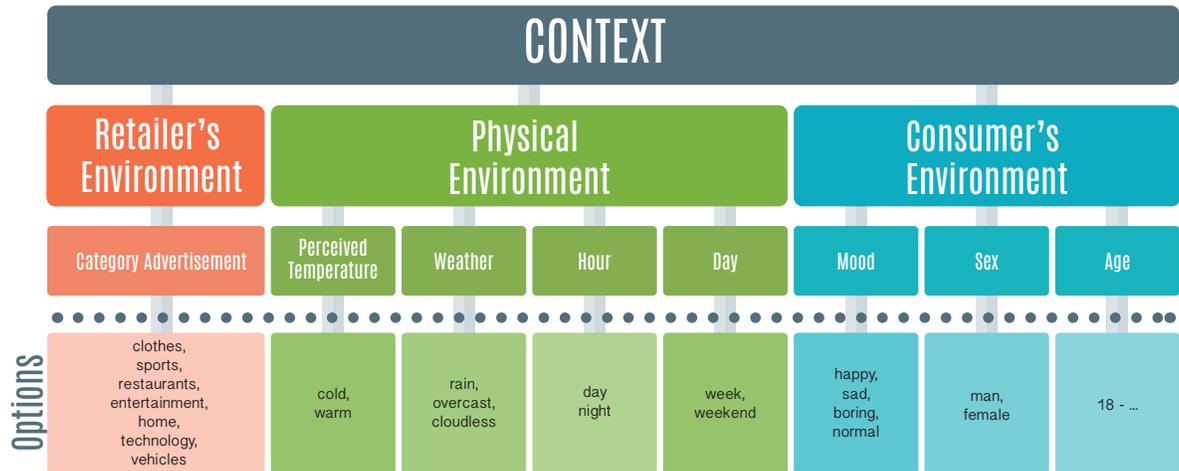


Figura 3.6: Variables de contexto seleccionadas Fuente: creación propia

- Dado a que en el momento de probar el SR se requieren los pesos de cada una de las variables de contexto (para poder calcular su similitud), se hizo necesario que el aplicativo permitiera tener un indicio de estos valores. Para ello se decidió solicitar al usuario la razón primordial por la cual expresó la calificación cada vez que realice este proceso. Al final, el peso de la variable sería equivalente a la probabilidad de que dicha variable sea seleccionada como razón primordial.

Finalmente, el diagrama de casos de uso presentado en la figura 3.7 resume todas las acciones que un usuario voluntario puede hacer con el aplicativo.

#### 3.5.2. Implementación y ejecución del experimento

Con todas las decisiones tomadas el siguiente paso consistió en construir el aplicativo Web planteado. Cabe destacar que se desarrolló utilizando el gestor de base de datos MySQL para albergar toda la información del dataset, JSEE para construir el aplicativo y DAO<sup>1</sup> para la conexión con la base de datos; la capa de presentación se desarrolló utilizando JSF 2.0<sup>2</sup> y el framework primefaces. Estas

<sup>1</sup>DAO: Data Access Object (Objeto de acceso a datos) es un patrón de diseño que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos, entre los que pueden incluirse bases de datos, archivos de texto plano, etc.[94]

<sup>2</sup>JSF: Java Server Faces es una tecnología y framework para aplicaciones JAVA basadas en Web que simplifica el desarrollo de interfaces de usuario [95]

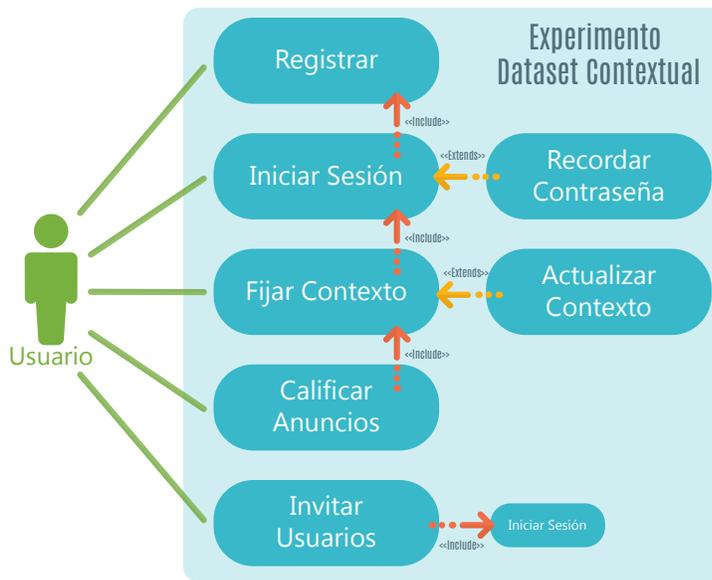


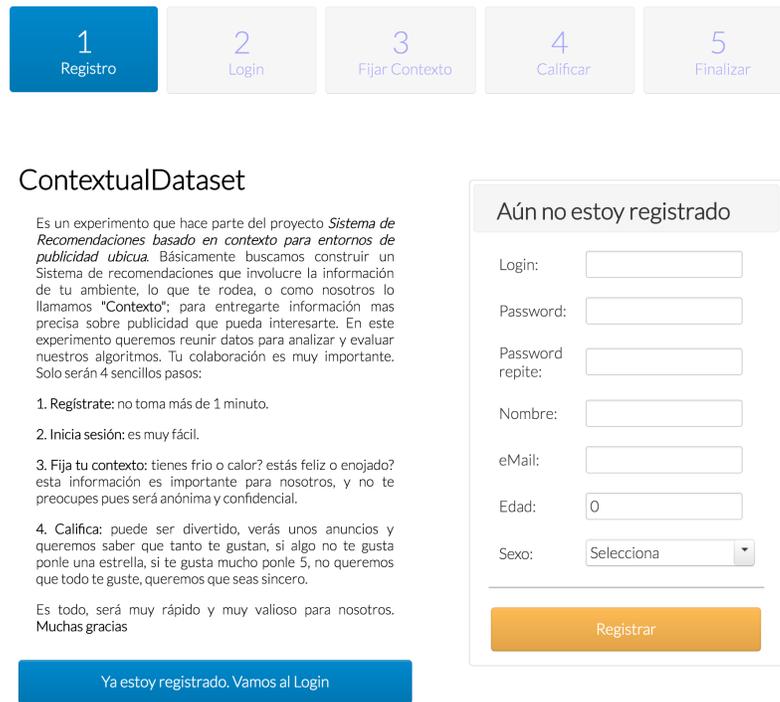
Figura 3.7: Diagrama de casos de uso definido para el experimento Fuente: creación propia

mismas herramientas y tecnologías fueron utilizadas posteriormente para construir la plataforma que permitiría hacer pruebas online y por este motivo se encuentran detalladas en el capítulo 5. En cuanto a la base de datos, se diseñó de forma que pudiera ser utilizada posteriormente en el desarrollo del SR para pruebas online, por esta razón, se encuentra descrita en el capítulo 4 sección 4.4.1.3.

A partir del diagrama de casos de uso enseñado en la figura 3.7 se diseñaron las siguientes interfaces:

- **Registro:** Permite solicitar información del usuario como su género y edad. Adicionalmente, la interfaz permite crear un usuario y contraseña para permitir al usuario realizar una futura prueba online y presenta una introducción donde explica brevemente en qué consiste el experimento. La imagen de la figura 3.8 presenta la interfaz de registro.
- **Login:** Permite iniciar sesión a un usuario que ya ha sido registrado previamente. En la figura 3.9 se encuentra evidenciada esta interfaz. Para el inicio de sesión sólo se requieren el login y la contraseña que fueron usados durante el registro. En caso de haber olvidado estos datos, el usuario puede utilizar la función “*olvidé mi contraseña*” para recibir los datos por correo electrónico. Si el usuario no se ha registrado previamente, puede usar la opción “*No estoy registrado*” para regresar a la interfaz de registro.
- **Fijar contexto:** Esta interfaz permite inicializar variables del ambiente físico como el día, hora, clima y temperatura percibida; y del ambiente del consumidor como su estado de ánimo. Una vez las variables se han fijado, se presenta una ventana solicitando confirmación. Las imágenes de la figura 3.10 presentan la interfaz.

### 3.5. Construcción experimental de un Dataset para publicidad ubicua usando información contextual



The registration process is shown in a five-step flow: 1. Registro (highlighted), 2. Login, 3. Fijar Contexto, 4. Calificar, and 5. Finalizar.

#### ContextualDataset

Es un experimento que hace parte del proyecto *Sistema de Recomendaciones basado en contexto para entornos de publicidad ubicua*. Básicamente buscamos construir un Sistema de recomendaciones que involucre la información de tu ambiente, lo que te rodea, o como nosotros lo llamamos "Contexto"; para entregarte información mas precisa sobre publicidad que pueda interesarte. En este experimento queremos reunir datos para analizar y evaluar nuestros algoritmos. Tu colaboración es muy importante. Solo serán 4 sencillos pasos:

1. Regístrate: no toma más de 1 minuto.
2. Inicia sesión: es muy fácil.
3. Fija tu contexto: tienes frío o calor? estás feliz o enojado? esta información es importante para nosotros, y no te preocupes pues será anónima y confidencial.
4. Califica: puede ser divertido, verás unos anuncios y queremos saber que tanto te gustan, si algo no te gusta ponle una estrella, si te gusta mucho ponle 5, no queremos que todo te guste, queremos que seas sincero.

Es todo, será muy rápido y muy valioso para nosotros. Muchas gracias

[Ya estoy registrado. Vamos al Login](#)

#### Aún no estoy registrado

Login:

Password:

Password repite:

Nombre:

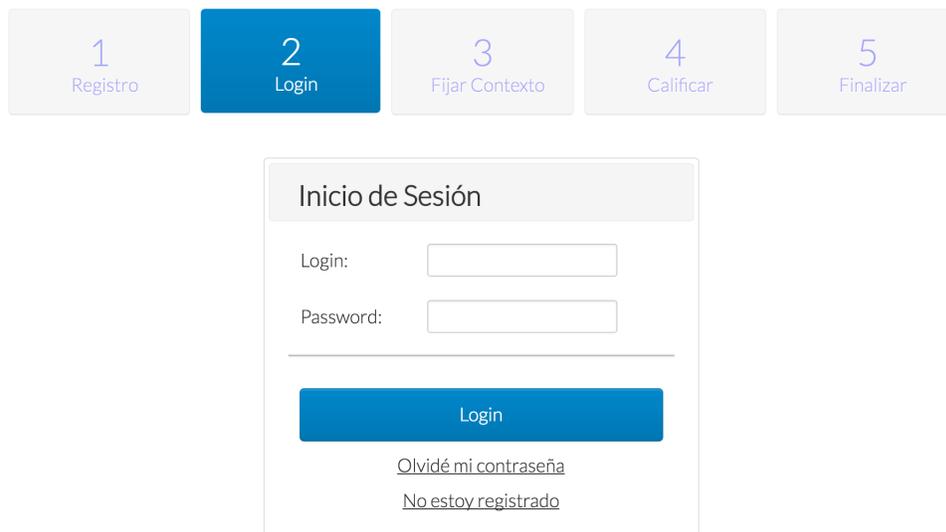
eMail:

Edad:

Sexo:

[Registrar](#)

Figura 3.8: Interfaz de registro para experimento dataset contextual Fuente: creación propia



The login process is shown in a five-step flow: 1. Registro, 2. Login (highlighted), 3. Fijar Contexto, 4. Calificar, and 5. Finalizar.

#### Inicio de Sesión

Login:

Password:

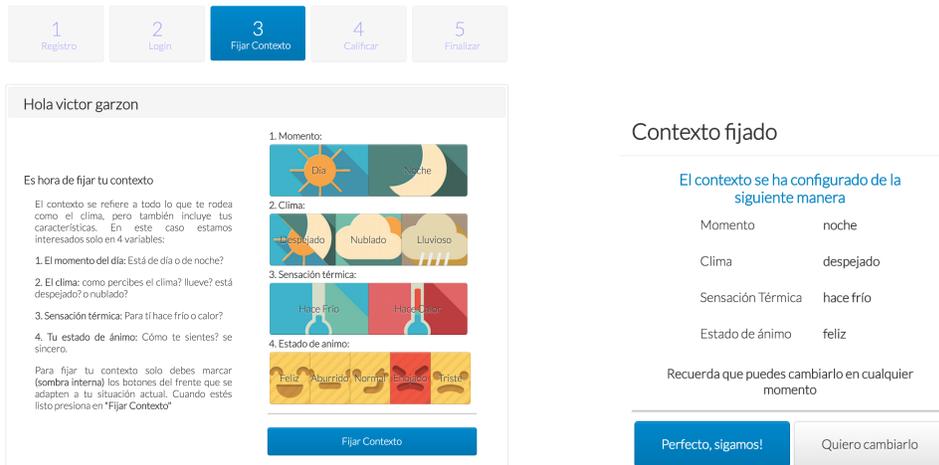
[Login](#)

[Olvidé mi contraseña](#)

[No estoy registrado](#)

Figura 3.9: Interfaz de login para experimento dataset contextual Fuente: creación propia

## Capítulo 3. Contexto en entornos de publicidad ubicua y su implantación en Sistemas de Recomendaciones

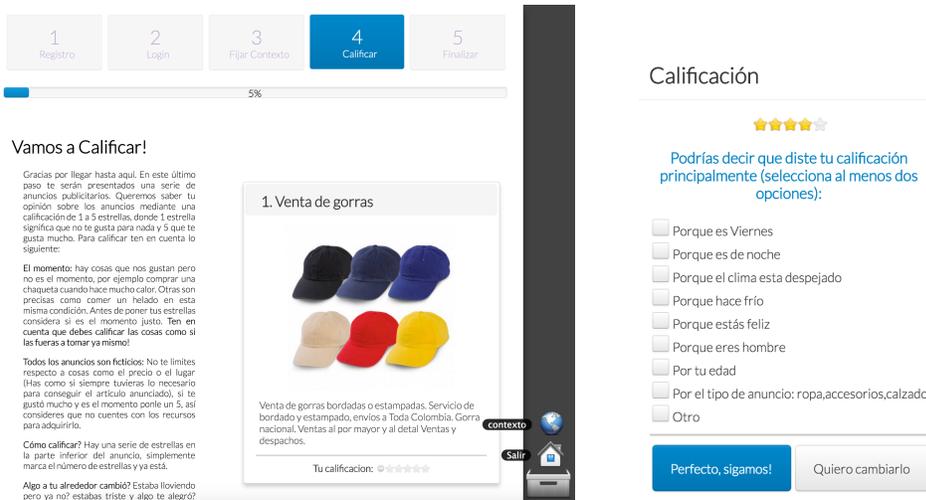


(a) Selección de variables contextuales

(b) Confirmación de variables contextuales

Figura 3.10: Interfaz de configuración del contexto para experimento dataset contextual Fuente: creación propia

- Calificar:** Las imágenes de la figura 3.11 presentan la interfaz de calificación de anuncios. En esta interfaz se resumen algunas reglas que el usuario debe tener en cuenta al calificar los anuncios. Durante el proceso de calificación se entregan 20 anuncios para que el usuario exprese su rating y se muestra el progreso a través de una barra en la parte superior. Para calificar, el usuario debe utilizar las estrellas que se encuentran debajo de la imagen del anuncio y una vez ha seleccionado la calificación, se presenta un cuadro de diálogo solicitando la razón por la cual dio dicha calificación. Sobra decir que se espera que esta razón esté relacionada con el contexto. Finalmente en la parte inferior derecha se encuentran los botones “contexto” y “salir”, que permiten reconfigurar el contexto y salir del experimento respectivamente.



(a) Calificación del anuncio

(b) Selección de razones para la calificación

Figura 3.11: Interfaz de calificación del anuncio para experimento dataset contextual Fuente: creación propia

### 3.5. Construcción experimental de un Dataset para publicidad ubicua usando información contextual

- **Referir:** Las imágenes de la figura 3.12 presentan el menú final del experimento. Las opciones que se tienen son “*calificar más anuncios*” donde se le presentarán 20 anuncios nuevos, “*Recomendar a alguien*” donde se podrá referir a nuevos usuarios para el experimento y de esta manera ayudar a que muchas más personas participen. Estos usuarios que son referidos recibirán un correo electrónico con la invitación. La ultima opción en el menú es “*salir*” que terminará la sesión en el aplicativo.



(a) Menú final



(b) Referir a otro usuario

Figura 3.12: Interfaz de finalización para expermineto dataset contextual Fuente: creación propia

Con las interfaces implementadas el aplicativo está listo para ser puesto en producción. Sin embargo, es importante mencionar que antes de tener las interfaces en el estado presentado en las figuras previas y poner en producción la aplicación para empezar a recolectar información, fue necesario realizar cuatro iteraciones del experimento en las que se analizó la forma como los usuarios interactuaban con el aplicativo, la calidad de la información recolectada para una posterior recomendación, y claridad que el experimento tenía para con el usuario.

La primera iteración se realizó con cuatro usuarios buscando determinar si era entendible lo que se esperaba hacer con el experimento. Como resultado se reportaron errores relacionados principalmente con la interfaz gráfica de la plataforma. La figura 3.13 muestra los errores de las interfaces en dicho estado del experimento. A continuación, en la segunda iteración se resolvieron los problemas reportados y se analizaron los datos recolectados previamente. Esto mostró que en la mayoría de los casos la razón de las calificaciones era la categoría del ítem, lo que es un comportamiento indeseado pues se espera tener una participación de todas las variables contextuales seleccionadas. En la tercera iteración se incluyeron cerca de 30 nuevos ítems que se consideraban más atractivos en ciertos contextos que en otros, algunos ejemplos son: publicidad de bares, bebidas calientes, bebidas frías, centros recreativos, entre otros. Esto mejoró mucho la cantidad de razones seleccionadas por los usuarios. Adicionalmente, por sugerencia de algunas personas, el seleccionar una sola razón no era suficiente para expresar el porqué de la calificación, además esto seguía siendo un problema, pues en muchos casos los usuarios simplemente escogían la categoría por facilidad y por hacer el experimento rápido. Finalmente, para resolver este problema, en la última iteración se permitió escoger de dos razones en adelante para las calificaciones. Con este último cambio el experimento mostró mejores resultados y era mucho más claro para los usuarios. De esta manera, esta versión el aplicativo fue puesta en producción y se invitaron usuarios, hombres o mujeres mayores de 18 años, para que

Característica	Valor
Usuarios	98
Ítems Activos	130
Ítems Totales	170
Contextos	57
Calificaciones	1794

Tabla 3.4: Información del dataset resultante Fuente: creación propia

realizaran el experimento. Las invitaciones se hicieron a través de redes sociales y también utilizando la fase “referir” donde los mismos usuarios remitían a nuevos.



(a) Error en página de registro

(b) Error en página de configuración del contexto

Figura 3.13: Errores en Interfaces Gráficas iniciales de Experimento de construcción del dataset Fuente: creación propia

### 3.5.3. Dataset resultante

Después de 3 semanas de recolección de información se dio por terminado el experimento contando con la información consignada en la tabla 3.4.

### 3.5.4. Cálculo de pesos para las variables de contexto

A partir de la información recolectada fue posible obtener un conjunto de pesos para cada variable de contexto. Para ello se tomó como referencia el número de veces que un usuario expresaba como razón principal de su calificación cada variable. Dado a que era posible seleccionar más de una razón se tuvo un total de 3331 de las cuales 928 (28 %) correspondieron al ambiente físico, 1344 (40 %) al ambiente del consumidor y 1059 (32 %) al ambiente del proveedor. La figura 3.14 muestra el peso porcentual de cada uno de los ambientes en la información de contexto y a su vez los gráficos de la figura 3.15 presentan los pesos de cada variable.

### 3.5. Construcción experimental de un Dataset para publicidad ubicua usando información contextual

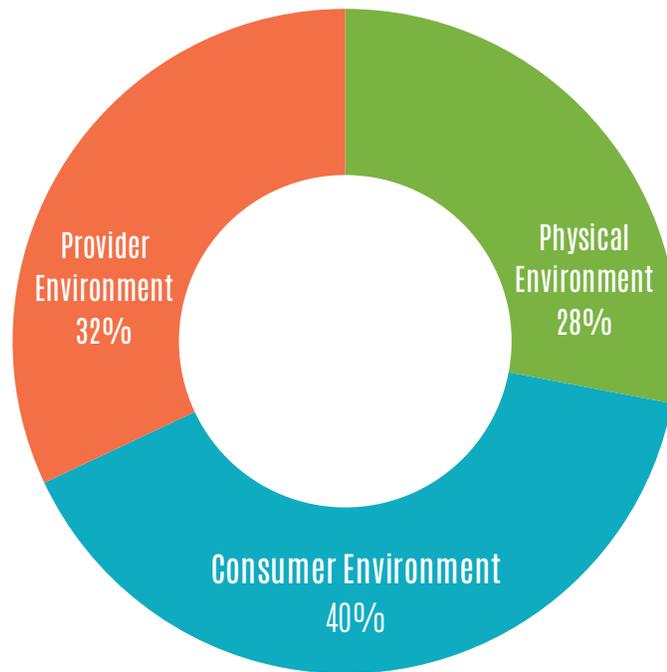


Figura 3.14: Pesos porcentuales de cada uno de los ambientes Fuente: creación propia

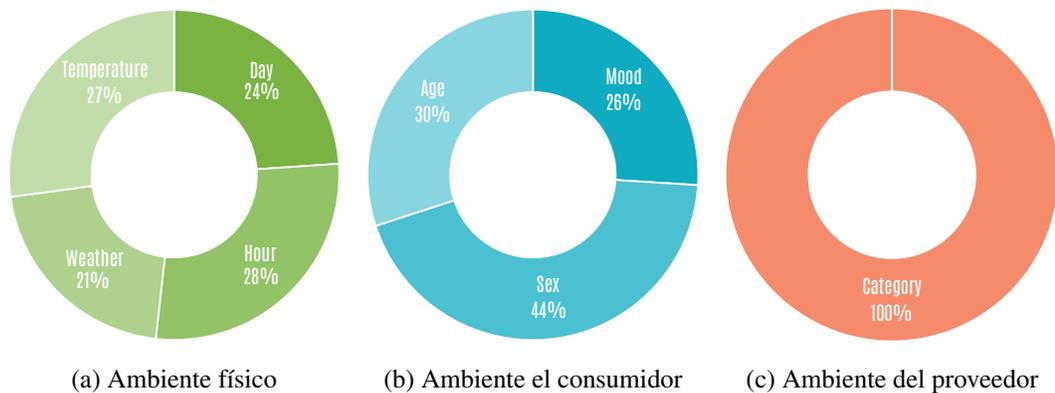


Figura 3.15: Pesos porcentuales de cada una de las variables del contexto separadas en ambientes Fuente: creación propia

A partir de los resultados se puede concluir que existe una gran importancia en las variables de contexto seleccionadas. Aunque los tres ambientes tienen una participación significativa al realizar las calificaciones, la variable de categoría al ser la única presente en el ambiente del proveedor cuenta con un porcentaje bastante alto en comparación con las otras variables individuales. El dataset resultante de este experimento se encuentra disponible en línea para ser utilizado en proyectos relacionados.

### 3.5.5. Cálculo de distancias entre los posibles valores de cada variable contextual

Para calcular la distancia se partió nuevamente de una distancia euclidiana entre los ratings expresados cuando cada variable se hacía presente. A continuación se mostrará paso a paso el cálculo para una variable particular teniendo en cuenta que las demás se calculan de forma similar.

Suponga que se quiere determinar la distancia entre las variable de hora: día y noche. Los pasos para hacerlo son los siguientes:

1. Se seleccionan todos los ratings para los cuales los usuarios expresaron la hora como razón de la calificación.
2. Se separa este subconjunto de ratings entre aquellos que se expresaron de día y aquellos que se expresaron de noche. Y se agrupan con el id del ítem y el promedio de todos los ratings expresados para dicho ítem dentro de cada subconjunto.
3. Se seleccionan los ítems que aparecen en ambos subconjuntos.
4. Se aplica una distancia euclidiana a partir de la ecuación 3.7. Donde  $d_i$  es la diferencia del rating expresado de día y el rating expresado de noche para el ítem  $i$ .
5. Finalmente la distancia al cuadrado es equivalente a la suma de los cuadrados de las diferencias.

En un caso práctico se tiene lo siguiente: Considérese el subconjunto de datos extraídos del dataset completo de la tabla 3.5.

Day			Night		
item_id	count	avg	item_id	count	avg
3	2	1	3	2	3
6	2	5	11	1	1
11	2	3,5	13	1	5
13	1	5	19	1	3
15	1	4	23	1	5
16	2	2,5	24	1	5
18	4	3,75	26	1	5
19	2	3	27	1	3
24	1	1	28	1	5
27	2	2	34	1	3
33	1	4	37	1	1

Tabla 3.5: Dataset extraído como ejemplo para el cálculo de distancia de la variable hora Fuente: creación propia

En este caso ya se han separado entre las calificaciones de día y las calificaciones de noche. También se puede observar en la columna “count” el número de calificaciones encontradas para cada ítem y en

### 3.5. Construcción experimental de un Dataset para publicidad ubicua usando información contextual

la columna “avg” el promedio de los valores de los ratings. En el siguiente paso se extraen sólo los ítems que se encuentran en ambos grupos. Ver tabla 3.6.

Day		Night	
item_id	avg	item_id	avg
3	1	3	3
11	3,5	11	1
13	5	13	5
19	3	19	3
24	1	24	5
27	2	27	3

Tabla 3.6: ítems comunes para el cálculo de distancia Fuente: creación propia

Ahora que se han extraído los ítems que se encuentran en ambos grupos, es posible encontrar la diferencia de las calificaciones y la distancia entre cada uno de los valores. Los resultados se encuentran consignados en la tabla 3.7.

item_id	day	night	d	d <sup>2</sup>
<b>3</b>	1	3	-2	4
<b>11</b>	3,5	1	2,5	6,25
<b>13</b>	5	5	0	0
<b>19</b>	3	3	0	0
<b>24</b>	1	5	-4	16
<b>27</b>	2	3	-1	1
			$\Sigma d^2$	<b>27,25</b>

Tabla 3.7: Cálculo final de la distancia para los posibles valores de la variable hora Fuente: creación propia.

En este caso se ha encontrado que la distancia entre día y noche elevada al cuadrado es 27,25. Cabe resaltar que el subdataset seleccionado sólo es una pequeña porción del dataset completo. Pues en éste se encontraron 262 ratings cuya razón fue la hora y en ellos, 48 ítems diferentes que se encontraban en ambos grupos (día y noche). Finalmente el resultado de la distancia elevada al cuadrado fue 156,988. La tabla 3.8 muestra los resultados para todas las variables seleccionadas del modelo.

### Capítulo 3. Contexto en entornos de publicidad ubicua y su implantación en Sistemas de Recomendaciones

	Day	Night
Day	0	156,988
Night		0

(a) Hora

	Week	Weekend
Week	0	33,438
Weekend		0

(b) Día

	Warm	Cold
Warm	0	227,733
Cold		0

(c) Temperatura

	Male	Female
Male	0	374,372
Female		0

(d) Género

	Overcast	Cloudless	Rain
Overcast	0	138,833	4
Cloudless		0	23,25
Rain			0

(e) Clima

	Normal	Boring	Happy	Sad
Normal	0	92,826	178,424	13,000
Boring		0	49,147	---
Happy			0	17,444
Sad				0

(f) Estado de ánimo

	clothes	sports	restaurants	home	entertainment	tech	vehicles
clothes	0	144,711	141,231	234,763	208,630	221,613	131,071
sports		0	111,951	142,945	96,054	163,099	98,732
restaurants			0	182,285	131,833	181,932	101,695
home				0	228,555	266,725	84,461
entertainment					0	182,728	105,998
tech						0	94,278
vehicles							0

(g) Categoría

Tabla 3.8: Distancias entre cada una de las variables de contexto Fuente: creación propia

## 3.6. Conclusiones del capítulo

- En este capítulo se definió un modelo para la información de contexto que puede ser utilizado para la recomendación de publicidad ubicua. Dicho modelo está basado en el de Bauer y Spiekerman, y adaptado a partir de información obtenida de la literatura. Las principales adaptaciones consistieron en eliminar los niveles macro y micro, pues se enfocó en el caso de publicidad conocido como “Digital Signage”, el cual es principalmente situacional; y en la agregación de la variable “interacción con el sistema”.
- Aunque existen tres posibles opciones para hacer la reducción de dimensionalidad, estudios previos demostraron que una aproximación basada en modelo puede generar mejores resultados respecto a la métrica de precisión, en la mayoría de los casos. Para este trabajo se realizaron algunas pruebas usando prefiltrado y postfiltrado, obteniendo con ellas resultados poco relevantes. Dichas conclusiones combinadas con afirmaciones de otros autores dieron paso a la selección de una aproximación basada en modelo como mecanismo de inclusión de contexto dentro del sistema de recomendaciones.
- Se logró generar una adaptación de las ecuaciones de filtrado colaborativo clásico user-user e item-item para agregar en ellas la información contextual. Al hacer estas adaptaciones se encontró que para un filtrado user-user no es posible incluir información del ambiente del proveedor, pues en este caso los ítems se dejan constantes mientras se evalúan las similitudes entre los usuarios. En el caso del filtrado item-item ocurre lo contrario y no es posible incluir información del ambiente del consumidor. El ambiente físico por su parte, puede ser tenido en cuenta en ambos filtrados.
- Las aproximación basada en modelo para la inclusión de información contextual sugiere la necesidad de determinar un mecanismo para medir la similitud entre dos contextos. Para ello se planteó una ecuación de similitud de contextos partiendo de la distancia euclidiana entre dos objetos. Cabe resaltar que esta ecuación de similitud requiere además de las variables contextuales, pesos de importancia para cada una de ellas y, por tanto, se requiere que el dataset permita tener indicios de estos pesos o que se haga un estudio de marketing que permita definirlos con más rigor.
- El dataset es una de las partes más importantes en el momento de hacer pruebas a un sistema de recomendaciones. Sin embargo, para el caso específico de publicidad no son muy comunes los datasets y aún menos que también involucren información contextual. Por este motivo fue necesario diseñar un aplicativo que permitiera recolectar calificaciones sobre algunos anuncios obtenidos del dataset de SMARTA e incluir en ellos información del contexto de la calificación. El aplicativo en cuestión fue desarrollado usando tecnologías Web, contó con la participación de cerca de 100 usuarios y logró conseguir más de 1500 ratings. Dicho dataset se encuentra disponible para ser utilizado en trabajos relacionados.



## Capítulo 4

# Construcción de un sistema de recomendaciones soportado en información contextual

Los sistemas de recomendaciones por su naturaleza requieren de ciertas pruebas que antes de su construcción, den un indicio de los componentes que mejor se adapten a los datos utilizados para hacer predicciones y por supuesto para recomendar. Para ello es necesario conocer cada uno de los pormenores del dominio del mismo, es decir, las variables que se tendrán en cuenta, como la composición del dataset y su forma de almacenamiento, la cantidad de usuarios e ítems, entre otras. Con esta información se pueden determinar datos relevantes, como el tipo de filtrado a utilizar, las técnicas de medida de similitud, los controladores necesarios para acceder a la información, etc. Por este motivo, en el capítulo 3 se definió un modelo contextual adaptado a SRs para publicidad ubicua y un mecanismo que permite incluir las variables definidas en dicho sistema. Este mecanismo parte de una aproximación basada en modelo y por lo tanto, en él se realizan algunas adaptaciones a las ecuaciones de filtrado colaborativo clásico con el objetivo de incluir contexto. Particularmente, el filtrado colaborativo user-user fue adaptado desembocando en la siguiente ecuación:

$$p_{u,i} = \frac{\sum_{u' \in N} s(u', u) \eta(c, c_{r_{u',i}}) r_{u',i}}{\sum_{u' \in N} |s(u', u)| \left| \eta(c, c_{r_{u',i}}) \right|} \quad (3.2)$$

donde el factor  $\eta(c, c_{r_{u',i}})$  representa la similitud de contextos que puede ser calculada a partir de la ecuación 3.9. Dadas las características de este tipo de filtrado, sólo el ambiente del consumidor y el físico pueden ser tenidos en cuenta como variables de contexto. Por otra parte, el filtrado colaborativo item-item se adaptó dando como resultado la ecuación:

$$p_{u,i} = \frac{\sum_{i' \in I} s(i', i) \eta(c, c_{r_{u,i'}}) r_{u,i'}}{\sum_{i' \in I} |s(i', i)| \left| \eta(c, c_{r_{u,i'}}) \right|} \quad (3.4)$$

donde nuevamente  $\eta(c, c_{r_{u,i}})$  se calcula a partir de la ecuación 3.9, pero en este caso, sólo puede considerarse el ambiente del proveedor y físico como variables de contexto. Adicionalmente y considerando que no fue posible obtener un dataset que se ajuste a las necesidades propias del sistema, se recolectó de forma experimental un conjunto de datos a partir del modelo contextual planteado, facilitando su experimentación y evaluación.

Aunque estos elementos son insumos importantes para la eventual implementación de un SR que involucre información contextual, aún hace falta determinar una arquitectura que se adecúe a las necesidades del sistema. Es importante destacar que dicha arquitectura puede depender en gran medida del framework utilizado para soportar la construcción, pues éste puede estar ya basado en una arquitectura que es necesario recuperar, revisar y determinar si puede ser adaptada y/o reutilizada.

En ese orden de ideas, este capítulo se enfoca en el diseño del SR en sí, del cual se espera construir una implementación que pueda servir como referencia para trabajos relacionados y también como piloto que permita realizar la posterior evaluación. El sistema tendrá como base las ecuaciones definidas en el capítulo 3 y para su construcción se definirá inicialmente el framework para SR utilizado. A partir de él se recuperará la arquitectura que será tomada como referencia y luego adaptada y modelada.

#### 4.1. Mahout: framework seleccionado para soportar el sistema

Aunque Mahout fue la selección final, es importante mencionar los otros frameworks que se tuvieron en cuenta y el proceso de selección antes de presentar las principales características del mismo. El listado completo de frameworks consultados incluye quince opcionados, de los cuales, a partir de un filtrado inicial donde se consultó su sitio web y se determinó que no era posible acceder a información del proyecto, se eliminaron seis. Los nueve restantes se filtraron a partir de las siguientes características:

- **Método de recomendaciones soportados:**

El modelo de inclusión de contexto planteado en el capítulo 3 está basado en filtrado colaborativo user-user e item-item. Por este motivo, se requiere que el framework soporte estos dos métodos de recomendaciones.

- **Licencia:**

Es importante que el framework pueda ser estudiado sin restricciones, no se pretende distribuir de forma comercial el resultado de la investigación pero es importante que legalmente sea posible estudiar y utilizar el framework con fines académicos.

- **Nivel de documentación:**

Se definieron tres niveles de documentación, los cuales son: *alto*: existe mucha documentación y una comunidad de personas soportando el framework; *medio*: existen al menos bosquejos del funcionamiento del framework; y *bajo*: no se encuentra documentación suficiente para ponerlo en funcionamiento o hacer adaptaciones.

## 4.2. Recuperación de la Arquitectura

### ■ Disposición de código fuente:

El desarrollo del sistema planteado puede implicar la modificación de algunos de los componentes del framework. Por tanto, se espera contar con el código fuente que facilite dicha adaptación o modificación.

A partir de esto se generó la tabla 4.1 que lista los marcos de trabajo consultados y sus características. Mahout por su parte, fue seleccionado para ser utilizado durante el diseño y construcción del sistema principalmente por su buena documentación. Adicionalmente es un framework que no sólo cuenta con las herramientas necesarias para construir SRs, sino que también implementa varias métricas de cálculo de similitud, controladores para acceso a datasets desde bases de datos, archivos planos, entre otros; también cuenta con herramientas de evaluación de sistemas y finalmente, porque es utilizado para ser puesto en producción. No en vano grandes compañías como Microsoft Azure y Amazon AWS cuentan con implementaciones del mismo. Otros frameworks como Cofi no contaban con el código fuente disponible y Lenskit que también tiene muy buenas características no presentó una documentación clara en el momento de la consulta.

Framework	Métodos soportados	Lenguaje de programación	Licencia	Nivel de documentación	Disposición de código fuente
EasyRec	c. filtering, hybrid	Java	GNU - Opensource	Medio	Disponible
Mahout	c. filtering, content based, hybrid	Java	GNU - Opensource	Alto	Disponible
Crab	C. filtering	Python	Opensource	Bajo	Disponible
Cofi	C. Filtering	Java	Desconocido	Bajo	No disponible
SVDFeature	C. Filtering	C++	GNU - Opensource	Medio	Disponible
Vogoo	C. Filtering	PHP	GNU - Opensource	Bajo	Disponible
Duine	Hybrid	Java	Opensource	Alto	Disponible
Lenskit	c. Filtering	Java	Opensource	Bajo	Disponible

Tabla 4.1: Frameworks para la construcción de SR explorados Fuente: creación propia a partir de datos obtenidos de: EasyRec[96], Mahout[43], Crab[97], Cofi[98], SVDFeature[99], Vogoo[100], Duine[101], Lenskit[102]

## 4.2. Recuperación de la Arquitectura

Mahout cuenta con una arquitectura basada en cuatro módulos sobre la cual puede soportarse un sistema de recomendaciones simple. La figura 4.1 presenta dicha arquitectura y los módulos que la componen se describen a continuación:

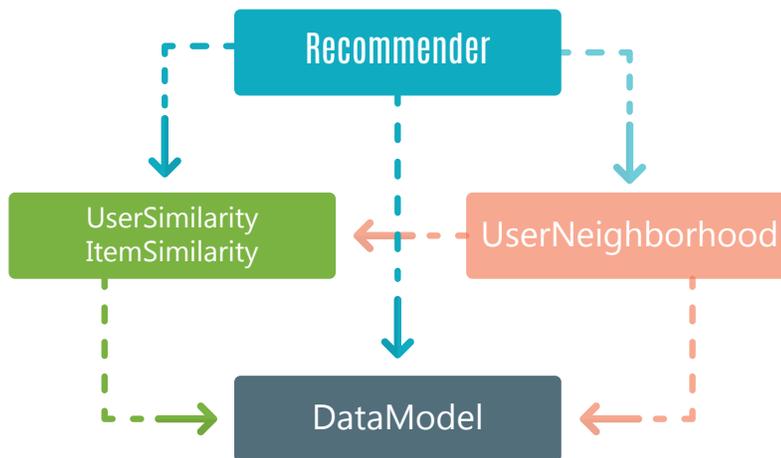


Figura 4.1: Módulos de un SR simple Fuente: Adaptación de [89]

#### 4.2.1. Modelo de datos (DataModel)

El modelo de datos es la representación software de la información contenida dentro del dataset utilizado. Mahout abstrae cada pieza en el modelo de datos con un objeto básico denominado “Preference” o preferencia, la cual contiene tres atributos: el identificador del ítem para el cual se ha expresado la preferencia, el identificador del usuario que ha expresado la calificación y la preferencia como tal, que es un número en una escala que depende del sistema e indica el gusto del usuario por el ítem dado.

#### 4.2.2. Mecanismo de similitud (UserSimilarity o ItemSimilarity)

Como ya se ha mencionado, los sistemas de recomendaciones y particularmente los filtrados colaborativos requieren de un mecanismo que permita medir la similitud entre los ítems o entre los usuarios y a partir de ello, seleccionar un conjunto de elementos -muy parecidos al actual- que permitan predecir los gustos del usuario por ítems que no ha calificado. Este módulo cuenta con implementaciones de distintas métricas de similitud entre las que se encuentran Pearson, Spearman, y Distancia Euclidiana.

#### 4.2.3. Vecindad (UserNeighborhood)

Este módulo es utilizado únicamente en el filtrado user-user. A partir de él y del mecanismo de similitud se determina un conjunto de usuarios que son muy parecidos al usuario actual del sistema. De esta manera, se evita introducir ruido a las predicciones a partir de usuarios que no tienen mucho que aportar. Entre sus implementaciones se encuentran las vecindades de tamaño fijo y las vecindades de tamaño dinámico generadas a partir de umbrales.

### 4.3. Arquitectura de referencia incluyendo información de contexto

#### 4.2.4. Recomendador (Recommender)

Finalmente, este módulo es el encargado de hacer las recomendaciones utilizando para ello todos los anteriores. Mahout cuenta con diferentes implementaciones para el recomendador entre las que se incluyen, los filtrados user-user e item-item que son genéricos y se configuran a partir de la selección de los módulos anteriores; y también algoritmos de recomendación específicos, como es el caso de SlopeOne o SVD.

### 4.3. Arquitectura de referencia incluyendo información de contexto

Como ya puede inferirse la arquitectura de referencia está basada en la arquitectura de un recomendador simple de Mahout pero esta vez se ha agregado un módulo esencial que habilita al sistema con la capacidad de considerar el contexto dentro de las recomendaciones. La figura 4.2 presenta un bosquejo de la arquitectura con el módulo adicional que se ha planteado.

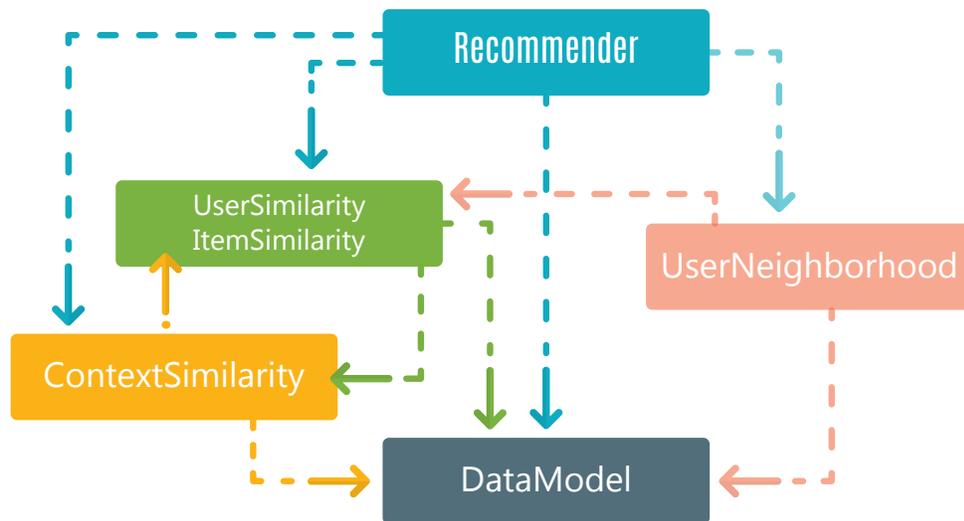


Figura 4.2: Módulos de un SR incluyendo contexto Fuente: creación propia

En ella cabe destacar que aunque los elementos iniciales tomados de Mahout no se ven modificados en su nombre, si sufren algunas variaciones para permitir el ingreso del nuevo módulo. Es importante rescatar también que Mahout utiliza una arquitectura basada en interfaces de JAVA y de esta manera, la modificaciones de las que se habla no se hacen directamente al código de Mahout sino que es posible implementar o extender dichas interfaces para habilitar el contexto. Como puede observarse la similitud del contexto genera nuevas conexiones en la arquitectura, y entre ellas tal vez una de las más importantes viene del recomendador, pues ahora todas las predicciones y recomendaciones se realizan considerando los valores arrojados por el módulo de similitud de contextos. La similitud de usuarios e ítems también se ven conectados con el nuevo módulo pues como se presentó en el capítulo 3, es posible que estos valores de similitud se vean afectados cuando el contexto es considerado. La

siguiente sección se enfoca en la descripción al detalle de la arquitectura de referencia utilizada, para ello se ha partido del modelo de 4+1 vistas del Proceso Unificado de Desarrollo (RUP) [103].

## 4.4. Descripción de la arquitectura de referencia mediante el modelo de 4+1 vistas de RUP

El modelo de 4+1 vistas arquitectónicas presentado por Kruchten en 1995 [104] describe la arquitectura de un sistema utilizando vistas concurrentes, de forma que cada una de ellas responden a requerimientos de diferentes stakeholders. A continuación se describe cada una de las vistas para la arquitectura del sistema.

### 4.4.1. Vista lógica

Se enfoca principalmente en dos elementos: el primero son los requisitos funcionales del sistema, es decir, aquellas funcionalidades que el sistema debe brindar a un usuario final y para ello se puede utilizar un diagrama de casos de uso; el segundo elemento es la abstracción del sistema, donde se representan en forma de objetos cada una de las partes importantes y para ello suele utilizarse un diagrama de clases o un diagrama entidad relación cuando el sistema cuenta con alguna base de datos. A continuación se presentan cada uno de los diagramas planteados.

#### 4.4.1.1. Diagrama de casos de uso del sistema

La figura 4.3 presenta las funcionalidades básicas que un usuario del sistema podría realizar con él. Por su parte, cabe destacar que aunque el caso de uso “Obtener recomendaciones usando contexto” se puede considerar como la funcionalidad principal del sistema, pues es la que realiza el proceso básico de recomendación, existen también otros casos que son importantes. Por ejemplo, agregar la información contextual, el cuál puede ejecutarse internamente en el sistema o hacerlo el usuario de forma explícita; o calificar un ítem, son funcionalidades que apoyan a la generación de recomendaciones ya que permiten mantener actualizada la información del dataset.

Una vez descritos cada uno de los casos de uso, el siguiente paso es presentar el diagrama de clases que soportarán los casos de uso planteados.

#### 4.4.1.2. Diagrama de clases del sistema

La figura 4.4 presenta el diagrama de clases, el cual ayuda a entender como se distribuyen y relacionan cada uno de los elementos desde un punto de vista de abstracción en objetos.

El diagrama de clases ha sido distribuido en cuatro paquetes importantes:

1. **Mahout:** Este paquete cuenta con todas las clases que se tuvieron en cuenta para realizar la extensión del framework y habilitar la inclusión de información de contexto. Aunque este paquete no cuenta con clases nuevas que hayan sido construidas es importante incluirlo dentro del diagrama para entender de una mejor manera como se organizan los otros elementos del sistema.

#### 4.4. Descripción de la arquitectura de referencia mediante el modelo de 4+1 vistas de RUP

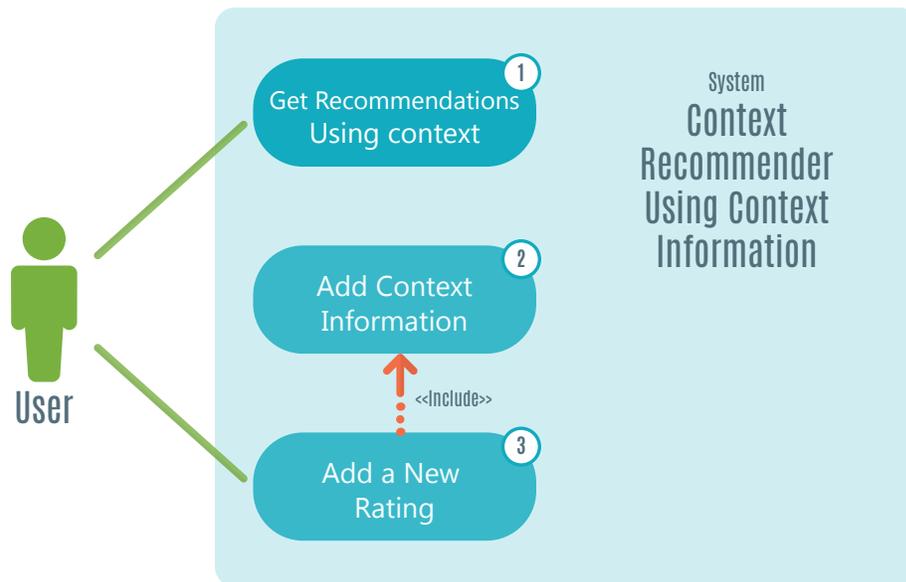


Figura 4.3: Diagrama de casos de uso del sistema Fuente: creación propia

2. **Datamodel:** Dentro de este paquete se encuentran todos los objetos relacionados con la construcción del modelo de datos del sistema. Las clases que aquí se incluyen son:

- **ContextPreference:** Como ya se ha mencionado, la unidad de información más simple utilizada por Mahout es conocida como *Preference* o preferencia, y básicamente es un elemento que combina un identificador de un usuario con un ítem a partir de un valor de preferencia. La clase *ContextPreference* es una hija de la clase *Preference* que además incluye un identificador del contexto en el momento de expresar la calificación.
- **UserInfo, ItemInfo y Context:** Estas tres clases contienen la información de usuarios, ítems y contextos respectivamente. Dicha información puede ser utilizada para realizar el proceso de cálculo de similitud entre contextos, pues los ítems contienen información del ambiente del proveedor, los usuarios contienen datos del ambiente del consumidor y los objetos de contextos contienen la información del ambiente físico y el estado de ánimo del usuario. Esta última es una característica que varía para cada momento en que el usuario haya calificado y por este motivo no se encuentra incluida en la información del usuario.
- **ContextDatamodel:** Como su nombre permite deducir, se trata de la clase encargada de generar el modelo de datos con información contextual. Para no salirse de los lineamientos de Mahout, esta clase es un interfaz que hereda de la clase *DataModel*. Puesto que es una interfaz, simplemente contiene un bosquejo de los métodos que deben implementarse para poner en marcha el modelo de datos. Dado a que la información recolectada como dataset del sistema se encuentra almacenada en una base de datos MySQL, para este proyecto se

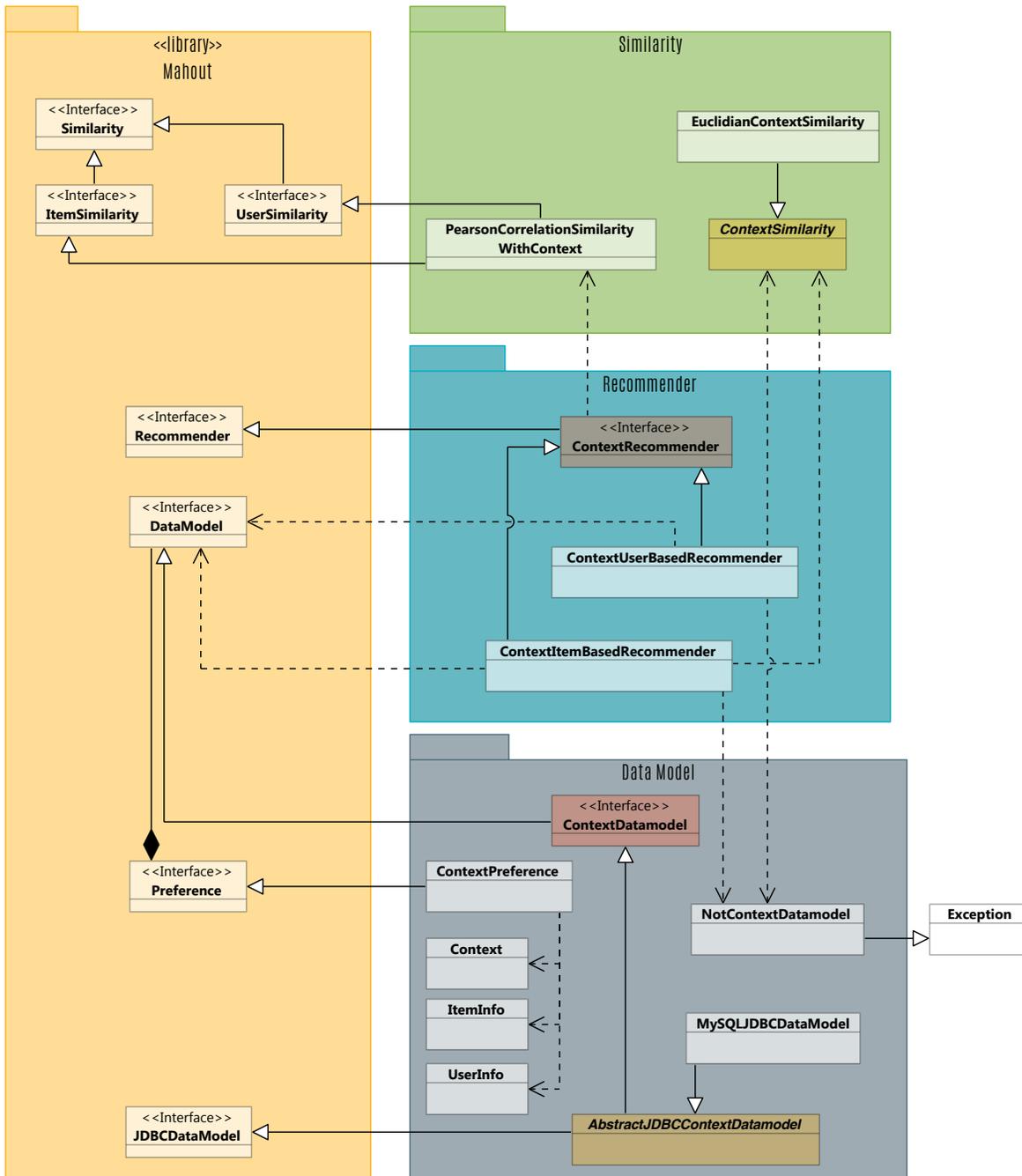


Figura 4.4: Diagrama de clases del sistema Fuente: creación propia

#### 4.4. Descripción de la arquitectura de referencia mediante el modelo de 4+1 vistas de RUP

---

desarrolló una implementación de esta interfaz que permite generar el modelo de datos a partir de una base de datos utilizando un conector JDBC. Dicha clase se construyó de forma abstracta y se implementó un clase hija específica para la conexión con una base de datos MySQL. De esta manera, si el dataset se encuentra almacenado en un motor de bases de datos distinto, solo bastará con desarrollar una clase hija que soporte dicho motor.

- **NoContextDatamodel:** Esta excepción es lanzada cada vez que se construya un recomendador con información contextual pero no utilice un objeto de tipo *ContextDatamodel* como fuente de entrada para los datos.
3. **Similarity:** Este paquete incluye todas las clases necesarias para realizar los cálculos de similitud, tanto la similitud entre contextos, como la similitud entre usuarios e ítems utilizando contextos. Para ello se construyeron dos clases importantes:
- **ContextSimilarity:** Es una clase abstracta con todos los métodos necesarios para calcular la similitud entre un par de contextos. Es implementada a partir de una clase hija llamada *EuclidianContextSimilarity*, la cual, realiza el proceso de comparación de contextos utilizando la distancia Euclidiana planteada en el capítulo 3.
  - **PearsonCorrelationSimilarityWithContext:** Esta clase implementa un cálculo de similitud entre usuarios e ítems teniendo en cuenta la información de contextos. En otras palabras es una implementación de lo planteado en la ecuación 3.6.
4. **Recommender:** Finalmente, en este paquete se encuentran contenidas las clases encargadas de hacer las recomendaciones. Las cuales son:
- **ContextRecommender:** Es una interfaz que hereda de la interfaz de Mahout *Recommender*, ella se encarga de presentar todos los métodos necesarios para realizar las recomendaciones. Es implementada a partir de las siguientes clases dentro del mismo paquete.
    - **ContextItemBasedRecommender:** Implementa un recomendador basado en ítems con información contextual. Para ello utiliza las predicciones planteadas en la ecuación 3.4.
    - **ContextUserBasedRecommender:** Al igual que la anterior, implementa un recomendador pero esta vez basado en usuarios con información contextual. Para ello utiliza las predicciones planteadas en la ecuación 3.2.

### 4.4.1.3. Diagrama entidad relación

La Figura 4.5 presenta el diagrama entidad relación de la base de datos. En él se han especificado la siguientes cuatro tablas:

- **Ads - Anuncios:** Es la tabla que contiene la información de los ítems, ó en este caso anuncios publicitarios. Sus atributos son: id o identificador único del anuncio, url donde se almacena una imagen alusiva, título, descripción y habilitado que indica si el anuncio se encuentra disponible para recibir calificaciones.
- **Users - Usuarios:** Es la tabla que almacena la información de los usuarios.
- **Contexts - Contextos:** Almacena la información de los contextos.
- **Taste preferences - Preferencias:** Esta tabla almacena las calificaciones que los usuarios dan a los ítems. En ella también se establece el contexto en el momento de la calificación y se incluyen campos para almacenar la razón por la cual se dio la calificación.

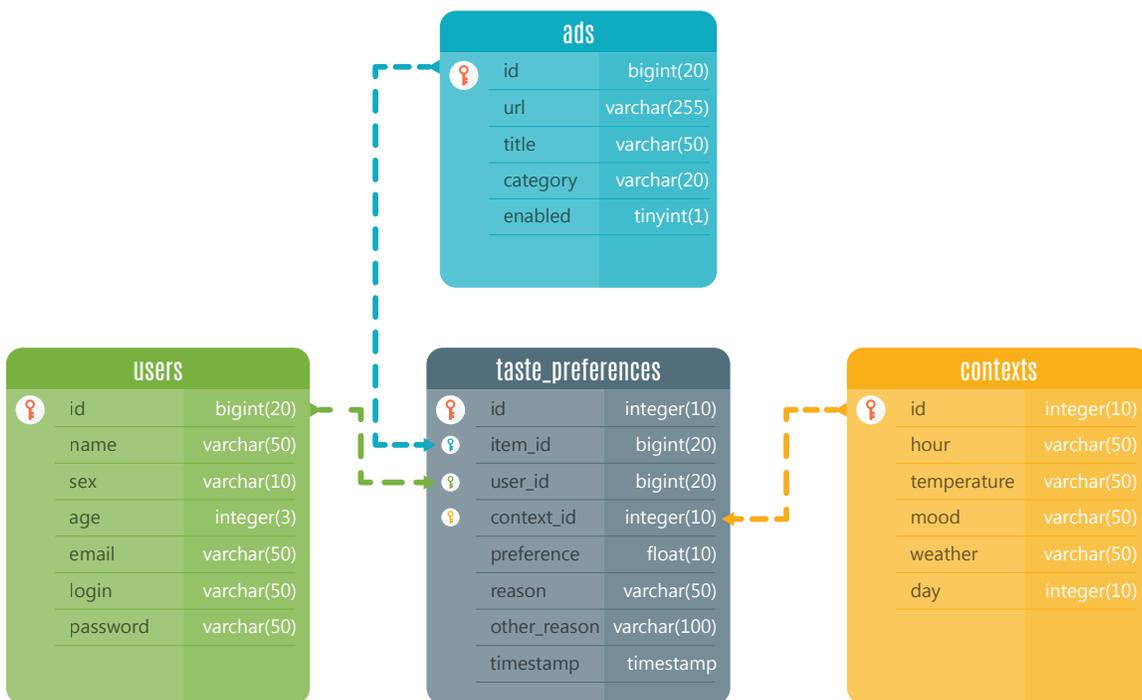


Figura 4.5: Diagrama entidad relación del dataset Fuente: creación propia

Con esto se da por completada la vista lógica. La siguiente vista a detallar es la vista de procesos.

## 4.4. Descripción de la arquitectura de referencia mediante el modelo de 4+1 vistas de RUP

---

### 4.4.2. Vista de procesos

Esta vista se enfoca en asuntos de concurrencia y distribución del sistema y especifica el hilo conductor o el flujo para los procesos más importantes. Para la presentación de esta vista se han considerado diagramas de secuencia para la solicitud de recomendaciones por parte del usuario al Recommender, pues permiten comprender las interacciones entre cada uno de los módulos presentados en el diagrama 4.2; También se han utilizado diagramas de actividad para los procesos de estimación de preferencias ejecutados dentro del recomendador y que representan la implementación de las ecuaciones presentadas en el capítulo 3.

#### 4.4.2.1. Diagramas de secuencia: Solicitud de recomendaciones para un recomendador user-user e item-item incluyendo contexto

La figura 4.6 presenta el diagrama de secuencia para el proceso ejecutado cuando un usuario solicita  $N$  recomendaciones a un recomendador que se basa en un filtrado user-user y además incluye información contextual. El proceso inicia cuando un usuario solicita la recomendación de algunos ítems. Para ello entrega el id del usuario para el cual requiere las recomendaciones, el número de ítems que desea y el id del contexto. El recomendador inicialmente realiza una validación de la configuración donde determina si todos los módulos se encuentren correctamente ajustados, verifica la existencia del usuario, el contexto en el modelo de datos y comprueba que se haya solicitado al menos un ítem. Cabe resaltar que para el caso de los filtrados user-user entra en juego el módulo de vecindad, el cual haciendo uso del módulo de similitud de usuarios calcula un conjunto de usuarios parecidos a aquel para el cual se desea recomendar. Posteriormente es éste conjunto de usuarios el que se utiliza para determinar el listado de ítems que son potencialmente recomendables. Acto seguido, para cada uno de estos ítems se realiza un proceso de estimación de preferencias haciendo uso también del módulo de similitud de contextos. Finalmente los ítems que cuentan con mayor estimación en la preferencia son seleccionados para ser recomendados al usuario.

Al igual que en la figura 4.6, la figura 4.7 presenta el diagrama de secuencia para el proceso de solicitud de recomendaciones incluyendo información contextual, pero esta vez el recomendador está basado en un filtrado item-item. Es importante rescatar que en este caso no existe el módulo de vecindad, pues dado a que la recomendación se enfoca principalmente en los ítems, no es necesario crear un conjunto de usuarios parecidos al actual. En este caso, el recomendador solicita directamente al modelo de datos todos los elementos que son potencialmente recomendables. Es decir, aquellos para los cuales el usuario no ha expresado una calificación y otros usuarios sí han calificado. A partir de ellos pasa a la fase de estimación de preferencias y nuevamente se escogen aquellos ítems con mayor preferencia estimada para ser recomendados.

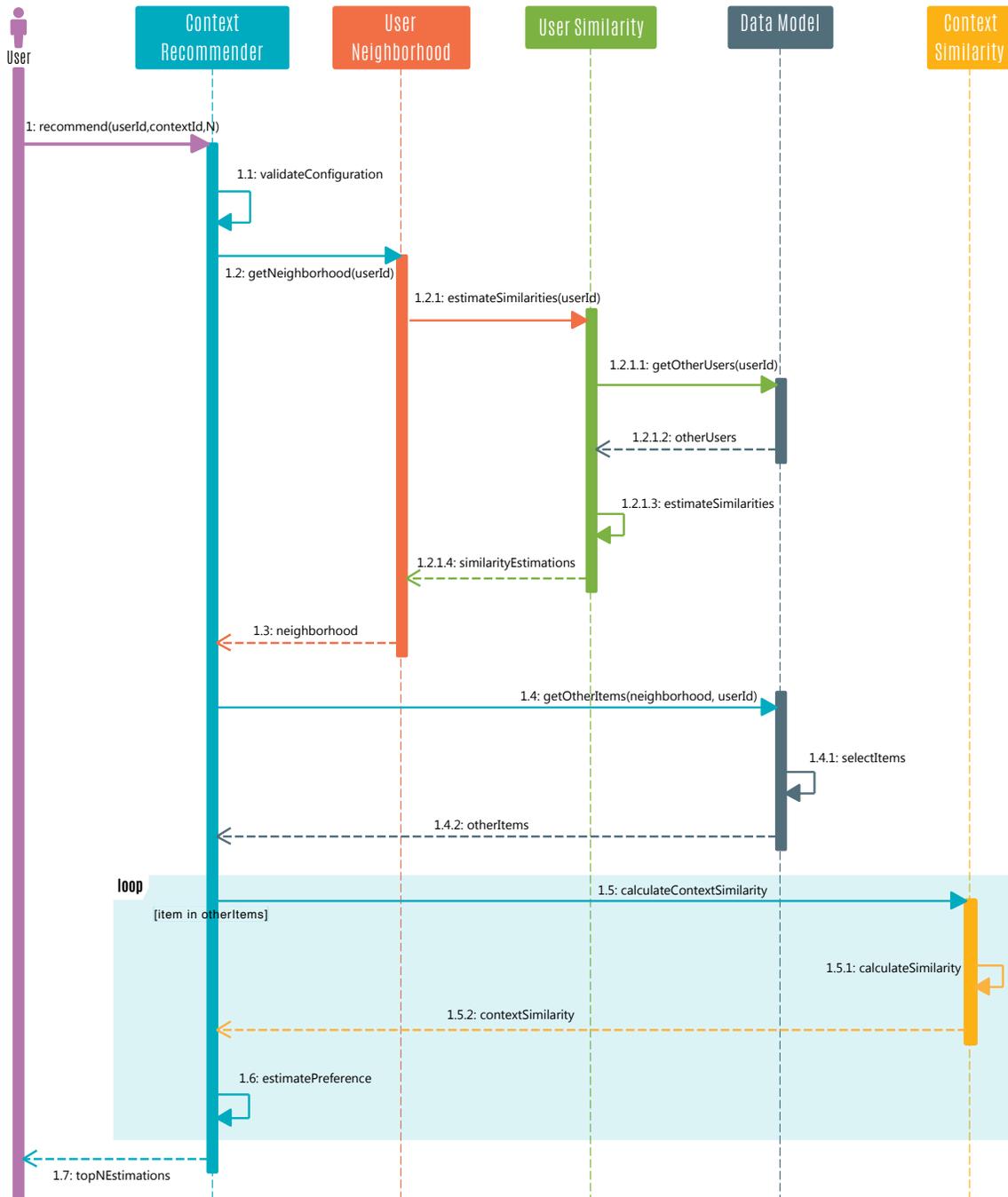


Figura 4.6: Diagrama de secuencia para un filtrado user-user incluyendo contexto Fuente: creación propia

#### 4.4. Descripción de la arquitectura de referencia mediante el modelo de 4+1 vistas de RUP

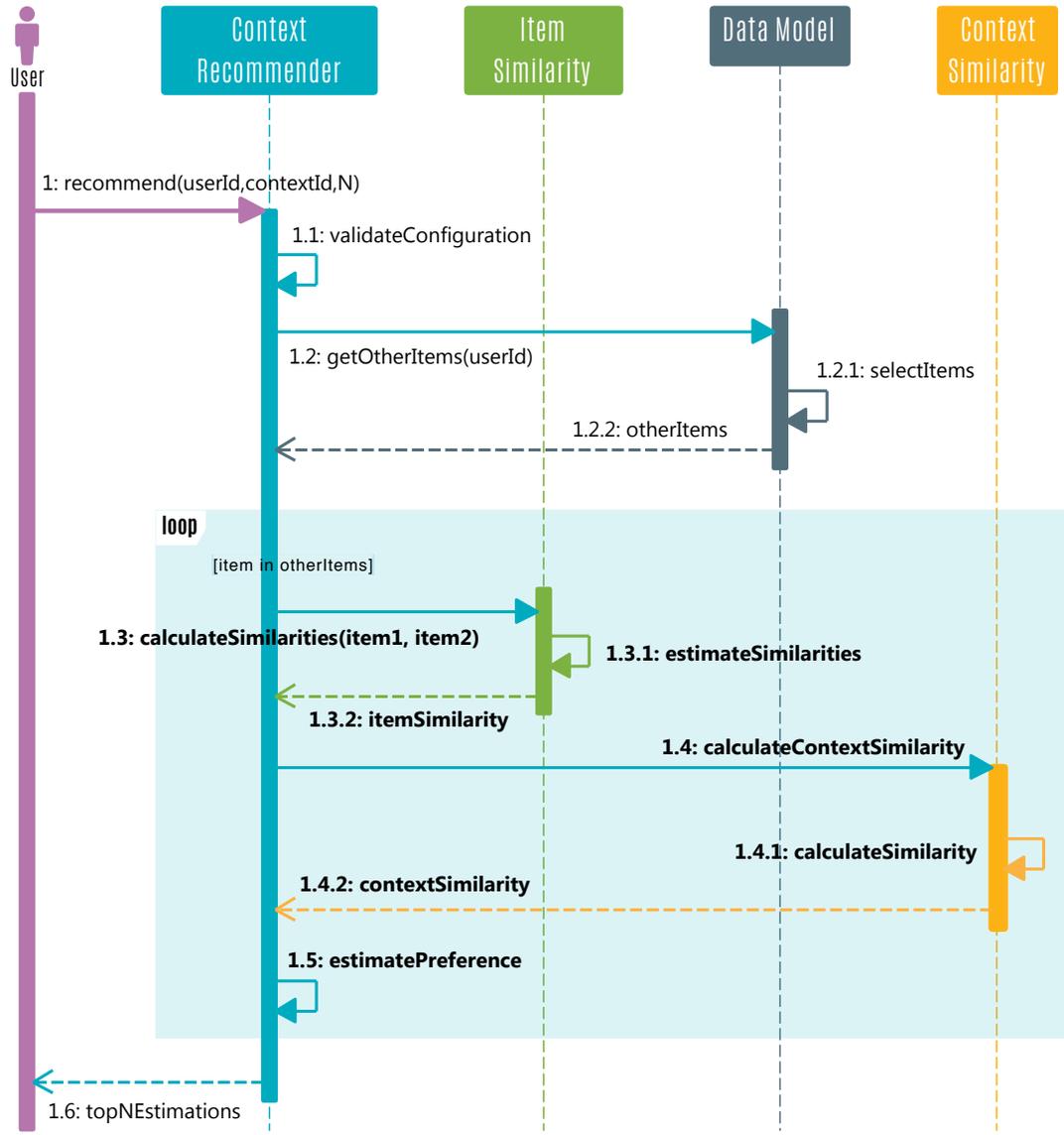


Figura 4.7: Diagrama de secuencia para un filtrado item-item incluyendo contexto Fuente: creación propia

4.4.2.2. Diagramas de actividad: Estimación de preferencias en un recomendador user-user e item-item incluyendo contexto

Los diagramas de actividad que se presentan a continuación dejan ver con mayor detalle el que es tal vez el paso más importante del proceso de recomendaciones. Es decir, la estimación de preferencias. Al igual que en el apartado anterior se ha definido un diagrama tanto para un recomendador basado en un filtro user-user como en uno item-item. La figura 4.8 permite ver el flujo de actividades realizadas para estimar la preferencia de un usuario por un ítem en un contexto dado utilizando un filtrado user-user.

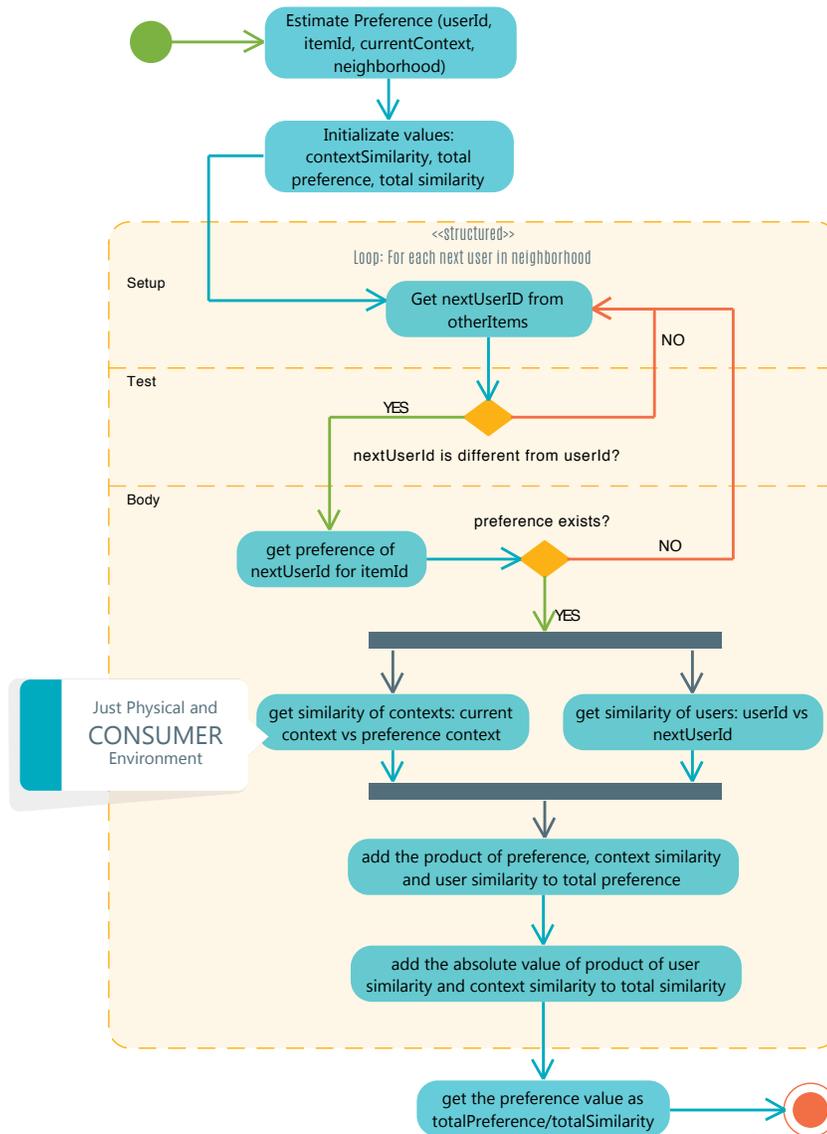


Figura 4.8: Diagrama de actividad para un filtrado user-user incluyendo contexto Fuente: creación propia

#### 4.4. Descripción de la arquitectura de referencia mediante el modelo de 4+1 vistas de RUP

En ella puede observarse que para estimar la preferencia de un usuario por un ítem, se tienen en cuenta sólo los usuarios que hacen parte de la vecindad y se utilizan las preferencias que dichos usuarios han expresado, combinándolas con la similitud de contextos y la similitud entre los usuarios. Es decir, es una implementación de la ecuación 3.2.

La figura 4.9 por su parte, permite ver este mismo paso del proceso de recomendación pero esta vez para un filtrado item-item. Como puede observarse no se tiene en cuenta la vecindad de usuarios, sino un conjunto de otros ítems que han sido calificados por otros usuarios del sistema. La similitud de contexto esta vez se calcula únicamente teniendo en cuenta el ambiente físico y del proveedor. Este diagrama representa la implementación de la ecuación 3.4.

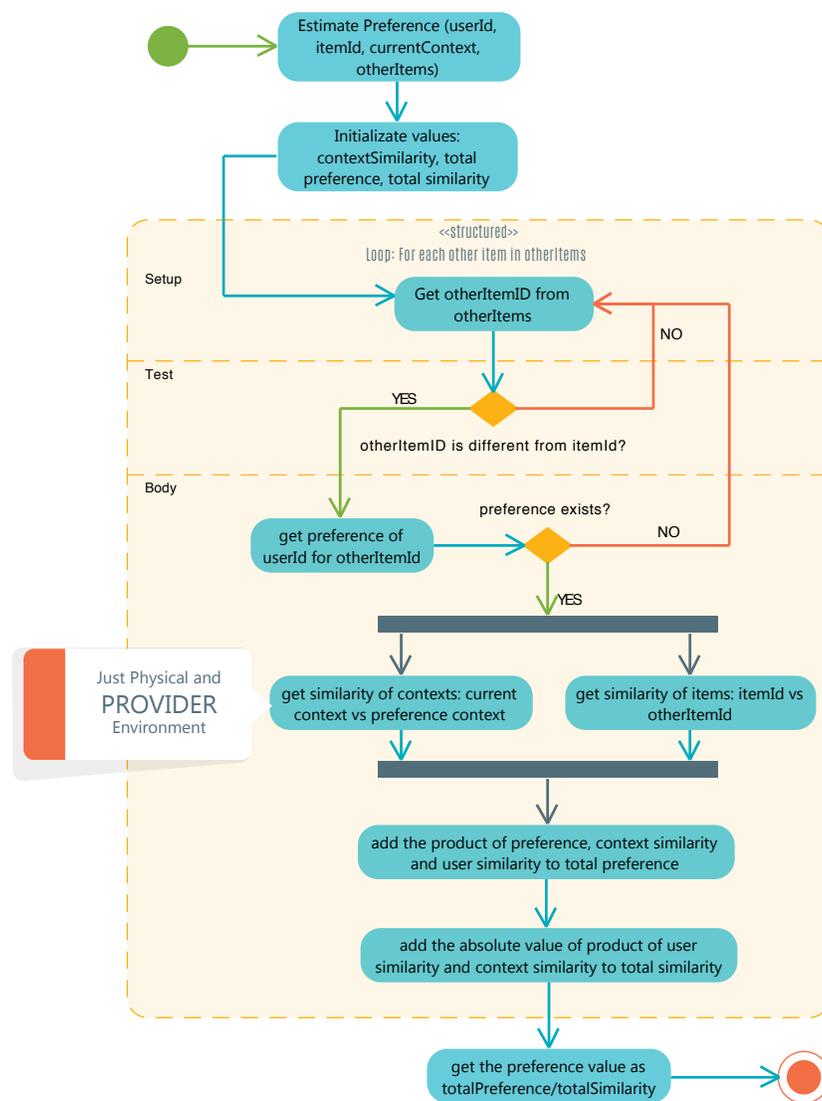


Figura 4.9: Diagrama de actividad para un filtrado item-item incluyendo contexto Fuente: creación propia

### 4.4.3. Vista de desarrollo

Como se menciona en [104], la vista de desarrollo se centra en la organización real de los módulos en el ambiente de desarrollo del software. Este es empaquetado en partes pequeñas –bibliotecas de programas o subsistemas– que pueden ser desarrollados por un programador o un grupo pequeño de desarrolladores. Dichos sub-sistemas se organizan en una jerarquía de capas, de tal forma que una capa solo requiera de las inferiores para subsistir.

#### 4.4.3.1. Diagrama de componentes del sistema

La figura 4.10 presenta un diagrama de componentes de la arquitectura del sistema.

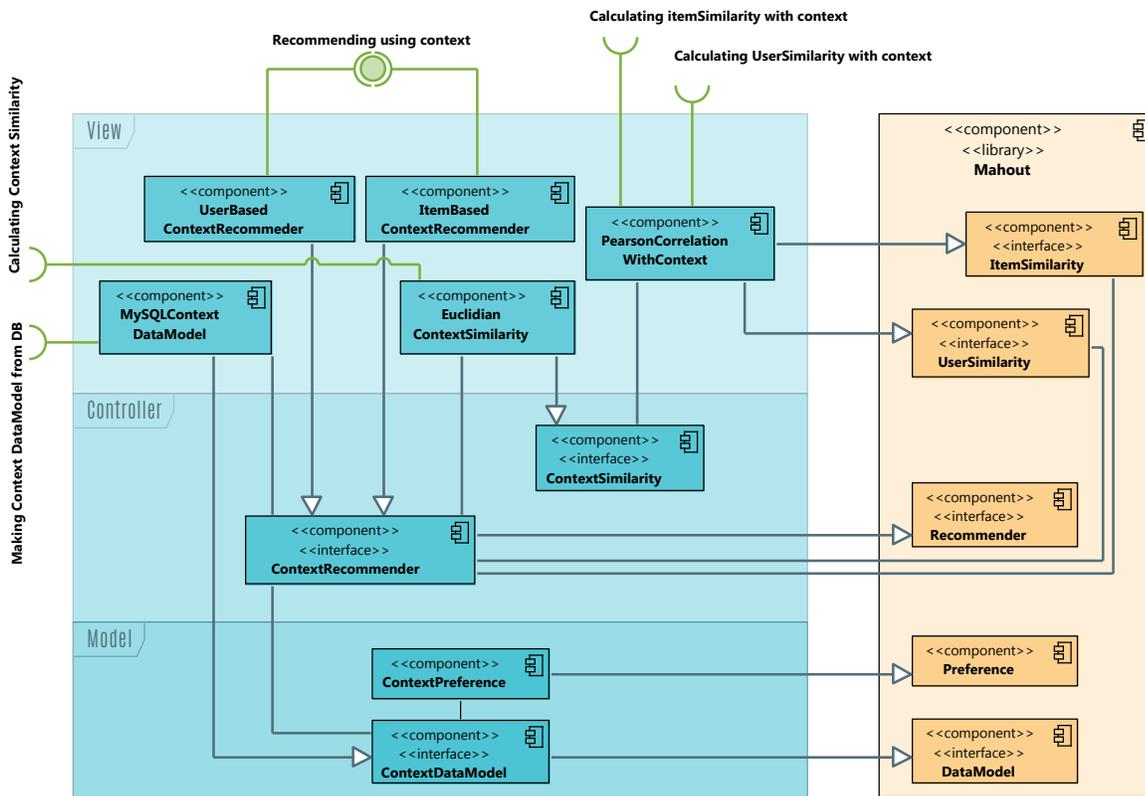


Figura 4.10: Diagrama de componentes Fuente: creación propia

En el diagrama de la figura 4.10 se presenta una distribución en capas del sistema. Para dicha distribución se ha partido de un modelo-vista-control donde el modelo se refiere a la abstracción de preferencias a partir del dataset que puede estar almacenado en una base de datos o en archivos de texto plano; en el control se encuentran los componentes esenciales para la lógica del sistema y en él se han ubicado algunas interfaces para el recomendador y el medidor de similitud de contextos; finalmente, en la vista se ubican algunas implementaciones de las interfaces creadas en el control,

## 4.4. Descripción de la arquitectura de referencia mediante el modelo de 4+1 vistas de RUP

---

de forma que dichas implementaciones pueden ser utilizadas para ejecutar las acciones que fueron planteadas en el diagrama de casos de uso de la figura 4.3. Dado a que el sistema se ha planteado a partir de Mahout, muchos de los componentes dependen o son generalizaciones de elementos propios del framework.

Por otro lado, con la vista de desarrollo se presenta cómo se organizan cada uno de los componentes en una arquitectura en capas. Sin embargo, es necesario determinar cómo debe ser la distribución del software para su eventual puesta en marcha, y es para ello que se presenta la vista de despliegue.

### 4.4.4. Vista de despliegue

La vista de despliegue o arquitectura física se enfoca principalmente en los requisitos no funcionales del software, de forma que permite mapear el software en el hardware y la distribución física del mismo. Para describir la arquitectura física se ha recurrido al diagrama de despliegue presentado en la figura 4.11.

#### 4.4.4.1. Diagrama de despliegue del sistema

La figura 4.11 describe la configuración de equipos utilizada en la implementación del sistema de recomendaciones que incluye información contextual. Cabe destacar que este diagrama muestra la configuración utilizada en el laboratorio para realizar pruebas del desarrollo, pero no significa que no puedan utilizarse equipos hardware de distintos fabricantes. Básicamente se trabajó en un equipo con sistema operativo Mac OS X. Sobre él se instaló la máquina virtual de Java V7 y en ella se utilizó el framework mahout el cual se incluye como librerías al proyecto. En cuanto al dataset, fue almacenado en un motor de bases de datos MySQL 5.1 en un equipo en la nube al que se accede a través de la API JDBC sobre protocolo HTTP.

En este diagrama de despliegue se debe destacar el componente ***Mahout-Context-Recommender*** el cual es una librería para la construcción de sistemas de recomendaciones basados en contexto que fue construida como un aporte adicional de este proyecto y que se presenta como una contribución a la comunidad de desarrolladores, usuarios y otros involucrados en el marco de trabajo Mahout. Dicha librería contiene todas las clases planteadas en el diagrama 4.4 y fue diseñada para utilizarse como una extensión del framework, de forma que es altamente escalable y adaptable a diferentes dominios. En el anexo A se encuentra la descripción de los casos de uso o funcionalidades que un usuario desarrollador podría realizar con la misma, y en el anexo B se presenta un tutorial para la construcción de un SR basado en contexto partiendo de la librería.

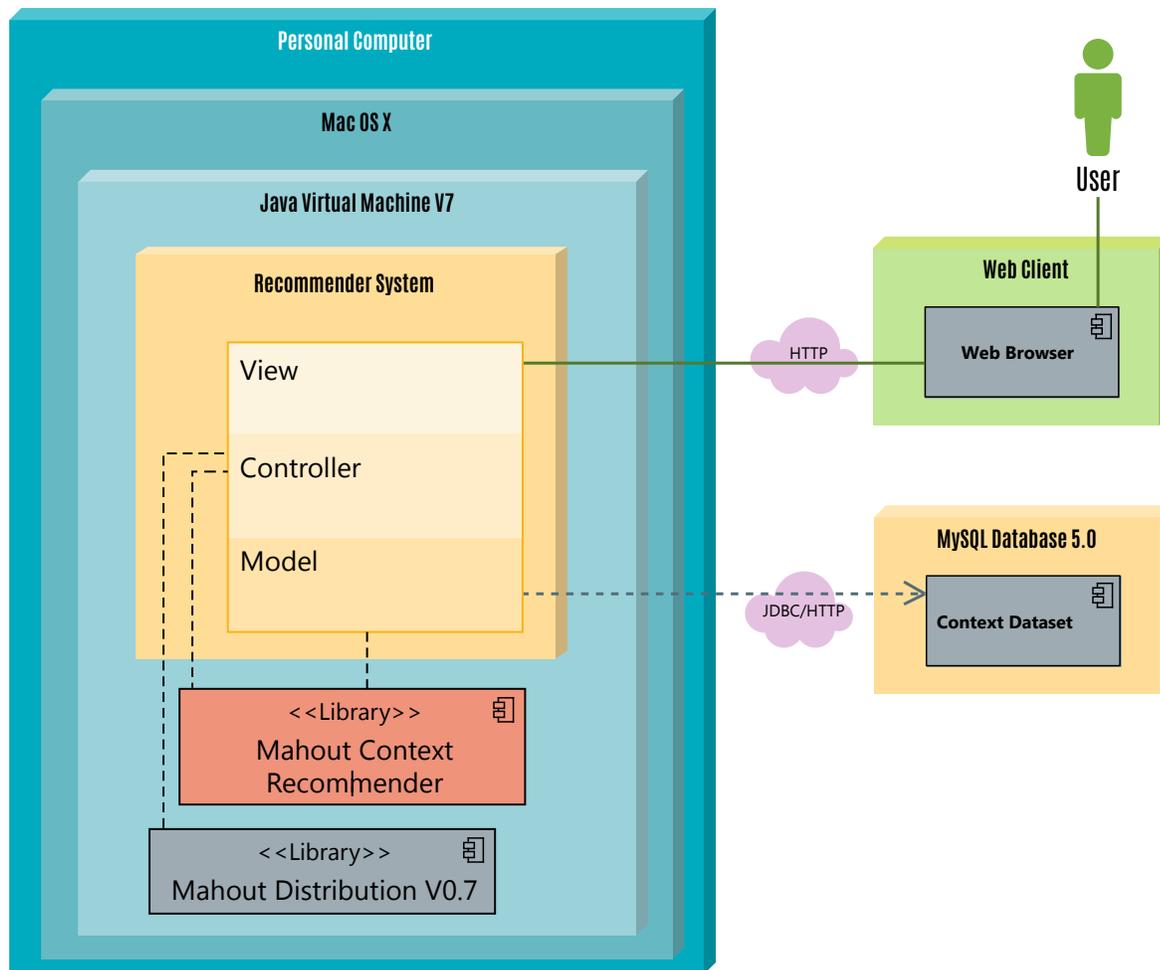


Figura 4.11: Diagrama de despliegue Fuente: creación propia

### 4.5. Conclusiones del capítulo

- Tomando como base la arquitectura de un sistema de recomendaciones simple de Mahout, en este capítulo logró construirse una arquitectura de referencia para sistemas de recomendaciones que involucran información contextual utilizando las ecuaciones planteadas en el capítulo 3. Dicha arquitectura, puede superponerse al marco de trabajo Mahout y facilitar su utilización por parte de terceros. Adicionalmente se construyó un piloto que permite la evaluación del sistema construido, y que será descrito con detalle en el capítulo 5 donde, a partir del plan de pruebas, será más fácil de comprender.
- Dado a que se diseñó una arquitectura basada en *interfaces* de JAVA, se hace fácilmente escalable para sistemas de recomendaciones en otros entornos. De hecho, se facilita la generación algoritmos de recomendaciones utilizando contextos diferentes a los planteados durante el desarrollo de este proyecto, pues para ello sólo bastaría con extender la clase “*ContextRecommender*”, demostrando la facilidad de escalabilidad del sistema.
- Como aporte adicional, se construyó la librería “Mahout Context Recommender”, la cual fue diseñada para que la construcción de los SRs basados en contexto se puedan realizar de una forma simple. Esto puede verse evidenciado en el anexo B donde con cinco sencillos pasos puede construirse un sistema de este tipo completamente habilitado para generar recomendaciones. En cuanto a la facilidad de evaluación de los sistemas desarrollados utilizando dicha librería, vale la pena destacar que guarda una estrecha relación con los módulos básicos de Mahout y por tanto, es posible utilizar sus algoritmos de evaluación sin mayores modificaciones.



## Capítulo 5

# Experimentación y resultados

El estudio de Sistemas de Recomendaciones trae consigo un alto monto de experimentación. De hecho durante el desarrollo de este proyecto se hallaron distintos trabajos donde sus autores afirman que no es posible encontrar una configuración para este tipo de sistemas que mágicamente permita obtener siempre los mejores resultados en todos los ambientes y dominios. Por este motivo, es necesario hacer una fase de evaluación de la solución previa a su puesta en producción. Esta evaluación permitirá tomar decisiones respecto al producto desarrollado, pues dependiendo de ciertos elementos, como el dominio del sistema, su objetivo o incluso características del dataset, los resultados pueden variar mucho y darle ventajas a ciertas configuraciones sobre otras. Por ejemplo, la técnica de similitud utilizada, el tamaño de la vecindad, o incluso el algoritmo de predicciones, pueden presentar muy buenos resultados en un entorno y al mismo tiempo ser un completo fracaso en otro. De ahí que diferentes autores se han enfocado en presentar mecanismos de evaluación de sistemas de recomendaciones y muchos de ellos coinciden en que esta evaluación debe realizarse en dos pasos: inicialmente con un test offline donde las pruebas se hacen únicamente con los datos contenidos en el dataset, y posteriormente con un test online donde se utilizan usuarios reales del sistema para corroborar los resultados de la experimentación offline. De la misma manera, para este proyecto en particular la evaluación busca determinar si eventualmente el incluir contexto con alguna de las ecuaciones planteadas en el capítulo 3 genera mejores resultados que los algoritmos de recomendaciones clásicos, e incluso, identificar si alguna de dichas ecuaciones sobresale con respecto a las otras.

En consecuencia a lo anterior, este capítulo está enfocado en el proceso de evaluación del mecanismo de inclusión de contexto presentado en el capítulo 3 a partir del piloto del sistema de recomendaciones construido en el capítulo 4 y la librería descrita en el anexo B. Para ello, inicialmente se presentará un listado de los algoritmos que serán evaluados, enseguida se indicarán las métricas de evaluación utilizadas para posteriormente enseñar las pruebas realizadas junto con sus resultados y su respectivo análisis, tanto en experimentación offline como online. Algo que es importante enfatizar es que para la experimentación online fue necesario utilizar el aplicativo construido como piloto del sistema presentado en el capítulo 4, por consiguiente, esta sección también incluye la descripción del mismo. Por último, se finaliza con algunas conclusiones respecto a la evaluación.

## 5.1. Definición de algoritmos a evaluar

El capítulo 3 estuvo enfocado en la generación de un modelo contextual que pudiera ser tenido en cuenta para la construcción de un sistema de recomendaciones de publicidad ubicua. Adicionalmente, en el mismo capítulo se presentaron algunas ecuaciones que fueron consideradas en el momento de crear los algoritmos que permiten involucrar dicho modelo dentro del SR. Ahora bien, aquellas ecuaciones fueron construidas a partir de los algoritmos de filtrado colaborativo clásico user-user e item-item y dieron como resultado las ecuaciones 3.2 y 3.4 respectivamente. Este par de funciones matemáticas pueden considerarse como una inclusión de contexto que combina tanto esta información, como la similitud de usuarios e ítems, y se han convertido en la primera aproximación que se pretende evaluar. Sin embargo, también resulta interesante determinar que pasa si se tiene en cuenta únicamente el contexto y se pasa por alto la información de similitud de usuarios e ítems, las ecuaciones anteriores podrían modificarse dando como resultado las funciones de predicción de ratings 5.1 y 5.2 para usuarios e ítems respectivamente.

$$p_{u,i} = \frac{\sum_{u' \in N} \eta(c, c_{r_{u',i}}) r_{u',i}}{\sum_{u' \in N} |\eta(c, c_{r_{u',i}})|} \quad (5.1)$$

$$p_{u,i} = \frac{\sum_{i' \in I} \eta(c, c_{r_{u,i'}}) r_{u,i'}}{\sum_{i' \in I} |\eta(c, c_{r_{u,i'}})|} \quad (5.2)$$

Como puede observarse, estas funciones reflejan un enfoque similar al antes planteado, pero esta vez se reemplazan las similitudes de usuarios e ítems por las similitudes de contexto. Sobra recordar que las variables de contexto para cada enfoque son diferentes, pues cuando se trata de user-user sólo es posible considerar el ambiente del consumidor y cuando se trata de ítem-ítem, es el ambiente del proveedor el que tiene sentido ser considerado. De esta manera, estas últimas se convierten en algoritmos que serán evaluados. Por último, es importante tener un punto de referencia para hacer comparaciones, y es por esto que los algoritmos clásicos user-user e item-item también se incluirán en la evaluación.

Desde otro punto de vista, en la sección 3.3.2.3 se concluyó que incluir la información de contexto al calcular la similitud de usuarios e ítems podría generar cambios en dichos cálculos. Por lo tanto, se podría considerar esta como una variación de los algoritmos que incluyen contexto. De igual forma, otra variación viene del apartado “*Similitud de contextos incluyendo la similitud de los usuarios o ítems*” de la sección 3.4.1 donde se indica que dentro de las distancias utilizadas para calcular el contexto puede incluirse la similitud entre usuarios e ítems, dependiendo el caso. En conclusión, los algoritmos que incluyen contexto pueden evaluarse a partir de tres variaciones: **A) Simple**, donde se consideran los algoritmos tal como fueron pensados inicialmente; **B) Con contexto en la similitud de usuarios e ítems:**  $s(\eta)$ , donde el cálculo de similitud se hace incluyendo la información contextual; y **C) Con similitud de usuarios e ítems dentro del contexto:**  $\eta(s)$ , donde los valores de similitud de

## 5.2. Métricas de evaluación

---

usuarios e ítems son considerados en el momento de calcular la similitud del contexto.

En resumen los algoritmos que serán evaluados son los siguientes:

- **Filtrado colaborativo clásico user-user (user-user):** no incluye información contextual y es considerado como punto de referencia para los algoritmos basados en filtrado user-user.
  
- **Filtrado colaborativo Context-User (context-user):** Es un algoritmo que combina tanto la similitud entre contextos como la similitud clásica entre usuarios. Será evaluado en sus variaciones A, B y C.
  
- **Algoritmo usando sólo contexto basado en usuarios (context-based-user):** Es un algoritmo donde sólo se tiene en cuenta el contexto basado en el ambiente del consumidor. Será evaluado en sus variaciones A, B y C.
  
- **Filtrado colaborativo clásico item-item (item-item):** no incluye información contextual y es considerado como punto de referencia para los algoritmos basados en filtrado item-item.
  
- **Filtrado colaborativo Context-Item (context-item):** Es un algoritmo que combina tanto la similitud entre contextos como la similitud clásica entre ítems. Será evaluado en sus variaciones A, B y C.
  
- **Algoritmo usando sólo contexto basado en ítems (context-based-item):** Es un algoritmo donde sólo se tiene en cuenta el contexto basado en el ambiente del proveedor. Será evaluado en sus variaciones A, B y C.

## 5.2. Métricas de evaluación

Cuando se habla de evaluar un sistema siempre surge una pregunta referente a los aspectos que deben ser evaluados del mismo. En el caso de los sistemas de recomendaciones, comúnmente la evaluación se hace en dos aspectos: uno relacionado con la calidad de las recomendaciones, es decir, que tan bien el sistema recomienda ítems; y otro referente al rendimiento del sistema, o en otras palabras, que tan rápido el sistema es capaz de recomendar. Este último aspecto depende en gran medida de los atributos del equipo hardware en el que se encuentra alojado el software y además, siempre es posible utilizar técnicas de procesamiento distribuido, clustering, entre otras, para mejorar este valor. Por otro lado, la calidad de las recomendaciones puede depender de aspectos como el algoritmo utilizado, el dataset, la configuración de los módulos, entre otras; por este motivo, en este proyecto la evaluación se enfoca principalmente en la calidad de las recomendaciones y para ello se ha recurrido a dos métricas.

### 5.2.1. RMSE: Root Mean Square Error

Esta es tal vez la métrica más utilizada para evaluar la precisión de las predicciones que realiza el sistema. Para medirla, el sistema debe generar predicciones de calificaciones para un conjunto de pares usuario-ítem para los cuales ya se conoce el rating real [105]. Comúnmente este rating es conocido porque ha sido apartado del dataset final en un experimento offline, o porque se ha obtenido previamente en un experimento online. El valor de RMSE entre un rating predicho  $\hat{r}_{ui}$  y un rating real  $r_{ui}$  para un conjunto de datos  $\mathcal{T}$  puede calcularse a partir de:

$$RMSE = \sqrt{\frac{1}{|\mathcal{T}|} \sum_{(u,i) \in \mathcal{T}} (\hat{r}_{ui} - r_{ui})^2} \quad (5.3)$$

Con el objetivo de incluir la información contextual, la ecuación debe tener una pequeña modificación que permita la consideración de un valor de contexto  $c$ , por lo tanto, el RMSE puede ser calculado de la siguiente manera:

$$RMSE = \sqrt{\frac{1}{|\mathcal{T}|} \sum_{(u,i,c) \in \mathcal{T}} (\hat{r}_{uic} - r_{uic})^2} \quad (5.4)$$

Debe notarse que RMSE mide la diferencia entre la realidad y la predicción y por tanto se espera que entre mejor sea el recomendador, menores sean los valores de RMSE.

### 5.2.2. Precisión

En muchas ocasiones no sólo resulta interesante conocer que tan preciso es el sistema para predecir las calificaciones del usuario sobre los ítems, sino también la capacidad del sistema para recomendar ítems que son interesantes para el usuario aún cuando este no los conoce. Cómo se muestra en la tabla 5.1, los ítems que le son recomendados a un usuario pueden clasificarse en cuatro grupos acorde al punto de vista del usuario. De modo que los ítems que son recomendados y al mismo tiempo representan utilidad para el usuario son considerados verdaderos-positivos, por otro lado aquellos que no representan utilidad se consideran falsos-positivos. La relación entre verdaderos-positivos y falsos positivos permite medir la precisión del sistema, de tal modo, que la convierte en una métrica que indica que tan bueno es el sistema para recomendar ítems que sean realmente atractivos para el usuario. La ecuación 5.5 permite calcular los valores de precisión.

$$precision = \frac{\#tp}{\#tp + \#fp} \quad (5.5)$$

### 5.3. Definición del plan de pruebas

	Recommended	No Recommended
Used	True Positive (tp)	True Negative (tn)
Unusued	False Positive (fp)	False Negative (fn)

Tabla 5.1: Clasificación de los posibles resultados de una recomendación de un ítem a un usuario Fuente: adaptación de [105]

### 5.3. Definición del plan de pruebas

A partir de lo anterior se ha definido un plan de pruebas para la evaluación de cada uno de los algoritmos definidos en la sección 5.1. El plan de pruebas se encuentra dividido en dos secciones, en primer lugar se realizarán tests offline utilizando únicamente la información contenida en el dataset, y en segundo lugar se realizarán tests online con usuarios reales. Además, en cada una de las secciones se evaluarán tanto la métrica RMSE, como la métrica de Precisión.

#### 5.3.1. Pruebas Offline

Las pruebas offline de sistemas de recomendaciones comúnmente se hacen a partir del particionado del dataset en dos grupos de calificaciones de la forma  $R \rightarrow U \times I \times C$ . El primer grupo es conocido como *dataset test* o en español, pruebas, y es una sección del dataset que se esconde de las evaluaciones y que se utiliza para determinar aquellos ratings que son conocidos; el segundo grupo es el *dataset training* o entrenamiento, y a partir de él se predicen los ratings del de pruebas. Dicho de otro modo, para calcular las métricas se comparan las predicciones generadas a partir el dataset de entrenamiento con los valores reales obtenidos de pruebas. Para este proyecto se tomará un dataset de pruebas del 25 % del tamaño real, es decir, un entrenamiento del 75 % del dataset. Adicionalmente, Mahout cuenta con las clases necesarias para realizar el particionado de forma aleatoria, y para evitar que se generen cambios en alguna de las partes se duplicará del dataset eliminando aquellos ratings que hacen parte de pruebas. Luego, con el objetivo de medir RMSE se pedirá al sistema que prediga los ratings eliminados para compararlos con los valores reales. Por otro lado, para la medida de precisión se ubicará un umbral que permita considerar los verdaderos-positivos como aquellos que se encuentren con una calificación por encima de él, este umbral será de 2,7, seguidamente se solicitará al sistema recomendar algunos ítems y se verificará cuántos de ellos hacen parte del grupo de verdaderos positivos.

#### 5.3.2. Pruebas Online

Para realizar las pruebas online se ha utilizado el aplicativo construido como piloto del sistema diseñado en el capítulo 4. A continuación se describe el aplicativo desarrollado.

##### 5.3.2.1. Aplicativo construido para la evaluación online de algoritmos de recomendaciones

La descripción del aplicativo construido para la evaluación online de los algoritmos de recomendaciones se realizará a partir del diagrama de procesos presentado en la figura ??, donde se indica uno a

uno los pasos que guiarán al usuario a través del experimento.

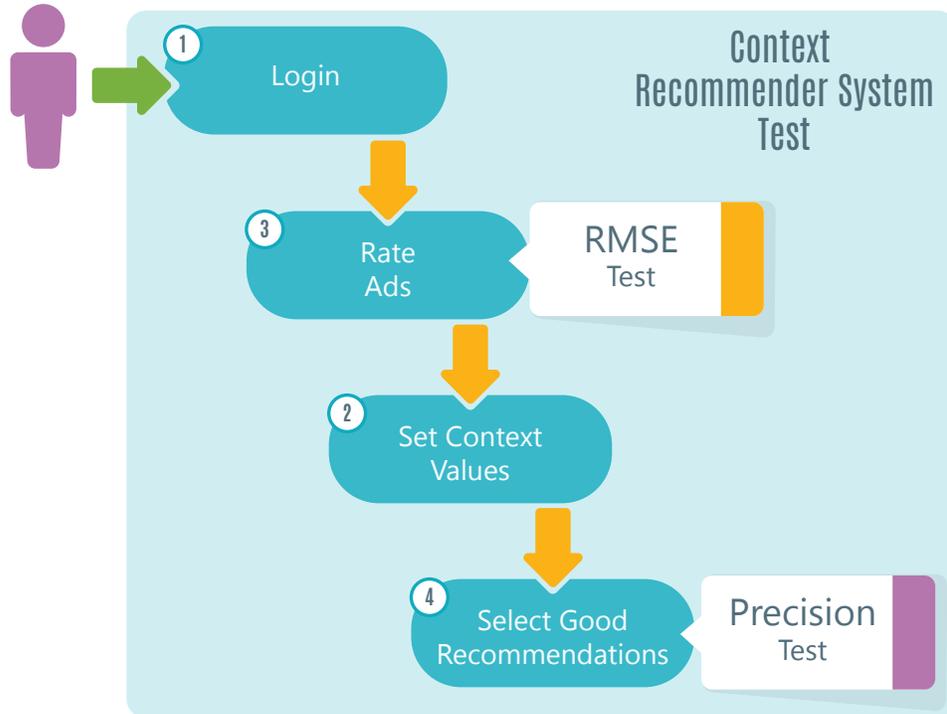


Figura 5.1: Diagrama de procesos para evaluación online Fuente: creación propia

**Fases del experimento para la prueba online** Las fases presentadas en la figura se describen a continuación:

### 1. Login

Se espera que sólo usuarios que hayan participado en el experimento de creación del dataset descrito en el capítulo 3 participen también en este experimento. Por este motivo, existe una interfaz de *Login* donde a partir del nombre de usuario y contraseña antes utilizados para el registro en el sistema, el usuario puede ser identificado e inicie el experimento. Esta interfaz es similar a la utilizada para iniciar sesión en el experimento de creación del dataset ver figura 3.9. En caso de no encontrar al usuario en la base de datos, el sistema presentará un mensaje de error y revocará el inicio de sesión. Adicionalmente, El usuario puede intentar recordar su contraseña, en caso de haberla olvidado. Para ello ingresará el correo utilizado en el registro y los datos le serán enviados vía email.

## 5.3. Definición del plan de pruebas

---

### 2. Fijar contexto

Al igual que en el experimento de creación del dataset, se espera que el usuario ingrese de forma explícita los valores para cada una de las variables de contexto. Una vez el usuario a ingresado estos valores, el sistema busca en la base de datos si existe un contexto con los valores indicados por el usuario, si el contexto existe, el sistema lo fija con dicho Id. De lo contrario, crea uno nuevo. Esta interfaz es similar a la utilizada para fijar el contexto en el experimento de creación del dataset ver figura 3.10

### 3. Calificación de anuncios: Prueba RMSE

Una vez el usuario ha fijado su contexto inicia la prueba RMSE, para ello debe calificar 10 anuncios que le son presentados. El sistema antes de presentarlos genera predicciones para dichos anuncios con cada uno de los algoritmos y sus variaciones, esta información luego será utilizada para comparar las calificaciones reales con las predichas. Esta interfaz es similar a la utilizada para calificar en el experimento de creación del dataset ver figura 3.11

### 4. Selección de anuncios relevantes: Prueba de Precisión

En este paso se espera contribuir con la prueba de precisión. Para ello, el sistema generará recomendaciones de 15 anuncios utilizando cada uno de los algoritmos definidos en el sección 5.1 y el usuario deberá indicar cuales de ellos considera que son una buena recomendación (verdaderos positivos) y cuales son una muy mala recomendación. La imagen de la figura 5.2 presenta la interfaz de selección de anuncios relevantes. En ella puede observarse que cada anuncio cuenta con tres botones: el botón verde para indicar que es relevante, el botón blanco para indicar que le es indiferente y el botón amarillo para indicar que es una muy mala recomendación.

Finalmente es importante mencionar que para la construcción del aplicativo se utilizaron las herramientas: Java Enterprise Edition<sup>1</sup>, Java Server Faces<sup>2</sup> para la capa de presentación soportada con la librería Primefaces<sup>3</sup>, y el dataset se encuentra alojado en una base de datos MySQL<sup>4</sup>. Finalmente, se utilizó también la librería *Mahout-Context-Recommender* creada como aporte de este proyecto para la generación de SRs basados en contexto.

---

<sup>1</sup>JEE: es la versión empresarial de JAVA y es construido bajo la contribución de expertos de la industria. Ofrece interfaces de usuario enriquecidas, alto rendimiento, versatilidad y portabilidad [106]

<sup>2</sup>JSF: esta tecnología establece el estándar para construir interfaces de usuario de lado del servidor [95]

<sup>3</sup>Primefaces: es una librería de código abierto con componentes para soportar interfaces de usuario en JSF. Se distribuye bajo licencia apache [107]

<sup>4</sup>MySQL Server: es tal vez el motor de base de datos de código abierto más popular, ofrece alto rendimiento y escalabilidad [108]

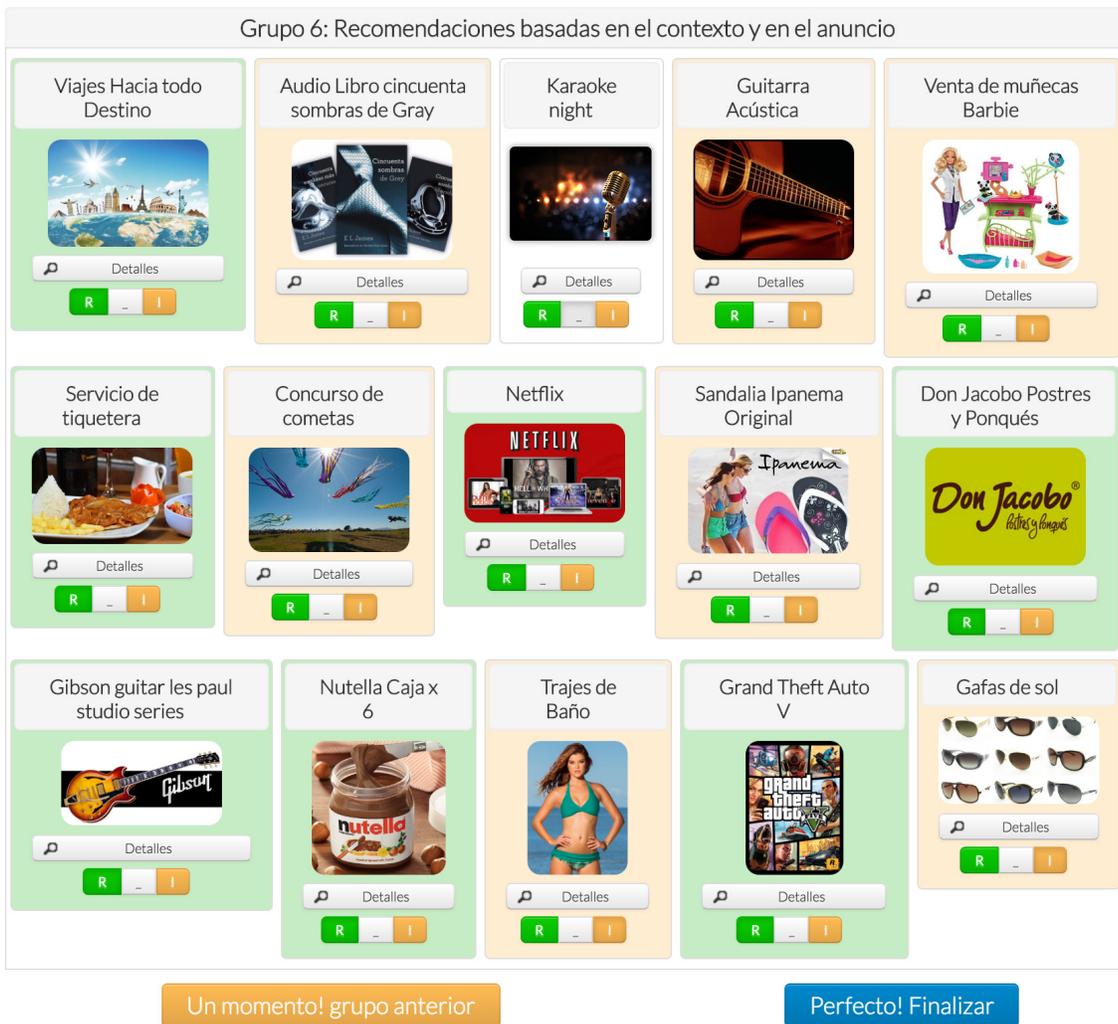


Figura 5.2: Interfaz de métrica de precisión Fuente: creación propia

## 5.4. Resultados y análisis del plan de pruebas

Después de definir el plan de pruebas que será llevado a cabo, el siguiente paso es ejecutar dicho plan. A continuación se describen los resultados obtenidos.

### 5.4.1. Resultados de la prueba offline

#### 5.4.1.1. Métrica RMSE

La primera prueba offline llevada a cabo estuvo relacionada con la métrica RMSE. La figura 5.3 muestra la diferencia entre los resultados del algoritmo de filtrado colaborativo clásico user-user y aquellos que se basan en éste pero a su vez incluyen contexto, es decir, las diferencias calculadas a partir de las ecuaciones 5.6 y 5.7. Por otro lado, la figura 5.4 presenta la diferencia de los algoritmos basados en filtrado colaborativo item-item, ó en otras palabras, las diferencias calculadas a partir de las ecuaciones 5.8 y 5.9.

$$RMSE_{cbu-u} = RMSE_{context-based-user} - RMSE_{user-user} \quad (5.6)$$

$$RMSE_{cu-u} = RMSE_{context-user} - RMSE_{user-user} \quad (5.7)$$

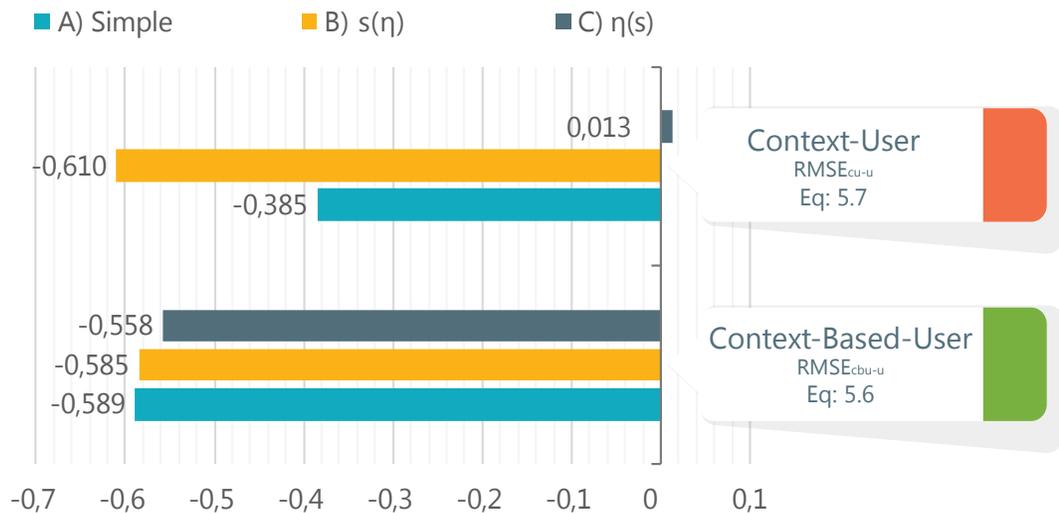


Figura 5.3: Resultados de la prueba RMSE offline para algoritmos user-user Fuente: creación propia

$$RMSE_{cbi-i} = RMSE_{context-based-item} - RMSE_{item-item} \quad (5.8)$$

$$RMSE_{ci-i} = RMSE_{context-item} - RMSE_{item-item} \quad (5.9)$$

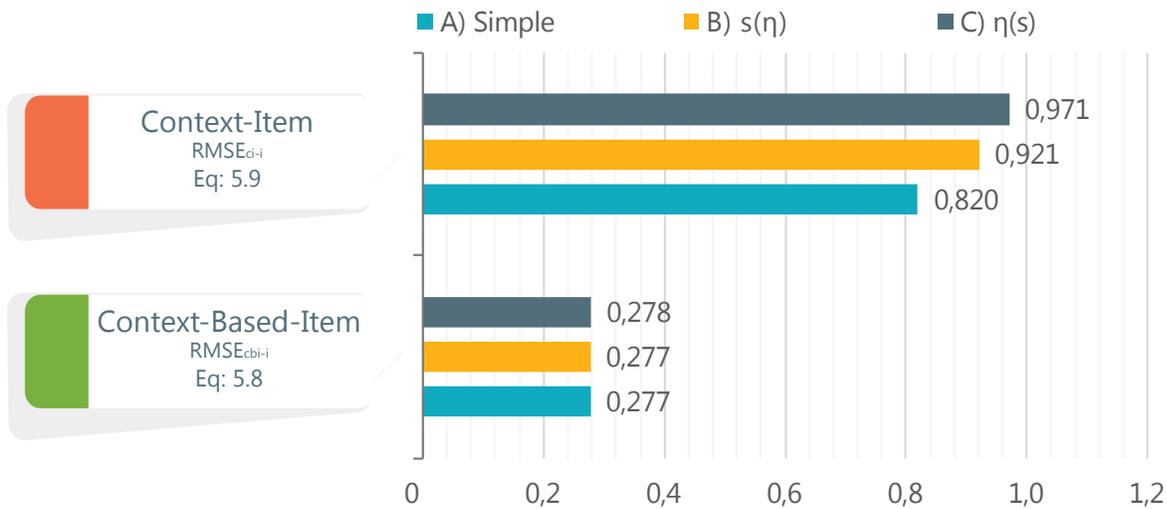


Figura 5.4: Resultados de la prueba RMSE offline para algoritmos item-item Fuente: creación propia

En los resultados de la prueba RMSE, es de apreciarse que el contexto puede mejorar la precisión de las predicciones en algoritmos basados en usuarios, ya que este genera una disminución en los resultados en casi todos los casos. No obstante, esta afirmación no es cierta en los algoritmos basados en ítems donde al parecer, agregar información de contexto empeora los resultados. Respecto a las variaciones, se ha determinado que incluir la similitud de usuarios o ítems al calcular la similitud entre contextos genera un deterioro en los valores de RMSE. Sin embargo, incluir el contexto dentro del cálculo de similitud de usuarios o ítems, no parece tener efectos relevantes en los resultados, aunque en algunos casos, como en el algoritmo context-user, puede resultar en una mejora. Siendo esto así, se ha determinado desechar la variación C ( $\eta(s)$ ) y continuar con las variaciones A y B para la prueba offline de precisión.

#### 5.4.1.2. Métrica de Precisión

Los gráficos de las figuras 5.5 y 5.6 muestran la diferencia porcentual de la precisión entre los algoritmos que incluyen contexto y los filtrados colaborativos clásicos, es decir, los resultados de las ecuaciones 5.10, 5.11, 5.12 y 5.13. En ellas puede notarse que los mejores resultados se obtuvieron con el algoritmo Context-Based-User, el cual, además pudo obtener recomendaciones relevantes

#### 5.4. Resultados y análisis del plan de pruebas

para 33 usuarios, muy por encima de otros algoritmos. Asimismo se puede observar una mejora en algunos algoritmos al incluir el contexto dentro de la similitud de usuarios. También es importante notar que el algoritmo Context-Based-User generó una mejora en la precisión de cerca del 100 % comparada con el filtrado user-user clásico, es decir que la precisión del algoritmo fue de más del doble que la de un filtrado colaborativo clásico. Por otra parte, en los algoritmos item-item la consideración de contexto parece desmejorar los resultados de precisión en la mayoría de los casos, esto puede deberse a que las calificaciones para los ítems se encuentran muy dispersas en términos de los contextos para los cuales fueron expresadas, y por tanto al existir calificaciones en contextos muy distintos se pueden disminuir los casos en los que un ítem relevante es recomendado y afectar directamente la métrica de precisión. Sin embargo, este es un problema que puede deberse principalmente a la información del dataset y es posible que en un análisis online se mejoren los resultados.

$$P_{cbu-u} = \frac{Precision_{context-based-user} - Precision_{user-user}}{Precision_{user-user}} \times 100 \quad (5.10)$$

$$P_{cu-u} = \frac{Precision_{context-user} - Precision_{user-user}}{Precision_{user-user}} \times 100 \quad (5.11)$$

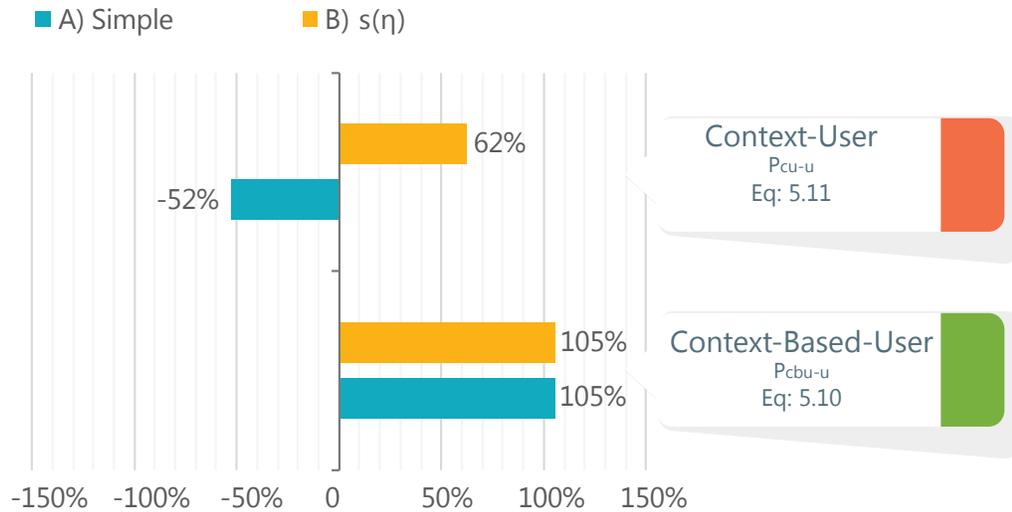


Figura 5.5: Resultados de la prueba de Precisión offline para algoritmos user-user Fuente: creación propia

$$P_{cbi-i} = \frac{Precision_{context-based-item} - Precision_{item-item}}{Precision_{item-item}} \times 100 \quad (5.12)$$

$$P_{ci-i} = \frac{Precision_{context-item} - Precision_{item-item}}{Precision_{item-item}} \times 100 \quad (5.13)$$

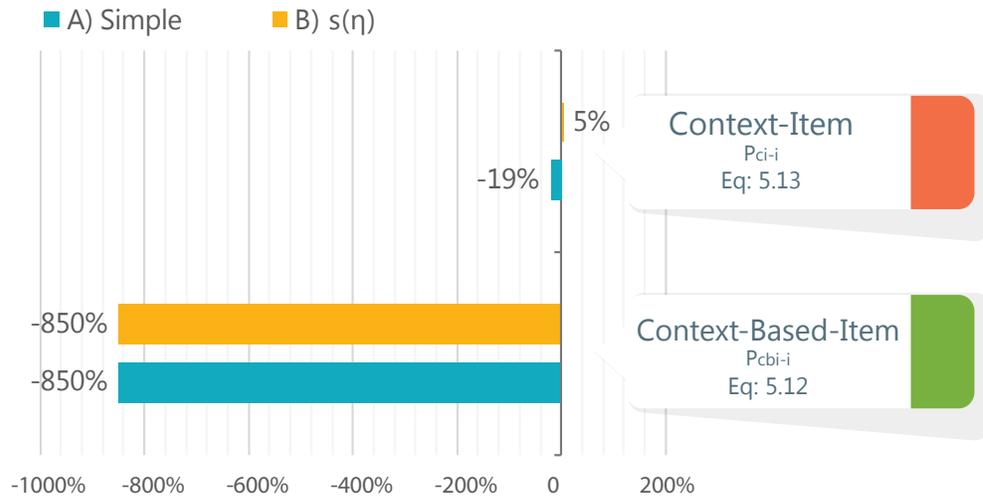


Figura 5.6: Resultados de la prueba de Precisión offline para algoritmos item-item Fuente: creación propia

Estos resultados conllevan a seleccionar la variación B para las pruebas online, es decir, contexto incluido en la similitud de usuarios e ítems. Se espera que en dichas pruebas se obtengan mejores valores de precisión y RMSE expresados por el usuario.

## 5.4.2. Resultados de la prueba online

El experimento online se llevó a cabo con 15 usuarios, los cuales pasaron por cada una de las fases planteadas. Al igual que en el experimento offline se tuvieron en cuenta las métricas RMSE y precisión. A continuación se presentan los resultados para cada una.

### 5.4.2.1. Métrica RMSE

La tabla 5.2 muestra los resultados de la prueba RMSE.

Al analizar los resultados obtenidos se encontró que:

## 5.4. Resultados y análisis del plan de pruebas

Algoritmo	RMSE <sub>Promedio</sub>	DESVIACIÓN <sub>TÍPICA</sub>
User-User	1,506	0,666
Context-Based-User	1,213	0,562
Context-User	1,422	0,342
Item-Item	2,088	0,723
Context-Based-Item	1,483	0,726
Context-Item	3,797	0,871

Tabla 5.2: Resultados de la métrica RMSE en la prueba online Fuente: creación propia

- Los algoritmos basados en filtrado colaborativo User-user siempre tienen un mejor resultado en cuanto al cálculo de las predicciones para este dataset específico, que los algoritmos basados en un filtrado item-item.
- Tanto el algoritmo Context-User como el Context-Based-User presentan mejoras en RMSE indicando que el contexto puede aumentar la precisión de las predicciones cuando se combina con un enfoque de un filtrado colaborativo user-user.
- Particularmente el algoritmo ContextBasedUser presenta el mejor resultado en RMSE.
- Los algoritmos item-item para este dataset no parecen tener muy buenos resultados y el incluir contexto puede resultar impredecible. En algunos casos como en Context-Based-Item pueden mejorar mucho, pero en otros como Context-Item puede ser contraproducente.

Aunque las conclusiones hasta este punto parecen muy alentadoras, sobre todo en los algoritmos user-user, es imprescindible realizar un análisis estadístico para determinar si efectivamente dichas mejoras son significativas o son producto de resultados aleatorios. Para realizar dicho análisis estadístico se llevaron a cabo los siguientes pasos:

1. **Definición de la hipótesis nula y su negación:** se partió de una hipótesis nula  $H_0$ : “No hay diferencia significativa en los resultados RMSE antes y después de aplicar context”, por su parte la hipótesis que permitiría negar la hipótesis nula es  $H_1$ : “Hay diferencia significativa en los resultados de RMSE antes y después de aplicar contexto”.
2. **Definición del nivel alpha ( $\alpha$ ):** El valor de alpha seleccionado es del 5%;  $\alpha = 0,05$ . Si al realizar la prueba de significancia se tienen valores menores que  $\alpha$ , se rechazará la hipótesis nula y se aceptará  $H_1$ .
3. **Elección de la prueba:** Dado a que se trata de unas muestras relacionadas utilizamos un estudio longitudinal donde nuestra variable fija es el utilizar o no el contexto. Finalmente nuestra variable aleatoria es el promedio de las diferencias entre la predicción y la realidad para cada usuario. Esta es una variable numérica, lo que indica que nuestro prueba debe ser un prueba **T-Student (muestras relacionadas)**.

Parejas de algoritmos comparados			p-value
Par 1	user-user	context-based-user	0,017
Par 2	user-user	context-user	0,625
Par 3	item-item	context-based-item	0,075
Par 4	item-item	context-based-user	0,000
Par 5	user-user	item-item	0,061

Tabla 5.3: Resultados de prueba T-Student para valores RMSE Fuente: creación propia

4. **Verificación del supuesto de normalidad:** la muestra cuenta con menos de 30 individuos, por este motivo el supuesto de normalidad fue evaluado a partir de una prueba de Chapiro-Wilk donde se estableció como criterio que al obtener un  $pvalor \geq \alpha$  se aceptaría la hipótesis de que los datos provienen de una distribución normal. Para dicha prueba se utilizó el software estadístico SPSS[109] y los resultados mostraron que para todos los algoritmos  $pvalor > \alpha$ , concluyendo que los datos provienen de una distribución normal y permitiendo utilizar la prueba de muestras relacionadas T-Student.
5. **Ejecución de la prueba T-Student:** Finalmente se llevó a cabo la prueba para determinar la significancia de los resultados. Utilizando nuevamente el software SPSS se obtuvieron los valores de significancia consignados en la Tabla 5.3. Estos resultados permiten concluir que existe evidencia estadística para asegurar que existe una diferencia significativa entre los resultados de la prueba de RMSE para el algoritmo clásico user-user y el algoritmo context-based-user, indicando que incluir información contextual en el filtrado user-user mejora significativamente las predicciones. Por su parte también se encontró diferencia significativa entre los algoritmos item-item y context-item, lo que permite afirmar que en el caso del filtrado item-item la información de contexto podría empeorar las predicciones. Para los demás pares de algoritmos, no se puede determinar con certeza si la diferencia es estadísticamente significativa.

Como último paso, dados los buenos resultados del algoritmo context-based-user se decidió hacer nuevamente la prueba de significancia pero esta vez comparando este algoritmo con las demás aproximaciones. Los resultados consignados en la tabla 5.4 mostraron que existe una diferencia significativa entre los resultados de la prueba RMSE de este algoritmo con todas las demás aproximaciones, excepto con context-user, indicando que esta aproximación junto con context-based-user podrían generar las mejores predicciones.

#### 5.4.2.2. Métrica de Precisión

La tabla 5.5 resume los resultados obtenidos en la prueba de precisión. En ella se puede apreciar en una primera impresión datos congruentes con el resultado de la prueba de RMSE, se esperaba que el algoritmo user-user tuviese una mejor precisión que el item-item. Adicionalmente, era de esperarse que el incluir contexto en el algoritmo user-user mejorara la precisión. En cuanto a los algoritmos item-item los resultados parecen mostrar una mejora al incluir el contexto. Sin embargo,

## 5.4. Resultados y análisis del plan de pruebas

Parejas de algoritmos comparados			p-value
Par 1	context-based-user	user-user	0,017
Par 2	context-based-user	context-user	0,224
Par 3	context-based-user	item-item	0,010
Par 4	context-based-user	context-based-item	0,024
Par 5	context-based-user	context-item	0,000

Tabla 5.4: Resultados de prueba T-Student para valores de RMSE de context-based-user Fuente: creación propia

Algoritmo	Precisión	Desviación típica
User-User	0,41	0,15
Context-Based-User	0,52	0,20
Context-User	0,51	0,18
Item-Item	0,28	0,20
Context-Based-Item	0,33	0,19
Context-Item	0,34	0,22

Tabla 5.5: Resultados de la métrica de Precisión en la prueba online Fuente: creación propia

al igual que en la prueba RMSE es necesario realizar un análisis estadístico que permita determinar la significancia de los resultados.

Para realizar el análisis estadístico se siguieron los mismos pasos desarrollados en la sección 5.4.2.1. En este caso la hipótesis nula planteada es  $H_0$ : “No hay diferencia significativa en la precisión antes y después de aplicar contexto” y la hipótesis  $H_1$ : “Hay diferencia significativa en la precisión antes y después de aplicar contexto” que permite rechazar la hipótesis nula. Se definió nuevamente un valor  $\alpha$  del 5% y la verificación de normalidad usando Chapiro-Wilk indicó que todas las muestras provienen de una distribución normal, permitiendo usar la prueba T-Student para determinar la significancia. Esta vez los resultados de significancia mostraron evidencia estadística que indica que hay diferencia significativa entre el algoritmo context-user y user-user, también entre context-based-user y user-user, indicando que un filtrado colaborativo user-user puede mejorar su precisión en las recomendaciones mediante la inclusión de la información de contexto. En cuanto a los algoritmos item-item, los resultados no fueron concluyentes, aunque cabe aclarar que se demostró diferencia significativa en la precisión del algoritmo clásico user-user e item-item. Nuevamente fue interesante determinar los resultados del algoritmo context-based-user comparado con los demás. Los resultados fueron resumidos en la tabla 5.6 y nuevamente se encontró una evidencia para asegurar que este algoritmo posee los mejores resultados en cuanto a la precisión de las recomendaciones.

Parejas de algoritmos comparados			p-value
Par 1	context-based-user	user-user	0,035
Par 2	context-based-user	context-user	0,897
Par 3	context-based-user	item-item	0,001
Par 4	context-based-user	context-based-item	0,000
Par 5	context-based-user	context-item	0,003

Tabla 5.6: Resultados de prueba T-Student para valores de Precisión de context-based-user Fuente: creación propia

## 5.5. Conclusiones del capítulo

- Este capítulo estuvo enfocado a la validación de los algoritmos generados en capítulo 3. Dicha validación se realizó en dos etapas: primero se hicieron unas pruebas offline usando métricas de precisión y RMSE. Estas pruebas permitieron descartar la variación de usar la similitud de ítems o usuarios como parte del contexto, pues desmejoraba el resultado de las predicciones. Luego estas mismas pruebas mostraron que el incluir el contexto dentro del calculo de similitud entre usuarios o ítems, podría mejorar la precisión del sistema casi en un 100 %, de esta manera se descartó la variación C para la inclusión de contexto y se realizó la segunda etapa de experimentación sólo con la variación B (contexto dentro de la similitud de usuarios o ítems). En la segunda etapa se realizaron pruebas online donde 15 usuarios hicieron el proceso de calificar ítems y recibir recomendaciones a partir de los seis algoritmos: user-user e item-item clásicos, considerando tanto el contexto como la similitud de los usuarios, considerando solo un contexto en un filtrado user-user (contexto basado en el usuario), considerando tanto el contexto como la similitud de los ítems y considerando solo un contexto en un filtrado item-item (contexto basado en el ítem).
- Finalmente, Las pruebas online demostraron que los algoritmos user-user tienen un mejor resultado en cuanto a RMSE y precisión que los algoritmos item-item para este dataset específico, y además para el filtrado colaborativo user-user, el incluir el contexto genera una mejora significativa en la predicción y en las recomendaciones. Los valores de precisión en la prueba online para los algoritmos context-user y context-based-user estuvieron por encima del 50 % lo que representa un muy buen valor en términos de medir la calidad del sistema de recomendaciones. Por su parte, la aproximación Context-Based-User presentó los mejores resultados en todas las pruebas y permite concluir que es la mejor manera, al menos para este dataset, de incluir información de contexto en un sistema de recomendaciones para publicidad ubicua.

## Capítulo 6

# Conclusiones y trabajo futuro

### 6.1. Conclusiones del estado actual del conocimiento

- El enfoque de vigilancia tecnológica mostró que el tema de Sistemas de recomendaciones basados en contexto está actualmente en investigación, puesto que aunque se encuentran algunos trabajos relacionados, aún es necesario realizar esfuerzos para sacar el mayor provecho de este tema. Adicionalmente, se evidenció que este tipo de sistemas pueden ser implementados en un sinnúmero de campos.
- Los sistemas de recomendaciones para publicidad ubicua son también un tema en constante investigación. Sin embargo, los sistemas de este tipo y que además incluyan información contextual han sido muy poco explorados, hasta el punto que son muy escasos los datasets con características de contexto usables en un sistema de recomendaciones de este tipo.
- El rango de fechas consultadas está entre 1997 y 2016, dado a que muchos conceptos se remontan a hace ya más de 15 años, pero sus implementaciones siguen solucionando problemas actuales. Adicionalmente, como se mencionó se trata de un tema en constante investigación por lo que es entendible que existan proyectos que se están publicando en el presente año.
- Se logró estudiar algunos trabajos enfocados en el concepto de publicidad ubicua, muchos de ellos se quedan en el concepto y no presentan una implementación de la misma. Algunos aunque si implementan sistemas de publicidad ubicua no se presentan como sistemas de recomendaciones y tampoco incluyen la información del contexto.
- Se estudiaron trabajos que presentan la importancia de utilizar contexto dentro de los sistemas de recomendaciones. Sin embargo, sólo se enfocan en enseñar el concepto de reducción de dimensionalidad como un mecanismo para incluir más variables dentro del espacio clásico  $R : Usuario \times Item$  agregando una nueva dimensión  $R : Usuario \times Item \times Contexto$  y no se expresa exactamente las variables de contexto utilizadas.

- Muchos trabajos se enfocan en la validación del mejor momento para incluir el contexto, en un prefiltro o en un postfiltro. Otros presentan la posibilidad de utilizar similitud entre contextos, pero no expresan un mecanismo exacto para medir dicha similitud y se limitan a determinar si el concepto es válido, dejando estáticos los valores de similitud para un conjunto de contextos predefinidos. Asimismo, Los pocos trabajos que muestran implementaciones de estos sistemas sólo utilizan una única variable y por tanto no especifican el dominio completo de las mismas.

### 6.2. Conclusiones de contexto en entornos de publicidad ubicua y su implantación en Sistemas de recomendaciones

- Dado a que existen distintas definiciones para el contexto, se debe hacer un estudio a fondo de lo que cada definición describe para llegar a una que se adapte completamente a los requerimientos del sistema. Para este caso particular se basó en la definición dada por Schillit en la que se establece que el contexto se compone de tres aspectos “dónde estás”, “con quién estás” y “con qué recursos cuentas”.
- Aunque existen algunos modelos de contexto para entornos de publicidad, no han sido pensados para ser usados en Sistemas de Recomendaciones. En este trabajo se hizo una adaptación del modelo de Bauer y Spiekerman pensado para un caso específico de publicidad ubicua conocida como digital signage. Las principales adaptaciones consistieron en eliminar los niveles macro y micro, pues la publicidad es principalmente situacional; y en la agregación de la variable “interacción con el sistema”.
- El incluir nuevas variables a un sistema de recomendaciones sugiere migrar del espacio bidimensional  $U \times I$  a un espacio multidimensional donde cada variable de contexto se convierte en una nueva dimensión que debe ser tomada en cuenta. Adomavicius ha planteado una alternativa para la solución de la nueva matriz de factorización, mediante una aproximación de reducción de dimensionalidad que permite regresar a una matriz bidimensional y aplicar algoritmos de filtrado colaborativo clásico.
- Aunque existen tres posibles opciones para hacer la reducción de dimensionalidad, estudios previos demostraron que una aproximación basada en modelo puede generar mejores resultados respecto a la métrica de precisión, en la mayoría de los casos. Para este trabajo se realizaron algunas pruebas usando prefiltrado y postfiltrado, obteniendo con ellas resultados poco relevantes. Dichas conclusiones combinadas con afirmaciones de otros autores dieron paso a la selección de una aproximación basada en modelo como mecanismo de inclusión de contexto dentro del sistema de recomendaciones.
- Se logró generar una adaptación de las ecuaciones de filtrado colaborativo clásico user-user e item-item para agregar en ellas la información contextual. Al hacer estas adaptaciones se encontró que para un filtrado user-user no es posible incluir información del ambiente del

## 6.2. Conclusiones de contexto en entornos de publicidad ubicua y su implantación en Sistemas de recomendaciones

---

proveedor, pues en este caso los ítems se dejan constantes mientras se evalúan las similitudes entre los usuarios. En el caso del filtrado item-item ocurre lo contrario y no es posible incluir información del ambiente del consumidor. El ambiente físico por su parte, puede ser tenido en cuenta en ambos filtrados.

- Algunos ejercicios demostraron que tener en cuenta el contexto puede cambiar por completo el nivel de similitud entre los usuarios y que estos resultados pueden hacerse extensibles a la similitud entre ítems. Por este motivo se planteó otra manera de incluir contexto en el modelo afectando directamente la ecuación encargada de medir similitud entre usuarios o ítems.
- La aproximación basada en modelo para la inclusión de información contextual sugiere la necesidad de determinar un mecanismo para medir la similitud entre dos contextos. Para ello se planteó una ecuación de similitud de contextos partiendo de la distancia euclidiana entre dos objetos. Cabe resaltar que esta ecuación de similitud requiere además de las variables contextuales, pesos de importancia para cada una de ellas y, por tanto, se requiere que el dataset permita tener indicios de estos pesos o que se haga un estudio de marketing que permita definirlos con más rigor.
- Una última aproximación a la inclusión de información contextual sugerida es eliminar el factor de similitud entre usuarios de la ecuación de filtrado y en su lugar poner la similitud entre contextos, con la modificación de que esta similitud de contextos incluiría en su cálculo la similitud entre usuarios o ítems como una distancia más.
- El dataset es una de las partes más importantes en el momento de hacer pruebas a un sistema de recomendaciones. Sin embargo, para el caso específico de publicidad no son muy comunes los datasets y aún menos que también involucren información contextual. Por este motivo fue necesario diseñar un aplicativo que permitiese recolectar calificaciones sobre algunos anuncios obtenidos del dataset de SMARTA e incluir en ellos información del contexto de la calificación. El aplicativo en cuestión fue desarrollado usando tecnologías Web, contó con la participación de cerca de 100 usuarios y logró conseguir más de 1500 ratings. Dicho dataset se encuentra disponible para ser utilizado en trabajos relacionados.
- Para determinar los pesos de cada una de las variables contextuales se solicitó en el mismo aplicativo la razón primordial por la cual los usuarios daban su calificación a los anuncios, luego el peso de la variable es equivalente a la probabilidad de que dicha variable sea seleccionada como razón principal.
- El hacer varias iteraciones del experimento para reunir información contextual permitió depurar el aplicativo antes de salir a producción. Muchos de los consejos aportados por potenciales usuarios, permitieron tomar decisiones clave que ayudaron a obtener excelentes resultados en el dataset final.

### 6.3. Conclusiones de Construcción de un sistema de recomendaciones soportado en información contextual

- Tomando como base la arquitectura de un sistema de recomendaciones simple de Mahout, logró construirse una arquitectura de referencia para sistemas de recomendaciones que involucran información contextual utilizando las ecuaciones planteadas en el capítulo 3. Dicha arquitectura, puede superponerse al marco de trabajo Mahout y facilitar su utilización por parte de terceros. Adicionalmente se construyó un piloto que permite la evaluación del sistema construido, y que será descrito con detalle en el capítulo 5 donde, a partir del plan de pruebas, será más fácil de comprender.
- Dado a que se diseñó una arquitectura basada en *interfaces* de JAVA, se hace fácilmente escalable para sistemas de recomendaciones en otros entornos. De hecho, se facilita la generación algoritmos de recomendaciones utilizando contextos diferentes a los planteados durante el desarrollo de este proyecto, pues para ello sólo bastaría con extender la clase “*ContextRecommender*”, demostrando la facilidad de escalabilidad del sistema.
- Se construyó la librería “Mahout Context Recommender” como una contribución a la comunidad de desarrolladores, usuarios y otros involucrados en el marco de trabajo Mahout. Dicha librería contiene todas las clases planteadas en el diagrama 4.4 y fue diseñada para utilizarse como una extensión del framework, de forma que es fácil de utilizar, altamente escalable y adaptable a diferentes dominios. Esto puede verse evidenciado en el anexo B donde con cinco sencillos pasos puede construirse un sistema de este tipo completamente habilitado para generar recomendaciones. En cuanto a la evaluación de los sistemas desarrollados utilizando dicha librería, vale la pena destacar que guarda una estrecha relación con los módulos básicos de Mahout y por tanto, es posible utilizar sus algoritmos de evaluación sin mayores modificaciones.

### 6.4. Conclusiones de experimentación y resultados

- Se realizó una validación de los algoritmos presentados en el capítulo 3 en dos etapas: primero se hicieron unas pruebas offline usando métricas de precisión y RMSE. Estas pruebas permitieron descartar la variación de usar la similitud de ítems o usuarios como parte del contexto, pues desmejoraba el resultado de las predicciones. Luego estas mismas pruebas mostraron que el incluir el contexto dentro del calculo de similitud entre usuarios o ítems, podría mejorar la precisión del sistema casi en un 100 %, de esta manera se descartó la variación C para la inclusión de contexto y se realizó la segunda etapa de experimentación sólo con la variación B (contexto dentro de la similitud de usuarios o ítems). En la segunda etapa se realizaron pruebas online donde 15 usuarios hicieron el proceso de calificar ítems y recibir recomendaciones a partir de los seis algoritmos: user-user e item-item clásicos, considerando tanto el contexto como la similitud de los usuarios, considerando solo un contexto en un filtrado user-user (contexto basado en el usuario), considerando tanto el contexto como la similitud de los ítems y considerando solo un contexto en un filtrado item-item (contexto basado en el ítem).

## 6.5. Trabajo Futuro

---

- Las pruebas online demostraron que los algoritmos user-user tienen un mejor resultado en cuanto a RMSE y precisión que los algoritmos item-item para este dataset específico, y además para el filtrado colaborativo user-user, el incluir el contexto genera una mejora significativa en la predicción y en las recomendaciones. Los valores de precisión en la prueba online para los algoritmos context-user y context-based-user estuvieron por encima del 50 % lo que representa un muy buen valor en términos de medir la calidad del sistema de recomendaciones. Por su parte, la aproximación Context-Based-User presentó los mejores resultados en todas las pruebas y permite concluir que es la mejor manera, al menos para este dataset, de incluir información de contexto en un sistema de recomendaciones para publicidad ubicua.

## 6.5. Trabajo Futuro

- Es necesario evaluar el sistema utilizando un dataset distinto al construido aquí para determinar nuevos hallazgos que puedan presentarse. Dichos datasets deben contener información contextual que se encuentre ajustada al modelo planteado para que pueda ser procesado por el sistema, estas mismas limitaciones llevaron a que no fuera posible encontrar distintos datasets durante la ejecución de este trabajo.
- El sistema no fue evaluado considerando métricas de rendimiento, por tanto, con el objetivo de ponerlo en producción es necesario realizar dichas pruebas.
- Este trabajo sólo se enfoca en la utilización del modelo contextual planteado pero no determina los mecanismos para recolectar dicha información de forma implícita. Para este proceso deben realizar esfuerzos utilizando diferentes tecnologías que permitan el descubrimiento de la información.
- Es posible que las ecuaciones planteadas funcionen en distintos dominios de aplicación. Sin embargo, puede darse el caso de que alguna de ellas funcione mejor que las otras en cada dominio, por tanto, se requieren pruebas adicionales acorde al objetivo del sistema para el cual se utilicen.
- Se considera que la arquitectura manejada por la librería y la estructura de clases puede ser aplicada a diferentes dominios, no sólo al de publicidad. Sin embargo, las medidas de similitud de contexto son específicas para el modelo de publicidad diseñado y para el dataset utilizado, por tanto se requiere un trabajo adicional para generar el mecanismo de cálculo de similitud entre contextos propio del entorno de aplicación.
- La librería puede extenderse para soportar diferentes mecanismos de inclusión de contexto. Para este trabajo se enfocó en medir la similitud entre contextos únicamente con una distancia euclidiana, y es posible que otras medidas de similitud como la correlación de Pearson o Spearman puedan utilizarse para realizar esta medida. La librería desarrollada cuenta con todo lo necesario para facilitar la implementación de estas medidas de similitud de contextos sin alterar otros módulos, como el recomendador o el modelo de datos.



# Bibliografía

- [1] K. Partridge and B. Begole, “Activity-based advertising,” in *Pervasive Advertising*, pp. 83–101, Springer, 2011. 3, 11
- [2] R. Lohtia, N. Donthu, and M. D. Guillory, “The impact of advertising, trustworthiness, and valence on the effectiveness of blogs,” *International Journal of Electronic Marketing and Retailing*, vol. 5, no. 4, pp. 317–339, 2013. 3
- [3] R. Uribe, C. Buzeta, and M. Velásquez, “Sidedness, commercial intent and expertise in blog advertising,” *Journal of Business Research*, 2016. 3
- [4] S. Elliott, “The Media Business: Advertising; A Survey of Consumer Attitudes Reveals the Depth of the Challenge that the Agencies Faces.” <http://www.nytimes.com/2004/04/14/business/media-businessadvertising-survey-consumer-attitudes-reveals-depth-challenge.html>, Accedido en 02 2016. 3
- [5] C. Bauer and S. Spiekermann, “Conceptualizing Context for Pervasive Advertising,” in *Pervasive Advertising* (J. Müller, F. Alt, and D. Michelis, eds.), Human-Computer Interaction Series, pp. 159–183, Springer London, jan 2011. 3, 9, 12, 19, 20, 21
- [6] P. Santos, F. Ribeiro, and J. Metrolho, “Using pervasive computing technologies to deliver personal and public ads in public spaces,” pp. 1–6, 2012. 3
- [7] J. Zhou, E. Gilman, M. Ylianttila, and J. Riekk, “Pervasive service computing: Visions and challenges,” in *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on*, pp. 1335–1339, IEEE, 2010. 3
- [8] J. Müller, F. Alt, and D. Michelis, “Pervasive Advertising,” pp. 1–29, Springer London, 2011. 3, 12
- [9] A. Ferscha, W. Swoboda, and C. Wimberger, “En passant Coupon Collection,” 2009. 3
- [10] Q. Gao, P.-L. P. Rau, and G. Salvendy, “Measuring perceived interactivity of mobile advertisements,” *Behaviour & IT*, vol. 29, no. 1, pp. 35–44, 2010. 3
- [11] D. Michelis and H. Send, “Engaging Passers-by with Interactive Screens-A Marketing Perspective.,” in *GI Jahrestagung*, pp. 3875–3881, 2009. 3
- [12] W. Oyomno, P. Jäppinen, and E. Kerttula, “Privacy preserving architecture for context-enhanced personalised pervasive screens,” in *Pervasive2010 workshop on pervasive advertising and shopping*, vol. 8, pp. 569–587, 2010. 3

- [13] F. Alt, M. Balz, S. Kristes, A. S. Shirazi, J. Mennenöh, A. Schmidt, H. Schröder, and M. Goe-dicke, "Adaptive User Profiles in Pervasive Advertising Environments," 2009. 3
- [14] S. Soroa-Koury and K. C. C. Yang, "Factors affecting consumers' responses to mobile adver-tising from a social norm theoretical perspective," *Telematics and Informatics*, vol. 27, no. 1, pp. 103–113, 2010. 3
- [15] S. Spiekermann, "Product context in EC websites: how consumer uncertainty and purchase risk drive navigational needs," in *Proceedings of the 5th ACM conference on Electronic commerce*, pp. 200–207, ACM, 2004. 4
- [16] M. A. Domingues, A. M. Jorge, and C. Soares, "Using contextual information as virtual items on top-n recommender systems," *arXiv preprint arXiv:1111.2948*, 2011. 4, 25
- [17] D. E. Vengroff, "RecLab: a system for eCommerce recommender research with real data, con-text and feedback," in *Proceedings of the 2011 Workshop on Context-awareness in Retrieval and Recommendation*, pp. 31–38, ACM, 2011. 4, 9, 12, 17
- [18] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications," in *Mobile Com-puting Systems and Applications, 1994. WMCSA 1994. First Workshop on*, pp. 85–90, IEEE, 1994. 4, 18
- [19] "Pmbok guide and standards." <http://www.pmi.org/PMBOK-Guide-and-Standards.aspx>, Acce-dido en 05 2016. 5
- [20] P. Kruchten, *The rational unified process: an introduction*. Addison-Wesley Professional, 2004. 5
- [21] W. B. Ashton and R. A. Klavans, *Keeping Abreast of Science and Technology: Technical Inte-lligence in Business*. Battelle Press, 1997. 10
- [22] F. Ricci, L. Rokach, and B. Shapira, "Recommender systems: Introduction and challenges," in *Recommender Systems Handbook*, pp. 1–34, Springer, 2015. 10
- [23] K. F. Yeung and Y. Yang, "A proactive personalized mobile news recommendation system," in *Developments in E-systems Engineering (DESE), 2010*, pp. 207–212, IEEE, 2010. 10
- [24] A. Lommatzsch and S. Albayrak, "Real-time recommendations for user-item streams," in *Pro-ceedings of the 30th Annual ACM Symposium on Applied Computing*, pp. 1039–1046, ACM, 2015. 10
- [25] G. Dror, Y. Koren, Y. Maarek, and I. Szpektor, "I want to answer; who has a question?: Yahoo! answers recommender system," in *Proceedings of the 17th ACM SIGKDD international confe-rence on Knowledge discovery and data mining*, pp. 1109–1117, ACM, 2011. 10
- [26] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," *Internet Computing, IEEE*, vol. 7, no. 1, pp. 76–80, 2003. 11
- [27] A. Tejada-Lorente, C. Porcel, J. Bernabé-Moreno, and E. Herrera-Viedma, "Refore: A re-commender system for researchers based on bibliometrics," *Applied Soft Computing*, vol. 30, pp. 778–791, 2015. 11

## BIBLIOGRAFÍA

---

- [28] C. C. Peddy and D. Armentrout, *Building Solutions with Microsoft® Commerce Server 2002*. °Reilly Media, Inc.”, 2011. 11
- [29] A. Krohn-Grimberghe, A. Nanopoulos, and L. Schmidt-Thieme, “A novel multidimensional framework for evaluating recommender systems,” in *Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERS-TI)*. CEUR-WS, to appear, 2010. 11
- [30] A. Lommatzsch, “Real-Time News Recommendation Using Context-Aware Ensembles,” in *Advances in Information Retrieval*, pp. 51–62, Springer, 2014. 11
- [31] M. Aharon, E. Hillel, A. Kagian, R. Lempel, H. Makabee, and R. Nissim, “Watch-It-Next: A Contextual TV Recommendation System,” in *Machine Learning and Knowledge Discovery in Databases*, pp. 180–195, Springer, 2015. 11
- [32] B. De Carolis, “Adapting news and advertisements to groups,” in *Pervasive Advertising*, pp. 227–246, Springer, 2011. 11
- [33] K. Taniguchi, Y. Harada, and N. T. Duc, “Adslot mining for online display ads,” in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pp. 355–364, IEEE, 2015. 11
- [34] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 6, pp. 734–749, 2005. 11
- [35] M. I. Martín-Vicente, A. Gil-Solla, M. Ramos-Cabrer, Y. Blanco-Fernández, and M. López-Nores, “Semantic inference of user’s reputation and expertise to improve collaborative recommendations,” *Expert Systems with Applications*, vol. 39, pp. 8248–8258, jul 2012. 11
- [36] B. Towle and C. Quinn, “Knowledge based recommender systems using explicit user models,” in *Proceedings of the AAAI Workshop on Knowledge-Based Electronic Markets*, pp. 74–77, 2000. 11
- [37] C. C. Aggarwal, “Knowledge-based recommender systems,” in *Recommender Systems*, pp. 167–197, Springer, 2016. 11
- [38] A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker, “Constraint-based recommender systems,” in *Recommender Systems Handbook*, pp. 161–190, Springer, 2015. 11
- [39] A. G. Parameswaran and H. Garcia-Molina, “Recommendations with prerequisites,” in *Proceedings of the third ACM conference on Recommender systems*, pp. 353–356, ACM, 2009. 11
- [40] A. G. Parameswaran, H. Garcia-Molina, and J. D. Ullman, “Evaluating, combining and generalizing recommendations with prerequisites,” in *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 919–928, ACM, 2010. 11
- [41] D. Bridge, M. H. Göker, L. McGinty, and B. Smyth, “Case-based recommender systems,” *The Knowledge Engineering Review*, vol. 20, no. 03, pp. 315–320, 2005. 11

- [42] A. Gatzoura and M. Sanchez-Marre, “A case-based recommendation approach for market basket data,” *Intelligent Systems, IEEE*, vol. 30, no. 1, pp. 20–27, 2015. 11
- [43] A. S. Foundation, “Apache Mahout: Scalable machine learning and data mining.” [www.mahout.apache.org](http://www.mahout.apache.org), Accedido en 05 2016. 11, 47
- [44] M. D. Ekstrand, M. Ludwig, J. Kolb, and J. T. Riedl, “LensKit: a modular recommender framework,” *RecSys ’11*, (New York, NY, USA), pp. 349–350, ACM, 2011. 11
- [45] I. Wakeman, A. Light, J. Robinson, D. Chalmers, and A. Basu, “Deploying pervasive advertising in a farmers’ market,” in *Pervasive Advertising*, pp. 247–267, Springer, 2011. 11
- [46] N. Taylor and K. Cheverst, “Rural Communities and Pervasive Advertising,” in *Pervasive Advertising*, pp. 269–286, Springer, 2011. 11
- [47] C. Bauer, N. Kryvinska, and C. Strauss, “The Business with Digital Signage for Advertising,” in *Information and Communication Technologies in Organizations and Society*, pp. 285–302, Springer, 2016. 12
- [48] A. Balsamo, “The cultural work of public interactives,” *A Companion to Digital Art*, pp. 330–351, 2016. 12
- [49] A. Di Rienzo, F. Garzotto, P. Cremonesi, C. Frà, and M. Valla, “Towards a smart retail environment,” in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*, pp. 779–782, ACM, 2015. 12
- [50] F. Alt, A. Schmidt, and A. Schmidt, “Advertising on Public Display Networks,” *Computer*, vol. 45, pp. 50–56, may 2012. 12, 20, 21
- [51] H. J. Müller and A. Krüger, “Towards situated public displays as multicast systems,” in *UbiqUM 2006 Workshop on Ubiquitous User Modeling, The 17th European Conference on Artificial Intelligence*, 2006. 12, 21
- [52] J. Müller, A. Schlottmann, and A. Krüger, “Self-optimizing digital signage advertising,” in *Adjunct Proceedings of Ubicomp*, 2007. 12, 21
- [53] A. Di Ferdinando, A. Rosi, R. Lent, A. Manzalini, and F. Zambonelli, “MyAds: A system for adaptive pervasive advertisements,” *Pervasive and Mobile Computing*, vol. 5, no. 5, pp. 385–401, 2009. 12
- [54] M. Sharifi, T. Payne, and E. David, “Public display advertising based on bluetooth device presence,” *Mobile Interaction with the Real World (MIRW 2006)*, p. 52, 2006. 12
- [55] F. R. Ribeiro and J. Metrôlho, “Context-aware information systems for public spaces: The public and private dichotomy. overview, challenges, and experiments,” *International Journal of Multimedia and Ubiquitous Engineering*, vol. 11, no. 3, pp. 11–22, 2016. 12
- [56] C. Bauer and C. Strauss, “Location-based advertising on mobile devices,” *Management Review Quarterly*, pp. 1–36, 2016. 12

## BIBLIOGRAFÍA

---

- [57] L. Carrara and G. Orsi, “A new perspective in pervasive advertising,” tech. rep., Technical report, Department of Computer Science, University of Oxford, 2011. 12
- [58] A. Broder, M. Fontoura, V. Josifovski, and L. Riedel, “A semantic approach to contextual advertising,” in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 559–566, ACM, 2007. 12
- [59] M. Baldauf, S. Dustdar, and F. Rosenberg, “A survey on context-aware systems,” *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 2, no. 4, pp. 263–277, 2007. 12
- [60] J.-y. Hong, E.-h. Suh, and S.-J. Kim, “Context-aware systems: A literature review and classification,” *Expert Systems with Applications*, vol. 36, no. 4, pp. 8509–8522, 2009. 12
- [61] A. K. Dey and G. D. Abowd, “The context toolkit: Aiding the development of context-aware applications,” in *Workshop on Software Engineering for wearable and pervasive computing*, pp. 431–441, 2000. 12
- [62] A. Ferscha, C. Holzmann, and S. Oppl, “Context awareness for group interaction support,” in *Proceedings of the second international workshop on Mobility management & wireless access protocols*, pp. 88–97, ACM, 2004. 12
- [63] L. Liu, F. Lecue, N. Mehandjiev, and L. Xu, “Using context similarity for service recommendation,” in *Semantic Computing (ICSC), 2010 IEEE Fourth International Conference on*, pp. 277–284, IEEE, 2010. 13, 31
- [64] U. Panniello, M. Gorgoglione, and A. Tuzhilin, “Research note—in cars we trust: How context-aware recommendations affect customers’ trust and other business performance measures of recommender systems,” *Information Systems Research*, 2016. 13
- [65] B. Newsom, R. Mittu, M. A. Livingston, S. Russell, J. W. Decker, E. Leadbetter, I. S. Moskowitz, A. Gilliam, C. Sibley, J. Coyne, *et al.*, “Modeling user behaviors to enable context-aware proactive decision support,” in *Context-Enhanced Information Fusion*, pp. 231–267, Springer, 2016. 13
- [66] C.-S. Wang, S.-L. Lin, and H.-L. Yang, “Impersonate human decision making process: an interactive context-aware recommender system,” *Journal of Intelligent Information Systems*, pp. 1–13, 2016. 13
- [67] M. Braunhofer and F. Ricci, “Contextual information elicitation in travel recommender systems,” in *Information and Communication Technologies in Tourism 2016*, pp. 579–592, Springer, 2016. 13
- [68] M. Riegler, M. Larson, C. Spampinato, P. Halvorsen, M. Lux, J. Markussen, K. Pogorelov, C. Griwodz, and H. Stensland, “Right flight?: a dataset for exploring the automatic prediction of movies suitable for a watching situation,” in *Proceedings of the 7th International Conference on Multimedia Systems*, p. 45, ACM, 2016. 13
- [69] Y. Gu, B. Zhao, D. Hardtke, and Y. Sun, “Learning global term weights for content-based recommender systems,” in *Proceedings of the 25th International Conference on World Wide Web*, pp. 391–400, International World Wide Web Conferences Steering Committee, 2016. 13

- [70] H. Drachsler, K. Verbert, O. C. Santos, and N. Manouselis, “Panorama of recommender systems to support learning,” in *Recommender systems handbook*, pp. 421–451, Springer, 2015. 13
- [71] M. Schedl, P. Knees, B. McFee, D. Bogdanov, and M. Kaminskas, “Music recommender systems,” in *Recommender Systems Handbook*, pp. 453–492, Springer, 2015. 13
- [72] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin, “Incorporating contextual information in recommender systems using a multidimensional approach,” *ACM Transactions on Information Systems (TOIS)*, vol. 23, no. 1, pp. 103–145, 2005. 13, 22, 23, 24
- [73] X. Liu and K. Aberer, “SoCo: a social network aided context-aware recommender system,” in *Proceedings of the 22nd international conference on World Wide Web*, pp. 781–802, International World Wide Web Conferences Steering Committee, 2013. 13, 25
- [74] R. G. Tiwari, M. Husain, B. Gupta, and A. Agrawal, “Amalgamating Contextual Information into Recommender System,” in *Emerging Trends in Engineering and Technology (ICETET), 2010 3rd International Conference on*, pp. 15–20, IEEE, 2010. 13, 24
- [75] S. H., “An approach to integration of contextual information in case-based recommender systems.” 13, 25
- [76] M. A. Domingues, A. M. Jorge, and C. Soares, “Exploiting Additional Dimensions as Virtual Items on Top-N Recommender Systems,” in *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*, vol. 1, pp. 92–95, IEEE, 2011. 13
- [77] G. Adomavicius and A. Tuzhilin, “Context-aware recommender systems,” in *Recommender systems handbook*, pp. 217–253, Springer, 2011. 13
- [78] U. Panniello, A. Tuzhilin, M. Gorgoglione, C. Palmisano, and A. Pedone, “Experimental comparison of pre-vs. post-filtering approaches in context-aware recommender systems,” in *Proceedings of the third ACM conference on Recommender systems*, pp. 265–268, ACM, 2009. 14, 25
- [79] Y. Zheng, B. Mobasher, and R. Burke, “Similarity-based context-aware recommendation,” in *Web Information Systems Engineering—WISE 2015*, pp. 431–447, Springer, 2015. 14
- [80] H. Lieberman and T. Selker, “Out of context: Computer systems that adapt to, and learn from, context,” *IBM Systems Journal*, vol. 39, no. 3.4, pp. 617–632, 2000. 18
- [81] A. K. Dey, “Understanding and using context,” *Personal and ubiquitous computing*, vol. 5, no. 1, pp. 4–7, 2001. 18
- [82] K. Maeda, M. Nishi, T. Yoshida, K. Suzuki, and H. Inoue, “Digital signage with audience detection using tv broadcasting waves,” in *Applications and the Internet (SAINT), 2010 10th IEEE/IPSJ International Symposium on*, pp. 225–228, IEEE, 2010. 20
- [83] N. M. Puccinelli, R. C. Goodstein, D. Grewal, R. Price, P. Raghubir, and D. Stewart, “Customer experience management in retailing: understanding the buying process,” *Journal of Retailing*, vol. 85, no. 1, pp. 15–30, 2009. 20

## BIBLIOGRAFÍA

---

- [84] P. Tian, A. V. Sanjay, K. Chiranjeevi, and S. M. Malik, “Intelligent advertising framework for digital signage,” in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1532–1535, ACM, 2012. 21
- [85] J. Müller, J. Exeler, M. Buzeck, and A. Krüger, “Reflectivesigns: Digital signs that adapt to audience attention,” in *Pervasive computing*, pp. 17–24, Springer, 2009. 21
- [86] A. Klaus-Dieter, T. Roth-Berghofer, and K. Bach, “MyCBR Project.” <http://www.mycbr-project.net/index.html>, Accedido en 05 2016. 25
- [87] L. Liu, N. Mehandjiev, and L. Xu, “Using contextual information for service recommendation,” in *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, pp. 1–9, IEEE, 2011. 25
- [88] U. Panniello, A. Tuzhilin, and M. Gorgoglione, “Comparing context-aware recommender systems in terms of accuracy and diversity,” *User Modeling and User-Adapted Interaction*, vol. 24, no. 1-2, pp. 35–65, 2014. 25
- [89] S. Owen, R. Anil, T. Dunning, and E. Friedman, *Mahout in action*. Manning Shelter Island, 2011. 30, 48
- [90] B. Lika, K. Kolomvatsos, and S. Hadjiefthymiades, “Facing the cold start problem in recommender systems,” *Expert Systems with Applications*, vol. 41, no. 4, pp. 2065–2073, 2014. 30
- [91] Epinions, “Epinions.” [www.epinions.com](http://www.epinions.com), Accedido en 05 2016. 32
- [92] Nopcommerce, “NopCommerce.” <http://www.nopcommerce.com/>, Accedido en 05 2016. 32
- [93] F. Martinez Pabon, J. Caicedo Guerrero, J. J. Ibarra Samboni, G. Ramirez Gonzalez, M. Munoz Organero, and A. Chantre Astaiza, “Smart TV a Smartphone Cooperation Model on Digital Signage Environments: An Implementation Approach,” pp. 25–31, July 2014. 32
- [94] X. A. W. Shilin, “A common way implements data access object pattern for j2ee application [j],” *Computer Applications and Software*, vol. 9, p. 051, 2005. 33
- [95] Oracle, “Javasever faces technology.” <http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>, Accedido en 05 2016. 33, 71
- [96] R. S. A. F. mbH, “Easyrec: Add recommendations to your web site.” <http://easyrec.org/>, Accedido en 05 2016. 47
- [97] scikits.recommender Crab developers, “Crab: a recommender framework in python.” <http://muricoca.github.io/crab/install.html>, Accedido en 05 2016. 47
- [98] “Cofi: A java-based collaborative filtering library.” <http://www.nongnu.org/cofi/>, Accedido en 05 2016. 47
- [99] T. Chen, W. Zhang, Q. Lu, K. Chen, Z. Zheng, and Y. Yu, “Svdfeature: a toolkit for feature-based collaborative filtering,” *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 3619–3622, 2012. 47

- [100] “Vogoo - web site personalization and collaborative filtering.” <http://sirius.cs.put.poznan.pl/inf59829/vogoo/docs/MANUAL.html>, Accedido en 05 2016. 47
- [101] T. Instituut/Novay, “Duine framework - recommender software toolkit.” <http://www.duineframework.org/>, Accedido en 05 2016. 47
- [102] L. Contributors, “Lenskit recommender toolkit.” <http://lenskit.org/>, Accedido en 05 2016. 47
- [103] K. Miles, R. y Hamilton, *Learning UML 2.0*. O’Reilly Media, Inc., 2006. 50
- [104] P. Kruchten, “Planos arquitectónicos: El modelo de 4+ 1 vistas de la arquitectura del software,” *IEEE Software*, vol. 12, no. 6, pp. 42–50, 1995. 50, 60
- [105] G. Shani and A. Gunawardana, “Evaluating recommendation systems,” in *Recommender systems handbook*, pp. 257–297, Springer, 2011. 68, 69
- [106] Oracle, “Java SE at a Glance.” <http://www.oracle.com/technetwork/java/javaee/overview/index.html>, Accedido en 05 2016. 71
- [107] PrimeTek, “Primefaces.” <http://www.primefaces.org>, Accedido en 05 2016. 71
- [108] Oracle, “Mysql: The world’s most popular open source database.” <https://www.mysql.com>, Accedido en 05 2016. 71
- [109] IBM, “Spss statistics base.” <http://www-03.ibm.com/software/products/es/spss-stats-base>, Accedido en 05 2016. 78

## Anexo A

# Descripción de casos de uso de la librería Mahout-Context-Recommender

Antes de presentar el diagrama de casos de uso, es importante resaltar que dado a que lo que se espera construir inicialmente es una librería que permita incluir la información de contexto dentro de los sistemas de recomendaciones en el framework Mahout, los usuarios de la librería serán aquellos programadores o desarrolladores de software que pretendan dotar a sus sistemas con esta funcionalidad. La figura A.1 presenta las funcionalidades que un desarrollador esperaría poder realizar con la librería.

Aunque el caso de uso “Obtener recomendaciones usando contexto” se puede considerar como la funcionalidad principal de la librería, existen también otros casos que son valiosos. Por ejemplo, el construir un modelo de datos usando la información contextual o el poder calcular la similitud entre contextos, son funcionalidades que apoyan a la generación de recomendaciones y que también pueden ser utilizadas de forma aislada con otros objetivos. A continuación se describen cada uno de los casos de uso del diagrama.

### 1. CU1: Construir un modelo de datos contextual

**Actores:** Usuario Programador.

**Requisitos:** El usuario cuenta con la información necesaria para construir el modelo de datos. Dicha información puede estar contenida en un archivo de texto plano, base de datos o generarse por código.

**Flujo de eventos:**

- a) El usuario determina el tipo de almacenamiento de los datos y crea una conexión según el caso.
- b) El usuario inicializa el modelo con la conexión creada.
- c) El sistema retorna un modelo de datos con los elementos almacenados.

**Flujos alternos:** Si el elemento que almacena los datos no cuenta con la información requerida

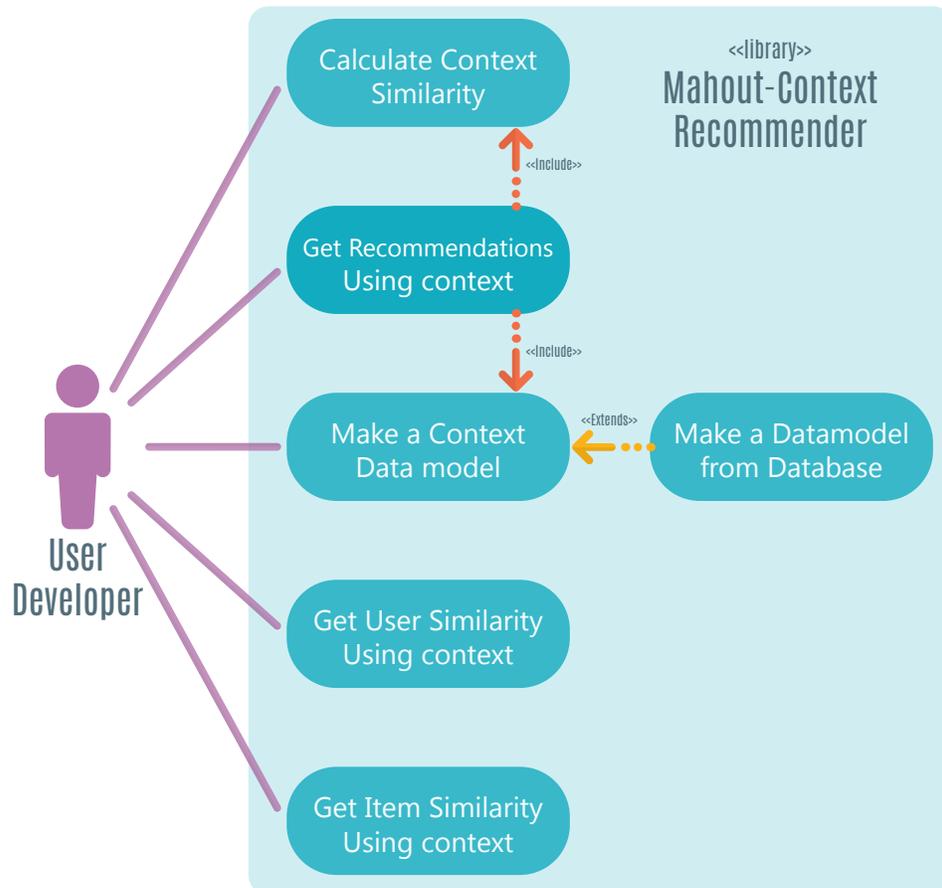


Figura A.1: Diagrama de casos de uso de la librería Fuente: propia

---

se genera una excepción.

**Resultado:** Modelo de datos con los elementos almacenados.

## 2. CU2: Construir un modelo de datos a partir de base de datos

**Actores:** Usuario Programador.

**Requisitos:** El usuario cuenta con una base de datos con la información necesaria para construir el modelo. Dicha base de datos cuenta con una tabla que almacena las preferencias y los identificadores de usuarios, ítems y contextos.

**Flujo de eventos:**

- a) El usuario importa los controladores necesarios para la conexión con la base de datos.
- b) El usuario crea una conexión con la base de datos indicando la tabla que almacena las preferencias y los nombres de columnas para el id de usuario, ítem y contexto.
- c) El sistema retorna un modelo de datos con los elementos almacenados en la base de datos.

**Flujos alternos:** Si el elemento que almacena los datos no cuenta con la información requerida se genera una excepción.

**Resultado:** Modelo de datos con los elementos almacenados en la base de datos.

## 3. CU3: Calcular similitud entre contextos

**Actores:** Usuario Programador.

**Requisitos:**

- El usuario ha configurado correctamente los pesos para cada una de las variables contextuales, así como las distancias.
- El usuario ha construido el modelo de datos y los contextos que se desean comparar se encuentran incluidos en dicho modelo.

**Flujo de eventos:**

- a) El usuario instancia un objeto que permita realizar el cálculo de similitud.
- b) El usuario invoca el cálculo de similitud entregando como parámetros los ids de cada uno de los contextos.
- c) El sistema retorna un valor de similitud entre los dos contextos comparados.

**Flujos alternos:** Si alguno de los contextos o ambos no son encontrados dentro del modelo de datos se genera una excepción indicando el problema.

**Resultado:** Valor de similitud entre dos contextos dados.

#### 4. CU4: Calcular similitud entre usuarios usando contexto

**Actores:** Usuario Programador.

**Requisitos:** El usuario ha definido un mecanismo de cálculo de similitud entre contextos.

**Flujo de eventos:**

- a) El usuario instancia un objeto que permita realizar el cálculo de similitud entre usuarios utilizando contexto.
- b) El usuario invoca el cálculo de similitud entregando como parámetros los ids de cada uno de los usuarios.
- c) El sistema retorna un valor de similitud entre los dos usuarios comparados.

**Flujos alternos:** Si alguno de los usuarios o ambos no son encontrados dentro del modelo de datos se genera una excepción indicando el problema.

**Resultado:** Valor de similitud entre dos usuarios dados.

#### 5. CU5: Calcular similitud entre ítems usando contexto

**Actores:** Usuario Programador.

**Requisitos:** El usuario ha definido un mecanismo de cálculo de similitud entre contextos.

**Flujo de eventos:**

- a) El usuario instancia un objeto que permita realizar el cálculo de similitud entre ítems utilizando contexto.
- b) El usuario invoca el cálculo de similitud entregando como parámetros los ids de cada uno de los ítems.
- c) El sistema retorna un valor de similitud entre los dos ítems comparados.

**Flujos alternos:** Si alguno de los ítems o ambos no son encontrados dentro del modelo de datos se genera una excepción indicando el problema.

---

**Resultado:** Valor de similitud entre dos ítems dados.

## 6. CU6: Generar recomendaciones usando contexto

**Actores:** Usuario Programador.

**Requisitos:**

- a) El usuario ha construido un modelo de datos que incluye información contextual.
- b) El usuario ha definido un mecanismo de cálculo de similitud entre contextos.
- c) El usuario ha definido un mecanismo de cálculo de similitud entre usuarios o ítems.
- d) El usuario ha definido una vecindad en caso de ser necesario.

**Flujo de eventos:**

- a) El usuario instancia un objeto que permita generar las recomendaciones para un usuario dado.
- b) El usuario invoca la generación de recomendaciones entregando como parámetros el id del usuario para el cual desea recomendar, el número de recomendaciones solicitadas y el id del contexto para el cuál se recomendará.
- c) El sistema retorna un listado de ítems recomendados cuya longitud es igual o menor a la cantidad de ítems solicitados por el usuario.

**Flujos alternos:** Si el modelo de datos no es un modelo de datos contextual se genera una excepción indicando el problema.

**Resultado:** Listado de ítems recomendados.



## Anexo B

# Implementación de un Sistema de Recomendaciones que involucra información contextual utilizando la librería “Mahout Context Recommender”

En este anexo se describe el paso a paso para la construcción de un Sistema de recomendaciones que involucra información contextual a partir de la utilización de la librería **Mahout-Context-Recommender**. Para ello se utilizará el IDE de desarrollo Eclipse en su versión Kepler, el kit de desarrollo de JAVA (JDK) versión 7 y la distribución 0.7 de Mahout. En esta sección se presentarán los pasos necesarios para llevar a cabo la tarea principal que puede realizarse con la librería. Es decir, la generación de recomendaciones utilizando contexto. Esta funcionalidad se encuentra expuesta en el diagrama de casos de uso presentado en la figura A.1 y hace uso de las otras funcionalidades de la librería.

Las siguientes secciones describen los pasos para construir el SR con información contextual.

### B.1. Agregar las librerías necesarias al proyecto

Las librerías que se requieren son básicamente la distribución 0.7 de Mahout que se puede agregar al proyecto mediante sus archivos .jar y que pueden ser descargados del siguiente link: <https://archive.apache.org/dist/mahout/0.7/mahout-distribution-0.7.tar.gz>

Adicionalmente se requieren el driver JDBC para MySQL (para este caso se utilizó la versión 5.1.22) y la librería generada en este proyecto (*MahoutContextRecommender.jar*), ambos se encuentran también en paquetes .jar.

## B.2. Instanciar un Modelo de datos que incluye información contextual

Durante el desarrollo de la librería solo se implementaron las clases necesarias para construir un modelo a partir de una base de datos MySQL. Sin embargo, es posible implementar las clases necesarias para construir modelos de datos a partir de archivos de texto plano o bases de datos en motores diferentes a MySQL. Para este caso particular se presentará cómo construir el modelo utilizando las clases ya implementadas. Para ello se debe contar con una base de datos como la de la figura 4.5.

La instanciación del modelo de datos se realiza en los siguientes pasos:

- **Crear un DataSource:**

La fuente de datos o DataSource es una clase del Driver JDBC de MySQL que permite hacer una conexión con la base de datos. Para generarla se requiere inicialmente importar la clase *MysqlDataSource* y luego inicializarla con los datos de conexión de la base de datos. El código de la figura B.1 es un ejemplo para la creación de la fuente de datos.

```
import com.mysql.jdbc.jdbc2.optional.MysqlDataSource;

public class ContextRecommender {

    MysqlDataSource dataSource = new MysqlDataSource();
    dataSource.setServerName("localhost");
    dataSource.setUser("user_name");
    dataSource.setPassword("password");
    dataSource.setDatabaseName("database_name");

}
```

Figura B.1: Creación de la fuente de datos Fuente: propia

- **Crear un “almacén” de conexiones:**

Una buena práctica cuando se trabaja con base de datos es utilizar *almacenes* o en inglés “pools” de conexiones. Con ello, se mantienen algunas conexiones abiertas y de esta manera las consultas se hacen mucho más rápido. Mahout permite implementar este pool de conexiones mediante la clase *ConnectionPoolDataSource*. La figura B.2 contiene el código con la implementación.

- **Crear el modelo de datos:**

### B.3. Crear un objeto para medir la similitud entre contextos

---

```
import com.mysql.jdbc.jdbc2.optional.MysqlDataSource;
import org.apache.mahout.cf.taste.impl.model.jdbc.ConnectionPoolDataSource;

public class ContextRecommender {

    MysqlDataSource dataSource = new MysqlDataSource();
    ...
    ConnectionPoolDataSource poolDataSource
        = new ConnectionPoolDataSource(dataSource);
}
```

Figura B.2: Creación de una piscina de conexiones para la base de datos Fuente: propia

Una vez se tiene el almacén de conexiones, es posible generar el modelo de datos. Para ello simplemente es necesario importar las clases *ContextDataModel* y *MySQLJDBCContextDataModel* e instanciar. La instanciación solicita como parámetros el nombre de la tabla donde se encuentran los registros de preferencias y los nombres de las columnas: usuario, ítem, contexto, preferencia y timestamp. La figura B.3 presenta el código para la creación del modelo de datos.

### B.3. Crear un objeto para medir la similitud entre contextos

Después de creado el modelo de datos es posible generar un objeto encargado de medir la similitud entre contextos. Para ello es necesario inicialmente importar las clases *ContextSimilarity* y *EuclidianContextSimilarity*, siendo esta última una implementación de la primera. La instanciación sólo requiere el modelo de datos como parámetro de entrada. El código de la figura B.4 presenta los resultados.

### B.4. Crear un objeto para medir similitud entre usuarios o ítems incluyendo contexto

Como se presentó en el capítulo 3 la similitud de usuarios o ítems puede verse afectada al incluir información contextual. Para que esto se vea reflejado en el SR es necesario implementar la similitud entre usuarios o ítems a partir de la clase *PearsonCorrelationSimiliarityWithContext*, la cual es una implementación de las interfaces *UserSimilarity* e *ItemSimilarity*. La figura B.5 incluye el código para realizar las instancias. Cabe rescatar que esta clase solicita como parámetro además del modelo de datos, un objeto de tipo *ContextSimilarity*.

## Anexo B. Implementación de un Sistema de Recomendaciones que involucra información contextual utilizando la librería “Mahout Context Recommender”

```
import com.mysql.jdbc.jdbc2.optional.MysqlDataSource;
import org.apache.mahout.cf.taste.impl.model.jdbc.ConnectionPoolDataSource;
import org.unicauca.mahout.cf.taste.impl.recommender.ContextDataModel;
import org.unicauca.mahout.cf.taste.impl.recommender.MySQLJDBCContextDataModel;

public class ContextRecommender {

    MysqlDataSource dataSource = new MysqlDataSource();
    ...
    ConnectionPoolDataSource poolDataSource
        = new ConnectionPoolDataSource(dataSource);
    ContextDataModel dataModel = new MySQLJDBCContextDataModel(
        poolDataSource,
        "taste_preferences",
        "user_id",
        "item_id",
        "context_id",
        "preference",
        "timestamp",
    );
}
```

Figura B.3: Creación del modelo de datos con información contextual Fuente: propia

```
import ...
import org.unicauca.mahout.cf.taste.impl.recommender.ContextSimilarity;
import org.unicauca.mahout.cf.taste.impl.recommender.EuclidianContextSimilarity;

public class ContextRecommender {

    MysqlDataSource dataSource = new MysqlDataSource();
    ...
    ConnectionPoolDataSource poolDataSource
        = new ConnectionPoolDataSource(dataSource);
    ContextDataModel dataModel = new MySQLJDBCContextDataModel(...);
    ContextSimilarity contextSimilarity = new EuclidianContextSimilarity(dataModel);
}
```

Figura B.4: Instanciación de un objeto para medir similitud entre contextos Fuente: propia

## B.5. Crear el recomendador utilizando información de contexto

---

```
import ...
import org.unicauca.mahout.cf.taste.impl.recommender
    .PearsonCorrelationSimilarityWithContext;

public class ContextRecommender {
    ...
    //Cuando se trata de un SR basado en filtrado user-user
    UserSimilarity userSimilarity
        = new PearsonCorrelationSimilarityWithContext(dataModel, contextSimilarity);

    //Cuando se trata de un SR basado en filtrado item-item
    ItemSimilarity itemSimilarity
        = new PearsonCorrelationSimilarityWithContext(dataModel, contextSimilarity);
}
```

Figura B.5: Instanciación de un objeto para medir similitud entre usuarios e items incluyendo contexto  
Fuente: propia

## B.5. Crear el recomendador utilizando información de contexto

Una vez se han creado todo los objetos necesarios, el último paso para construir un sistema de recomendaciones utilizando contexto a partir de la librería “Mahout Context Recommender” es crear el objeto encargado de hacer las recomendaciones. Para ello, es necesario instanciar un objeto del tipo *UserBasedContextRecommender* cuando se desea trabajar con un filtrado user-user o del tipo *ItemBasedContextRecommender* cuando se desea un filtrado item-item. Es importante mencionar que en el caso de user-user se requiere también crear una vecindad y para ello sólo basta con utilizar un objeto del tipo *UserNeighborhood* de Mahout. La figura B.6 presenta el código final para la implementación de un SR user-user incluyendo contexto (La implementación para el SR item-item es similar).

Una vez se cuenta con la instancia del recomendador, es posible solicitar recomendaciones al sistema. Para ello basta con ejecutar el método “*recommend*”, el cual recibe como parámetros el id del usuario, el número de recomendaciones esperadas y el id del contexto en el que se dará la recomendación. También es importante rescatar que dado a que la arquitectura de la librería fue diseñada para que pudiera convivir correctamente con la arquitectura de Mahout, es posible utilizar las herramientas de evaluación de sistemas de recomendaciones que Mahout trae por defecto también para sistemas que incluyen contexto.

Al finalizar este paso se cuenta con la implementación de un sistema de recomendaciones que involucra información contextual y que ha sido diseñado para el campo específico de la publicidad ubicua, pues los cálculos de pesos y el formato de los elementos de contexto han sido definidos a partir del Dataset para publicidad ubicua generado en el capítulo 3. Como se mencionó al principio de este capítulo, esta librería podría ser extendida a diferentes ambientes de recomendaciones pero para ello deben generarse implementaciones del medidor de similitud entre contextos propias para cada entorno.

## Anexo B. Implementación de un Sistema de Recomendaciones que involucra información contextual utilizando la librería “Mahout Context Recommender”

```
import com.mysql.jdbc.jdbc2.optional.MysqlDataSource;
import org.apache.mahout.cf.taste.impl.model.jdbc.ConnectionPoolDataSource;
import org.unicauca.mahout.cf.taste.impl.recommender.ContextDataModel;
import org.unicauca.mahout.cf.taste.impl.recommender.MySQLJDBCContextDataModel;
import org.apache.mahout.cf.taste.similarity.UserSimilarity;
import org.unicauca.mahout.cf.taste.impl.recommender
    .PearsonCorrelationSimilarityWithContext;
import org.apache.mahout.cf.taste.neighborhood.UserNeighborhood;
import org.unicauca.mahout.cf.taste.impl.recommender.ContextRecommender;
import org.unicauca.mahout.cf.taste.impl.recommender.UserBasedContextRecommender;

public class ContextRecommender {

    MysqlDataSource dataSource = new MysqlDataSource();
    dataSource.setServerName("localhost");
    dataSource.setUser("user_name");
    dataSource.setPassword("password");
    dataSource.setDatabaseName("database_name");

    ConnectionPoolDataSource poolDataSource
        = new ConnectionPoolDataSource(dataSource);

    ContextDataModel dataModel = new MySQLJDBCContextDataModel(
        poolDataSource,
        "taste_preferences",
        "user_id",
        "item_id",
        "context_id",
        "preference",
        "timestamp",
    );

    ContextSimilarity contextSimilarity = new EuclidianContextSimilarity(dataModel);

    UserSimilarity userSimilarity
        = new PearsonCorrelationSimilarityWithContext(dataModel, contextSimilarity);
    UserNeighborhood neighborhood
        = new NearestUserNeighborhood(50, userSimilarity, dataModel);

    ContextRecommender recommender
        = new UserBasedContextRecommender(dataModel,
            neighborhood,
            userSimilarity,
            contextSimilarity);
}
```

Figura B.6: Código para la implementación de un SR user-user incluyendo información contexto Fuente: propia

## **Anexo C**

# **Ficha técnica de publicaciones**



# Ficha Técnica de Publicaciones

Ing. Víctor Garzón Marín  
Universidad del Cauca  
Maestría en Ingeniería Telemática  
2016

*Descripción de las publicaciones obtenidas durante la realización de esta investigación*

## Artículo de Revista

- Título **Sistema de Recomendaciones para entornos de Digital Signage soportado en un esquema de colaboración SmartTV - Smartphone**
- Autores Francisco Martínez-Pabón, Víctor Garzón-Marín, Juan Camilo Ospina Quintero, Jhon Jairo Ibarra-Samboni, Jaime Caicedo- Guerrero, Ángela Chantre-Astaiza, Gustavo Ramírez-González
- Revista Revista de Ingenierías Universidad de Medellín
- Clasificación Colciencias A2
- ISSN 1692-3324/ISSNe:2248-4094
- Año 2014

## Abstract

Advertisement in Digital Signage environments demands the enrichment of the classic approach of recommendation focused on individuals by means of the delivery of ads for a group of people observing a public screen. Even though the accuracy that can be achieved in the recommendations is important, the degree of novelty perceived by the users is even more important. This balance can be obtained not only through the improvement of the recommendation algorithms, but also through an adequate multiscreen cooperation scheme. Therefore, this paper proposes an approach for the construction of a recommendations system for Digital Signage environments based on a Smart TV- Smartphone cooperation scheme.

## Artículo de Conferencia

- Título **Enriching public displays ads recommendations using an individual - group cooperation model**
- Autores Francisco Martínez-Pabón, Víctor Garzón-Marín, Juan Camilo Ospina Quintero, Ángela Chantre-Astaiza, Gustavo Ramírez-González, Mario Munoz-Organero
- Conferencia 3rd International Symposium on pervasive displays - **Copenhaguen Denmark**
- Editor ACM New York
- DOI 10.1145/2611009.2617196
- Año 2014

## Abstract

Ads recommendations delivery on digital signage environments must consider group and individual profiles. Although the ads recommendation precision is relevant and frequently explored, a good serendipity/precision balance must be a concern on Digital Signage context. This paper introduces a public display recommender system approach based on an individual-group cooperation model implemented throughout a Smart TV -- Smartphone interaction scheme.

## Abstract

The advertising has an important role in the relationship between the goods and services providers and their potential clients. However, the classic approach follows a broadcast philosophy where the same content is delivered to the whole people, so the content it is not enough personalized and it reduces the effectiveness of the advertising campaign. Recently, a new paradigm known as pervasive advertising offers a new dimension of customization and ubiquity features by applying pervasive computing technologies in the advertising domain. Specifically, the use of Recommender systems, which applies filtering techniques for the delivery of personalized content to the target users, may add robust personalization capabilities but they may not be enough if the ads are delivery in the inappropriate context. This research proposes an experimental approach to deliver ads recommendations from information about the user context. The approach is supported on the definition of a reference model about the context variables to consider in the advertising domain, the definition of algorithms based on dimensionality reduction techniques to process the context during the recommendations calculation and a reference implementation of these algorithms using the Apache Mahout framework.

## Artículo de Revista

Título	<b>Context aware recommender system for pervasive advertising: experimental approach</b>
Autores	Víctor Garzón-Marín, Francisco Martínez-Pabón, Gustavo Ramírez-González, Mario Muñoz-Organero
Revista	Information Science
Homologada Colciencias	A1 - Indexada JCR
ISSN	0020 - 0255
Año	2016

*En revisión*

## Anexo D

# Información del Dataset para publicidad ubicua con información contextual generado experimentalmente

En este anexo se describe el dataset para publicidad ubicua que contiene información contextual generado en una de las fases de experimentación de este trabajo.

### D.1. Información contenida en el dataset

La información contenida dentro del dataset puede apreciarse en la figura D.1

1. **Usuarios (users):** Contiene la información de las personas que utilizan el sistema. Sus atributos se listan a continuación:
  - **id (int):** Identificador único del usuario.
  - **name (string):** Nombre del usuario.
  - **sex (string [m,f]):** Género del usuario, puede ser masculino (m) o femenino (f).
  - **age (int):** Edad del usuario. Para este experimento mayores de 18 años.
  - **email (string):** Correo electrónico del usuario, utilizado para mantener contacto con el mismo e invitarle a futuros experimentos, también para enviar la contraseña en caso de olvidarla.
  - **login (string):** Nombre de usuario utilizado para iniciar sesión en el aplicativo.
  - **password (string):** Contraseña utilizada para iniciar sesión en el aplicativo.
2. **Anuncios (ads):** Contiene la información de los anuncios publicitarios que le son recomendados a los usuarios. Sus atributos se listan a continuación:
  - **id (int):** Identificador único del anuncio.
  - **url (string):** Dirección de la imagen que acompaña el anuncio.

## Anexo D. Información del Dataset para publicidad ubicua con información contextual generado experimentalmente

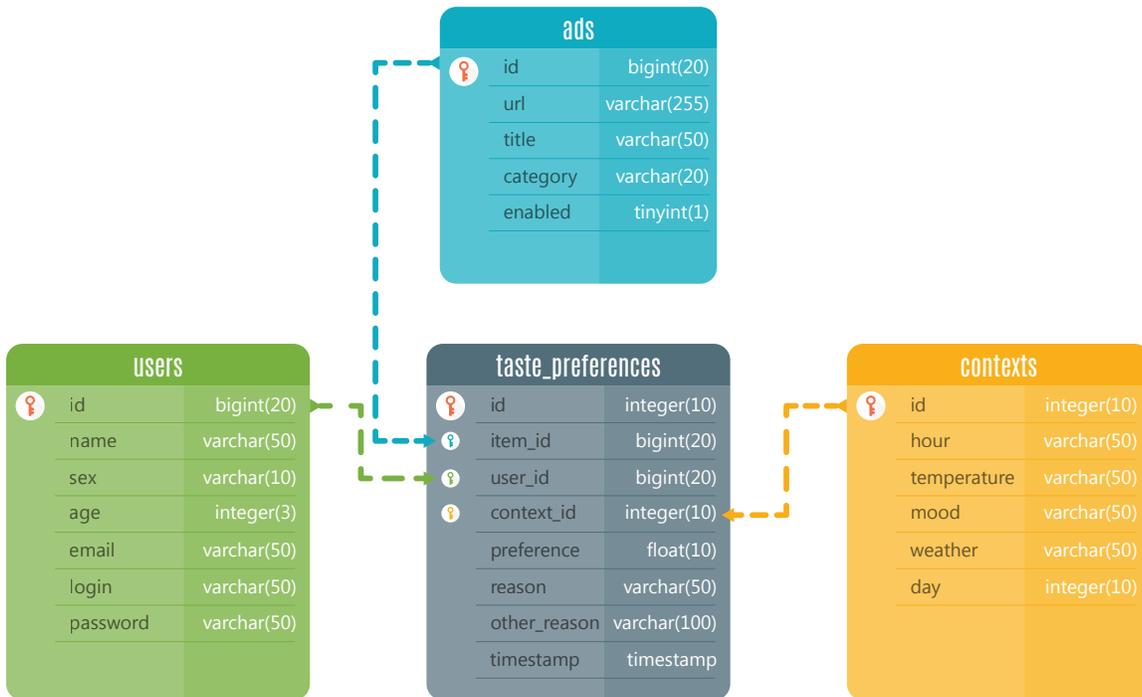


Figura D.1: Información del dataset contextual para publicidad ubicua Fuente: creación propia

- **title (string):** Título del anuncio.
- **description (string):** Descripción detallada.
- **category (string):** Categoría en la cual se encuentra clasificado. Las opciones son: ropa, deportes, restaurantes, entretenimiento, hogar, tecnología y vehículos.
- **enabled (bool):** Para efectos del experimento algunos anuncios podían no ser muy útiles, por este motivo sólo se tenían en cuenta aquellos que se encontraran habilitados.

3. **Contextos (contexts):** Contiene la información del contexto en el momento en que fue dada una calificación. Sus atributos se listan a continuación:

- **id (int):** Identificador único del contexto (algunos contextos se reutilizan en distintas calificaciones).
- **hour (string [day, night]):** Momento del día en el que fue expresada la calificación, puede ser día o noche.
- **temperature (string [cold, warm]):** Temperatura percibida por el usuario en el momento de dar la calificación, puede ser frío o calor.
- **mood (string [happy, normal, sad, boring]):** Estado de ánimo del usuario en el momento de calificar, puede ser feliz, normal, triste o aburrido.
- **weather (string [cloudless, cloud, rain]):** Clima en el momento de calificar, puede ser despejado, nublado o lluvia.

## D.2. Dataset resultante

---

- **day (int):** Día de la semana en el cual se expresa la calificación, puede ser cualquier día de lunes (0) a domingo (6).
4. **Calificaciones (taste-preferences):** Contiene el listado de calificaciones que conectan los usuarios con los items y en contextos específicos. Sus atributos son:
- **id (int):** Identificador único de la calificación.
  - **item-id (int):** identificador del anuncio.
  - **user-id (int):** identificador del usuario.
  - **context-id (int):** identificador del contexto.
  - **preference (float):** Valor de la calificación, para este experimento se utilizó una escala de 1 (no le gusta para nada) a 5 (le gusta mucho).
  - **reason (string [category, day, hour, mood, weather, sex, age]):** Razón que llevó al usuario a expresar la calificación. Puede ser por la categoría del anuncio, por la temperatura, por el día, por el momento, etc.
  - **other-reason (string):** Cuando el usuario expresa que la razón por la que dio la calificación no se encuentra enmarcada en ninguna de las variables planteadas, se le solicita que suministre una descripción de su razón. Con esto se espera identificar si alguna variable importante no fue tomada en cuenta en el modelo general.
  - **timestamp (timestamp):** Fecha y hora en la cual se expresó la calificación.

## D.2. Dataset resultante

Una vez llevado a cabo el experimento se consiguió un dataset con las siguientes características:

- **Usuarios:** 98
- **Anuncios totales:** 170
- **Anuncios activos:** 130
- **Contextos:** 57
- **Calificaciones:** 1794

