

# **Filtrado Colaborativo de Aplicaciones Móviles Basado en la información Social y Contextual del Usuario**



Universidad  
del Cauca

Monografía presentada para optar al título de Ingeniero en  
Electrónica y Telecomunicaciones

**Darío Fernando Chamorro Vela  
Pablo Esteban Calvache López**

Director: PhD. Ing. Juan Carlos Corrales

Asesor: Ing. Luis Antonio Rojas Potosí

*Universidad del Cauca*

**Facultad de Ingeniería Electrónica y Telecomunicaciones  
Departamento de Telemática  
Línea de Investigación Aplicaciones y Servicios sobre Internet  
Popayán, Junio de 2013**

## **Filtrado Colaborativo de Aplicaciones móviles Basado en la Información Social y Contextual del usuario**

Chamorro Vela, Darío Fernando

Calvache López, Pablo Esteban

Director: PhD Ing. Juan Carlos Corrales Muñoz

Asesor: Ing. Luis Antonio Rojas Potosí

Fecha de sustentación: 23 de Agosto del 2013

Lugar: Salón 228, Facultad de Ingeniería Electrónica y Telecomunicaciones

Concepto: Aprobado

Jurados:

Ing. Javier Alexander Hurtado

Ing. Gustavo Ramírez

Facultad de Ingeniería Electrónica y Telecomunicaciones  
Programa de Ingeniería Electrónica y Telecomunicaciones  
Departamento de Telemática  
Grupo de Ingeniería Telemática  
Línea de investigación Aplicaciones y Servicios sobre Internet  
Universidad del Cauca  
Popayán – Cauca, 2013

# TABLA DE CONTENIDO

	<b>Pag.</b>
<b>Capítulo I, Introducción.....</b>	<b>1</b>
1.1. Contexto de investigación.....	2
1.2. Motivación.....	3
1.3. Premisas e Hipótesis.....	4
1.3.1. Premisas.....	5
1.3.2. Hipótesis.....	5
1.4. Objetivos.....	6
1.4.1. Objetivo general.....	6
1.4.2. Objetivos específicos.....	6
1.5. Contribuciones y Principales Resultados.....	6
1.6. Contenido de la Monografía.....	8
<b>Capítulo II, Estado Del Arte .....</b>	<b>10</b>
2.1. Contexto General.....	10
2.1.1. Sistemas de recomendación.....	10
2.1.2. Comparación de técnicas de recomendación.....	12
2.1.3. Filtrado colaborativo.....	15
2.1.4. Contexto.....	16
2.2. Trabajos relacionados.....	19
2.2.1. Consumo de aplicaciones móviles.....	19
2.2.2. Filtrado Colaborativo Basado en Contexto .....	21
2.2.3. Consideración del contexto social en SR.....	21
2.3. Brechas existentes.....	23
<b>Capítulo III, Modelo de información contextual y Calificación por análisis de Consumo....</b>	<b>25</b>
3.1. Fuentes de Información.....	25
3.2. Tipos de información.....	26
3.2.1. Información Contextual .....	26
3.2.2. Información de Dispositivo .....	30
3.2.3. Información de Aplicaciones Móviles .....	31
3.3. Modelo Contextual.....	32

3.3.1.	Dimensión contextual.....	32
3.3.2.	Estado Contextual .....	32
3.4.	Calificación por análisis de Consumo .....	33
3.4.1.	Análisis del Consumo.....	34
3.4.2.	Tipos de Aplicaciones Móviles.....	34
3.4.3.	Calificación General por Consumo (Score).....	35
3.4.4.	Calificación Contextual por Consumo.....	36
3.4.5.	Detección de categoría por contexto.....	38
3.5.	Privacidad y Seguridad.....	40
3.6.	Resumen .....	40
<b>Capítulo IV, Vecindad, Predicción y Recomendación .....</b>		<b>41</b>
4.1.	Generalidades sobre predicción y recomendación en FC .....	42
4.1.1.	Vecindad .....	42
4.1.2.	Predicción.....	49
4.1.3.	Recomendación.....	49
4.1.4.	Ejemplo de implementación (Ekstrand, y otros, 2011) .....	49
4.2.	Propuesta .....	51
4.2.1.	Predicción.....	51
4.2.2.	Recomendación final.....	55
4.3.	Resumen .....	57
<b>Capítulo V, Vanilla: Arquitectura e implementación .....</b>		<b>59</b>
5.1.	Casos de Uso del Sistema .....	59
5.2.	Arquitectura del Sistema .....	66
5.2.1.	Aplicación Móvil .....	66
5.2.2.	Sincronización .....	67
5.2.3.	Servidor .....	67
5.3.	Consideraciones de la Implementación del prototipo.....	68
5.3.1.	Tecnologías Empleadas .....	68
5.3.2.	Publicación.....	70
5.4.	Resumen .....	71
<b>Capítulo VI, Experimentación y Evaluación .....</b>		<b>72</b>
6.1.	Metodología.....	72

6.1.1.	Planeación del experimento.....	72
6.1.2.	Métricas de evaluación.....	74
6.1.3.	Medida de <i>interés</i> , respecto al tiempo de observación de las interfaces de recomendación (on-line).....	76
6.1.4.	MAE (Off-line).....	77
6.2.	Plan de ejecución de pruebas y resultados.....	78
6.2.1.	Plan de pruebas - eficacia.....	79
6.2.2.	Plan de pruebas - interés.....	81
6.2.3.	Plan de pruebas - MAE.....	82
6.2.4.	Plan de pruebas – Disponibilidad .....	83
6.2.5.	Plan de pruebas – Datos estadísticos complementarios .....	84
6.3.	Resumen .....	87
<b>Capítulo VII, Conclusiones y trabajos Futuros .....</b>		<b>88</b>
7.1.	Contribuciones .....	88
7.2.	Conclusiones .....	89
7.3.	Trabajos futuros .....	91

# INDICE DE FIGURAS

	<b>Pag.</b>
FIGURA 1. PROCESO DE RECOMENDACIÓN DE SISTEMAS DE FILTRADO COLABORATIVO. ....	15
FIGURA 2. REPRESENTACIÓN GRÁFICA DE UN GRAFO. ....	18
FIGURA 3. FUENTES DE INFORMACIÓN. ....	26
FIGURA 4. CATEGORÍAS DE INFORMACIÓN CONTEXTUAL. ....	26
FIGURA 5. CICLO DE VIDA DE UNA APLICACIÓN MÓVIL.....	31
FIGURA 6. MODELO CONTEXTUAL DEL USUARIO. ....	33
FIGURA 7. ENCUESTA SOBRE EL USO DE BLUETOOTH, DIRIGIDA A LA COMUNIDAD DE LA UNIVERSIDAD DE CAUCA (PUBLICADA A TRAVÉS DE LAS CUENTAS EN "FACEBOOK" Y "TWITTER" DE ESTA INSTITUCIÓN), Y AL GRUPO DE PERSONAS COMPUESTO POR LOS CONTACTOS EN "FACEBOOK" Y SEGUIDORES EN "TWITTER", DE LOS TESISISTAS Y ASESOR. ....	45
FIGURA 8. RED SOCIAL DE LLAMADAS (EN EL CASO DE QUE EXISTA CIERTO NÚMERO MENSAJES ENTRE PABLO Y DARIO, ESTE SERÁ SUMADO AL NÚMERO DE LLAMADAS PARA DEFINIR EL PESO DE LA RELACIÓN).....	46
FIGURA 9. USUARIOS EXPERTOS DE LA RED SOCIAL DE CONTACTOS DEL USUARIO. ....	48
FIGURA 10. MATRIZ $N$ , APLICACIÓN VS. USUARIO.....	50
FIGURA 11. REGISTRO DE USUARIO .....	60
FIGURA 12. (A) WIDGET DE CLIENTE, (B) INGRESO DEL ESTADO DE ÁNIMO (C) INGRESO DE LA LOCALIZACIÓN RELATIVA.....	62
FIGURA 13. CONFIGURACIÓN DE LA PRIORIDAD DE LAS DIMENSIONES CONTEXTUALES. ....	63
FIGURA 14. CONFIGURACIÓN DEL MUESTREO DE LAS DIMENSIONES CONTEXTUALES.....	64
FIGURA 15. RECOMENDACIÓN DE APLICACIONES MÓVILES. ....	66
FIGURA 16. ARQUITECTURA DEL SISTEMA DE RECOMENDACIÓN PROPUESTO, VANILLA. ....	66
FIGURA 17. EFICACIA PROMEDIO GENERAL.....	80
FIGURA 18. EVOLUCIÓN DE LA EFICACIA EN FUNCIÓN DE TAO. ....	80
FIGURA 19. TIEMPO PROMEDIO DE OBSERVACIÓN POR PÁGINA. ....	81
FIGURA 20. ESTADÍSTICAS GENERALES DE LA BASE DE DATOS CENTRAL. ....	83
FIGURA 21. TIEMPO PROMEDIO DE USO DE APLICACIONES (EN SEGUNDOS) POR DÍA (INCLUYENDO USO DE APLICACIONES MÓVILES NATIVAS COMO GESTIÓN DE CONTACTOS, AGENDA, AJUSTES, ETC).....	84
FIGURA 22. TIEMPO PROMEDIO DE USO DE APLICACIONES (EN SEGUNDOS), DIARIAMENTE (SIN INCLUIR APLICACIONES NATIVAS).....	85
FIGURA 23. FRECUENCIA PROMEDIO DE USO DE APLICACIONES, DIARIAMENTE (INCLUYENDO USO DE APLICACIONES NATIVAS COMO GESTIÓN DE CONTACTOS, AGENDA, AJUSTES, ETC.).....	85
FIGURA 24. FRECUENCIA PROMEDIO DE USO DE APLICACIONES, DIARIAMENTE (SIN INCLUIR APLICACIONES NATIVAS). ....	86
FIGURA 25. PORCENTAJE PROMEDIO DE CATEGORÍAS CONSUMIDAS.....	86

# INDICE DE TABLAS

Pag.

TABLA 1. USO RELATIVO DE APLICACIONES POR CATEGORÍA Y HORA EN TÉRMINOS DE FRECUENCIA DE USO. EL VALOR DE CADA CELDA SE REFIERE AL PORCENTAJE DEL LANZAMIENTO DE APLICACIONES REALIZADO POR LOS USUARIOS EN CADA HORA POR CATEGORÍA. LOS COLORES SON NORMALIZADOS POR FILA (BÖHMER, Y OTROS, 2011).....	1
TABLA 2. DESCRIPCIÓN DE LOS COMPONENTES DE CADA TÉCNICA DE RECOMENDACIÓN.....	13
TABLA 3. COMPARATIVA TÉCNICAS DE RECOMENDACIÓN.....	13
TABLA 4. SENSORES DE ENTORNO EN DISPOSITIVOS ANDROID.....	27
TABLA 5. SENSORES DE MOVIMIENTO EN DISPOSITIVOS ANDROID.....	27
TABLA 6. SENSORES DE POSICIÓN EN EL DISPOSITIVO ANDROID.....	28
TABLA 7. DESCRIPCIÓN DEL SERVICIO RSS DE YAHOO WEATHER.....	28
TABLA 8. INFORMACIÓN PROCESADA.....	29
TABLA 9. PRINCIPALES TIPOS DE ESTADO DE ÁNIMO (CHENG-YU, 2004).....	29
TABLA 10. REDES SOCIALES EMPLEADAS.....	30
TABLA 11. INFORMACIÓN DEL DISPOSITIVO MÓVIL.....	30
TABLA 12. INFORMACIÓN DE APLICACIONES MÓVILES OBTENIDA EN GOOGLE PLAY.....	32
TABLA 13. ELEMENTOS Y COMPONENTES DE CADA TÉCNICA DE RECOMENDACIÓN.....	57
TABLA 14. CASO DE USO <i>REGISTRAR USUARIO</i> .....	60
TABLA 15. CASO DE USO <i>INGRESAR INFORMACIÓN EXPLÍCITA</i> .....	61
TABLA 16. CASO DE USO <i>INGRESAR INFORMACIÓN EXPLÍCITA</i> .....	62
TABLA 17. CASO DE USO <i>CONFIGURAR MUESTREO</i> .....	63
TABLA 18. CASO DE USO <i>SINCRONIZAR INFORMACIÓN</i> .....	64
TABLA 19. CASO DE USO <i>SOLICITAR RECOMENDACIÓN</i> .....	65
TABLA 20. INFORMACIÓN/MENSAJES DE SINCRONIZACIÓN.....	67
TABLA 21. DIMENSIONES CONTEXTUALES CAPTURADAS.....	69
TABLA 22. PLAN DE PRUEBAS - EFICACIA.....	79
TABLA 23. PLAN DE PRUEBAS – INTERÉS.....	81
TABLA 24. PLAN DE PRUEBAS – INTERÉS.....	82
TABLA 25. MAE ALGORITMOS DE FC TRADICIONAL Y BASADO EN SIMILITUD SOCIAL.....	82
TABLA 26. ESPECIFICACIONES DEL SERVIDOR UTILIZADO.....	83
TABLA 27. PLAN DE PRUEBAS – DISPONIBILIDAD.....	83
TABLA 28. PLAN DE PRUEBAS – DATOS ESTADÍSTICOS COMPLEMENTARIOS.....	84
TABLA 29. VALIDACIÓN HIPÓTESIS.....	87





# Capítulo I, Introducción

En los últimos años, el avance tecnológico en hardware y software, ha permitido un aumento considerable en las capacidades de los dispositivos móviles; cuya potencia actual puede ser equiparable a las supercomputadoras de hace algunos años, y el ancho de banda de sus canales de comunicación comparable con aquél disponible para conectar ciudades enteras tiempo atrás (Aharony, 2012). Gracias a esta evolución, hoy en día las personas tienen a su disposición equipos portables y avanzados, que soportan sus actividades diarias mediante aplicaciones “software” que se ejecutan en ellos. Estas aplicaciones son conocidas en la literatura como Aplicaciones Móviles<sup>1</sup> y su importancia radica en su portabilidad y variedad, soportada por el carácter genérico, robustez y facilidades de desarrollo de los sistemas operativos móviles actuales.

Debido principalmente a su portabilidad, estas App pueden ser utilizadas en diversas ocasiones, dependiendo de las características de la situación actual del usuario. Este hecho fue comprobado por los trabajos de Böhmer (Böhmer, y otros, 2011), Falaki (Falaki, y otros, 2010) y Yan (Yan, y otros, 2012), los cuales presentan resultados que permiten determinar que el consumo de aplicaciones móviles se encuentra estrechamente relacionado con las características contextuales del usuario. Como ejemplo, el trabajo de Böhmer expone la siguiente tabla que relaciona la dimensión contextual del tiempo con el consumo de categorías específicas de aplicaciones; donde se puede observar claramente que en horas de la mañana es más propenso el uso de aplicaciones de productividad, mientras que en la noche es habitual consumir aplicaciones que brindan contenido multimedia, relacionado con deportes y juegos.

**Tabla 1.** Uso relativo de aplicaciones por categoría y hora en términos de frecuencia de uso. El valor de cada celda se refiere al porcentaje del lanzamiento de aplicaciones realizado por los usuarios en cada hora por categoría. Los colores son normalizados por fila (Böhmer, y otros, 2011).

	12am	1am	2am	3am	4am	5am	6am	7am	8am	9am	10am	11am	12pm	1pm	2pm	3pm	4pm	5pm	6pm	7pm	8pm	9pm	10pm	11pm	% of Total Launches	Users	Apps
Browser	7.9%	7.7%	7.8%	7.6%	7.3%	7.4%	7.0%	7.9%	8.1%	8.0%	7.7%	7.3%	7.0%	6.9%	6.8%	6.4%	6.6%	6.6%	6.4%	6.6%	7.0%	7.4%	7.5%	7.4%	6.83%	2,398	9
Comics	4.5%	5.2%	5.4%	5.8%	5.8%	5.6%	5.5%	5.2%	5.4%	5.1%	4.7%	4.3%	4.2%	4.2%	4.3%	4.4%	4.0%	4.4%	4.2%	4.1%	4.1%	4.1%	4.1%	4.4%	4.31%	2,151	1,810
Communication	44.9%	41.1%	38.3%	35.4%	31.6%	31.8%	32.7%	34.7%	39.4%	44.8%	49.0%	52.6%	54.8%	55.2%	55.2%	56.1%	55.7%	56.8%	57.1%	56.1%	54.8%	53.3%	52.0%	49.0%	49.50%	2,769	550
Entertainment	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.02%	126	43
Finance	0.2%	0.3%	0.3%	0.2%	0.1%	0.1%	0.1%	0.2%	0.3%	0.3%	0.4%	0.5%	0.3%	0.3%	0.4%	0.3%	0.3%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.25%	604	164
Games	3.2%	3.0%	3.0%	2.7%	2.5%	2.3%	2.2%	1.7%	1.9%	1.9%	2.0%	2.1%	2.2%	2.2%	2.2%	2.3%	2.3%	2.2%	2.2%	2.4%	2.7%	3.0%	3.0%	3.2%	2.30%	1,716	1,702
Health	0.3%	0.4%	0.4%	0.4%	0.6%	0.6%	0.7%	0.6%	0.4%	0.3%	0.3%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.3%	0.2%	0.3%	0.2%	0.3%	0.3%	0.26%	540	227
Libraries & Demo	0.4%	0.5%	0.6%	0.7%	0.9%	0.8%	0.7%	0.6%	0.5%	0.4%	0.3%	0.3%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.3%	0.3%	0.3%	0.3%	0.30%	1,267	117
Lifestyle	0.8%	0.9%	1.0%	1.4%	1.3%	1.5%	1.4%	1.4%	1.1%	0.9%	0.6%	0.5%	0.5%	0.5%	0.5%	0.6%	0.5%	0.3%	0.4%	0.4%	0.5%	0.5%	0.5%	0.5%	0.60%	2,132	451
Multimedia	2.1%	2.1%	2.4%	2.4%	2.7%	2.4%	1.8%	1.8%	1.9%	1.7%	1.8%	2.0%	2.0%	2.0%	2.2%	2.1%	2.2%	2.4%	2.3%	2.3%	2.2%	2.1%	1.9%	2.0%	2.03%	1,713	76
News	2.6%	2.5%	2.6%	2.5%	2.5%	2.7%	3.3%	3.7%	4.1%	3.6%	3.0%	2.6%	2.5%	2.7%	2.5%	2.4%	2.2%	2.1%	2.3%	2.3%	2.3%	2.2%	2.3%	2.3%	2.46%	1,777	440
Productivity	3.6%	5.0%	5.0%	5.8%	6.3%	6.5%	6.0%	5.4%	4.8%	5.1%	4.9%	4.3%	4.2%	4.0%	4.0%	3.7%	3.4%	3.4%	3.0%	3.1%	3.1%	3.0%	2.9%	3.2%	3.76%	2,190	648
Reference	0.7%	0.7%	0.7%	0.7%	0.7%	0.6%	0.6%	0.7%	0.5%	0.5%	0.5%	0.4%	0.4%	0.4%	0.4%	0.3%	0.4%	0.4%	0.4%	0.4%	0.5%	0.5%	0.5%	0.6%	0.47%	903	346
Settings	1.3%	1.6%	1.5%	1.3%	1.6%	1.2%	1.2%	1.1%	1.3%	1.4%	1.4%	1.4%	1.2%	1.3%	1.2%	1.2%	1.3%	1.1%	1.1%	1.2%	1.2%	1.3%	1.3%	1.4%	1.23%	2,178	1
Shopping	3.9%	4.5%	3.7%	3.4%	3.2%	3.2%	3.1%	3.0%	3.1%	3.3%	3.2%	3.2%	3.2%	2.8%	2.9%	2.9%	2.7%	2.7%	2.7%	2.7%	2.7%	2.8%	3.1%	3.6%	2.96%	2,556	198
Social	5.7%	5.0%	4.9%	4.3%	4.2%	4.0%	4.4%	5.1%	5.3%	5.4%	5.2%	5.0%	4.7%	4.8%	4.9%	4.5%	4.5%	4.6%	4.6%	4.9%	5.2%	5.4%	5.8%	5.7%	4.77%	1,902	342
Sports	0.5%	0.3%	0.3%	0.2%	0.3%	0.3%	0.2%	0.3%	0.3%	0.3%	0.3%	0.4%	0.4%	0.6%	0.7%	0.8%	0.9%	0.8%	0.6%	0.6%	0.7%	0.8%	0.7%	0.7%	0.56%	571	215
Themes	0.2%	0.1%	0.2%	0.3%	0.4%	0.4%	0.4%	0.2%	0.2%	0.2%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.2%	0.1%	0.1%	0.1%	0.14%	249	231
Tools	10.9%	12.2%	14.6%	17.6%	20.3%	21.5%	21.4%	18.6%	14.7%	10.4%	8.4%	6.8%	6.1%	5.9%	5.9%	5.9%	6.0%	6.1%	5.8%	6.0%	6.3%	6.8%	7.4%	9.1%	7.89%	2,512	1,688
Travel	1.4%	1.6%	2.1%	2.2%	2.4%	2.6%	2.2%	1.9%	2.0%	2.1%	2.0%	1.8%	1.9%	1.9%	1.8%	2.0%	1.9%	2.2%	2.2%	1.9%	1.7%	1.6%	1.4%	1.86%	1,752	407	
Unknown	4.7%	5.3%	5.1%	5.0%	5.3%	4.4%	5.0%	5.9%	4.6%	4.4%	4.1%	3.8%	3.5%	3.8%	3.7%	3.7%	4.0%	3.6%	3.7%	3.7%	3.9%	4.1%	4.5%	3.88%	2,284	1,796	
Total Launches per Hour	103,604	77,053	53,633	40,332	33,438	30,949	38,161	56,895	83,488	109,550	127,069	142,642	158,876	168,082	169,018	172,935	173,963	179,801	184,012	176,050	163,080	153,835	141,303	123,639			

<sup>1</sup> Aplicaciones, Aplicaciones móviles o App son términos utilizados indistintamente en este documento

Con relación a la creciente adopción de este tipo de aplicaciones, se destacan fenómenos como el incremento del número de usuarios<sup>2</sup> y la utilización de dispositivos móviles en sitios tan diversos como el hogar (96%), el trabajo (69%), al movilizarse (83%), en tiendas (76%) y restaurantes (70%), entre otros (Ipsos MediaCT - The Media, 2012). Este escenario muestra un gran mercado relacionado con el desarrollo de Apps, impulsando el crecimiento de un nuevo sector empresarial tecnológico, cuya oferta ya ha sobrepasado el número de las 300.000 aplicaciones en el caso del sistema operativo para dispositivos móviles Android<sup>3</sup> (Aharony, 2012) (Chawla, 2012) (Yan, y otros, 2012); para el cual existen cerca de 500 modelos diferentes de dispositivos de distintos fabricantes. Este mismo fenómeno se observa claramente en otros mercados de aplicaciones como el de Apple<sup>4</sup> y Windows<sup>5</sup> (Xu, y otros, 2011). En noviembre de 2011, se reportó la existencia de alrededor de 200 millones de Smartphone activos al rededor del mundo, con al menos 500,000 dispositivos activados diariamente (Whitney, 2011). Actualmente Android tiene la cuota más alta del mercado, con el 43% de las ventas de dispositivos en agosto de 2012 (Gartner, 2012), con tendencia a mantenerse.

El crecimiento acelerado de este mercado beneficia en gran medida al usuario, quien tiene a su disposición una gran variedad de opciones desarrolladas para satisfacer sus necesidades específicas. Sin embargo, en este escenario es cada vez más difícil identificar aplicaciones útiles entre los cientos de miles disponibles (Ma, y otros, 2012) (Karatzoglou, y otros, 2012), convirtiéndose la búsqueda en un proceso tedioso debido a la inmensa oferta y las limitaciones morfológicas de los dispositivos de acceso<sup>6</sup>, que restringen el despliegue de información y la inserción de consultas (Balrunas, 2011) (Ricci, y otros, 2011). Ante esto, las tiendas de aplicaciones han permitido a sus usuarios realizar búsquedas por palabras clave o sugerir las más utilizadas en diferentes categorías. Sin embargo, a pesar de las facilidades brindadas, los métodos de búsqueda provistos por las tiendas de aplicaciones no tienen en cuenta la estrecha relación entre contexto y consumo de aplicaciones, entregando recomendaciones generalizadas mediante un índice de aplicaciones organizado por popularidad, dejando abierto el problema de *brindar una asistencia oportuna y personalizada en la búsqueda de aplicaciones móviles*.

## 1.1. Contexto de investigación

En respuesta a los problemas identificados, producidos por la sobrecarga de información (Maes, 1994), nacieron los Sistemas de Recomendación (SR), cuyo grado de importancia ha llegado a tal punto que el usuario prácticamente necesita de ellos, al no tener el suficiente conocimiento para realizar una decisión autónoma ante la gran cantidad de opciones presentes (Ricci, y otros, 2011). En esta área se destaca el avance en el desarrollo de SR diseñados para dispositivos móviles (Setten, y otros, 2004) (Yuan, y otros, 2003) (Kramer, y otros, 2006) (Ricci, 2005) (Park, y otros, 2007) (Nguyen, y otros, 2007) (Horozov, y otros, 2006) (Bellotti, y otros, 2008), debido a las consideraciones adoptadas por sus restricciones en interfaz (Maes, 1994), ya que en general, los terminales móviles sacrifican el tamaño de

---

<sup>2</sup> Por ejemplo: Estados Unidos pasó de un 31% en 2011 a un 38 % en 2012, al igual que el Reino Unido en donde la posesión de Smartphone pasó de un 30% en 2011 a un 45% en 2012 [IpsosMediaCTTheMedia2012a]

<sup>3</sup> <http://www.android.com/>

<sup>4</sup> <http://www.apple.com/itunes/>

<sup>5</sup> <http://www.windowsstore.com/>

<sup>6</sup> Dispositivos de acceso, dispositivos, Smartphone, dispositivos móviles, celulares, son utilizados indistintamente para describir al dispositivo móvil del usuario

su pantalla (aproximadamente 3.5-pulgadas) y facilidades de inserción de datos<sup>7</sup>, a cambio de mayor portabilidad para el usuario (Baltrunas, 2011).

Los **SR** son herramientas poderosas (técnicas y herramientas software) que ayudan a los usuarios a manejar la sobrecarga de información entregando recomendaciones personalizadas de varios tipos de productos y servicios. Los SR son sistemas de procesamiento de información que activamente reúnen varios tipos de datos para construir sus recomendaciones. Ellos son aplicaciones inteligentes capaces de identificar y sugerir productos, información o servicios que mejor encajen con las necesidades y preferencias del usuario (Baltrunas, 2011) (Ricci, y otros, 2011). Generalmente para implementar su función principal (identificar los ítems útiles para el usuario) un SR debe predecir que ítem cuenta con mayor valor para la recomendación. Para lograr esto, el sistema debe ser capaz de predecir la utilidad de algunos de los ítems, o al menos comparar la utilidad de algunos de ellos, y entonces decidir que ítems recomendar basado en esa comparación (Ricci, y otros, 2011). De este modo los SR logran establecer que características de los ítems satisfacen las necesidades de los usuarios y sus preferencias siendo de especial utilidad en procesos de toma de decisiones. En la industria existen varios ejemplos de estos sistemas, que generalmente se especializan en la predicción de las preferencias del usuario basados en el historial de sus descargas y calificaciones (Ma, y otros, 2012), como es el caso de las recomendaciones de libros de Amazon<sup>8</sup>.

Es en esta área donde surgen los sistemas de recomendación de aplicaciones móviles o ARS, definidos como poderosas herramientas software diseñadas para lidiar con la sobrecarga de información relacionada con aplicaciones móviles expuesta al usuario (Woerndl, 2007). El objetivo de estos sistemas es entregar recomendaciones personalizadas encontrando las aplicaciones más relevantes a través de la recolección de diferentes tipos de información (Ricci, y otros, 2011). En este sentido, se destacan aquellos sistemas conscientes del contexto, en los cuales, una de las principales fuentes de información es el móvil en sí, capaz de capturar información relacionada con el contexto y consumo del usuario, produciendo grandes mejorías respecto a los sistemas que no tienen en cuenta estos factores (Karatzoglou, y otros, 2012). De la misma forma, se resaltan fuentes de información como redes sociales las cuales han sido de suma utilidad para el mejoramiento de estos sistemas al ofrecer información relevante sobre las tendencias, gustos y preferencias del usuario (Pan, y otros, 2011) (Liu, y otros, 2010). Así, siguiendo esta línea, el área de acción del presente proyecto son los *sistemas de recomendación de aplicaciones móviles conscientes del contexto del usuario*, y el problema de investigación abordado consiste en:

**¿Cómo especializar el número de resultados relevantes de los sistemas actuales de recomendación de aplicaciones móviles?**

## 1.2. Motivación

En esta sección, se expone un escenario que ejemplifica el entorno para la recomendación de aplicaciones móviles, por medio de la descripción de una situación en la cual se

---

<sup>7</sup> Muchos de los dispositivos tienen un teclado de 12 teclas o un teclado en pantalla diseñado principalmente para insertar consultas basadas en palabras clave.

<sup>8</sup> <http://www.amazon.com>

presentará los beneficios de los sistemas de recomendación de aplicaciones móviles conscientes del contexto del usuario.

Imagine a Bob, un joven que acaba de comprar un Smartphone. Él es un estudiante universitario interesado en explorar todas las funcionalidades que le ofrece su nuevo dispositivo. Así, en su primera mañana explora la tienda de aplicaciones de Google y encuentra varios juegos entretenidos mientras se dirige a la universidad y deja pasar su autobús. Una vez en clase, se prepara para tomar nota de las indicaciones del profesor, mientras por un descuido, deja caer su nueva adquisición al piso. Luego en su restaurante retoma su búsqueda de aplicaciones, encontrando varios juegos que son muy útiles mientras esperar su almuerzo. En la tarde, después de salir de clase se dirige hacia su casa, mientras escucha música desde su celular. Una vez en su casa prende su computador y revisa que esa tarde sus compañeros se reunieron para hacer el trabajo que el profesor puso en la mañana al comienzo de la clase. De esta manera Bob decide dedicar tiempo de la noche a terminar con sus labores e ir a descansar muy tarde.

Al día siguiente Bob revisa las aplicaciones que ha instalado y se encuentra con “Vanilla” una de las tantas apps que instaló el día anterior, la cual se encarga de generar recomendaciones dependiendo del contexto del usuario. Así, al desayunar decide probar la recomendación dada por esta aplicación, obteniendo la sugerencia de una aplicación de noticias para salir de la casa bien informado. Al momento de salir, revisa nuevamente la recomendación y encuentra un App de clima demasiado útil, que le alerta sobre probabilidades de lluvia en 20 minutos. Además, encuentra la aplicación para transporte público de su ciudad, la cual le informa que en 4 minutos pasará su bus. Luego, antes de entrar a clase Bob vuelve a consultar las recomendaciones, y encuentra que debido a su ubicación, “Vanilla” le sugiere instalar una aplicación para la toma de notas en clase, la cual le es muy útil ayudándole a comprender el tema más fácilmente, ahorrándole mucho tiempo. Al almuerzo “Vanilla” le recomienda una variada oferta de juegos para entretenerse mientras espera su comida. En la tarde antes de salir “Vanilla” le sugiere a Bob un conjunto de aplicaciones sociales para encontrarse con sus amigos. Al utilizar estas aplicaciones, Bob se entera del cumpleaños de uno de sus compañeros y se dirige a celebrar la ocasión. Una vez en casa “Vanilla” le recomienda a Bob una aplicación de lectura, y de esta manera, entretenido con un muy buen libro, Bob termina su día.

Claramente en el ejemplo miramos un escenario simple en el cual se encuentra a un usuario novato en el uso de aplicaciones móviles, el cual se da a la tarea de encontrar las más útiles en el gran mercado de Google Play. Miramos que en el primer día, él pierde demasiado tiempo buscando y explorando aplicaciones, encontrando algunas muy interesantes pero poco útiles para su contexto actual, proceso que genera un sin número de problemas. Sin embargo al segundo día su búsqueda está soportada por un sistema de recomendación basado en contexto, el cual le sugiere aplicaciones muy acordes a su situación actual, ayudándole a llevar de mejor manera su día, permitiéndole explotar las capacidades de su teléfono móvil, para su propio beneficio.

### **1.3. Premisas e Hipótesis**

De esta manera, esta sección presenta la manera en que este trabajo *i)* concilia las aproximaciones de las principales investigaciones en esta área y *ii)* propone nuevas aproximaciones para aportar en este dominio.

### 1.3.1. Premisas

En esta sección se presenta las principales premisas que, fundamentadas en los trabajos de investigación más influyentes y relacionados con la temática de esta monografía, soportaron la propuesta de este proyecto. Un análisis más detallado de los mismos, puede ser encontrado en el capítulo dedicado al estado del arte. En este sentido las premisas utilizadas al interior de este proyecto son las siguientes:

**P1. La aceptación de la recomendación de aplicaciones móviles depende del contexto físico actual del usuario.** Tal como fue mencionado en la introducción de este capítulo, de acuerdo a diversos estudios, se ha establecido una fuerte relación entre el consumo de aplicaciones móviles y el contexto actual del usuario (Falaki, y otros, 2010) (Böhmer, y otros, 2011). A partir de esa premisa, varios trabajos de investigación en ARS han utilizado información contextual en la generación de sus aplicaciones móviles, teniendo resultados que demuestran una mejora significativa en la aceptación de las recomendaciones por parte del usuario final. Ejemplo de estos trabajos son los presentados por Woenrdl (Woenrdl, 2007), Böhmer (Böhmer, y otros, 2010) y Karatzoglou (Karatzoglou, y otros, 2012), entre otros.

**P2. La predicción de calificaciones de aplicaciones móviles basada en el consumo, tiene un impacto positivo en los ARS.** Uno de los principales problemas de los sistemas de recomendación, el cual es especialmente crítico en sistemas de FC, está relacionado con la poca interacción del usuario con los métodos explícitos de captura de información. Un ejemplo de esto es la cantidad abrumadora de calificaciones vacías encontradas en los registros de apreciación (ratings), hecho que impide garantizar un nivel de calidad aceptable en las recomendaciones (Burke, y otros, 2011) (Ricci, y otros, 2011). Ante este problema, en varios trabajos de investigación en ARS se ha propuesto el cálculo implícito de ese tipo de apreciaciones, con base en los indicadores de consumo de aplicaciones móviles, consiguiendo resultados satisfactorios, tal como lo presentan los estudios de Yan (Yan, y otros, 2011) y Karatzoglou, entre otros (Karatzoglou, y otros, 2012).

**P3. La utilización de Apps de una(s) categoría(s) en especial, depende del contexto actual del usuario.** Según varias investigaciones realizadas tanto en el área de usabilidad, consumo de energía y predicción de la localización del usuario en entornos móviles y en varios sistemas operativos móviles, se ha logrado observar una relación entre el contexto actual del usuario y las categorías de aplicaciones móviles consumidas. Muestra de ello se encuentran los trabajos de Böhmer (Böhmer, y otros, 2011), Yan (Yan, y otros, 2012) y Falaki (Falaki, y otros, 2011), entre otros. Estos trabajos se destacan por la exhaustividad de los datasets empleados para comprobar esta afirmación; donde la información utilizada ha sido recolectada durante periodos de tiempo prolongados, desde la infraestructura de operadores de telecomunicaciones y dispositivos móviles, entre otras fuentes.

### 1.3.2. Hipótesis

En esta sección serán presentadas las hipótesis trabajadas a lo largo de la presente investigación, las cuales han guiado este proceso, con el fin de explorar nuevas formas de aplicación de la información contextual y de consumo del usuario dentro de sistemas de recomendación de aplicaciones móviles, especialmente los de filtrado colaborativo. Así, las siguientes son las hipótesis trabajadas en esta investigación:

**H1a. El uso de calificaciones inferidas teniendo en cuenta el consumo y el contexto físico del usuario generan una mejor calidad en los ARS.** Esta hipótesis está motivada por las premisas P1 y P2, la cuales de manera independiente muestran una relación entre contexto, consumo y calificaciones en el dominio de los ARS. A partir del estudio presentado en el estado del arte de esta monografía, se logró determinar que a pesar de la existencia de trabajos que exploran esta hipótesis, se tiene que la información de consumo utilizada por ellos se limita la cantidad de veces que una aplicación es lanzada (frecuencia de consumo) Karatzoglou (Karatzoglou, y otros, 2012). En cambio, trabajos que utilizan una aproximación más completa en términos del consumo, considerando tanto frecuencia, periodo y el carácter reciente del consumo, no consideran características contextuales para el cálculo de las apreciaciones de los usuarios (Yan, y otros, 2011). Por lo tanto en el presente proyecto se diseñaron técnicas de recomendación con el uso de información completa del consumo (Frecuencia, Duración y carácter reciente) en conjunto con información sobre el contexto físico del usuario, para determinar el efecto de la utilización de este tipo de información sobre la calidad de los ARS.

**H2a. Otorgar una mayor relevancia a la apreciación de usuarios expertos dentro de la red social del usuario por cada categoría de aplicaciones, incrementa la calidad de las recomendaciones de los ARS basados en FC.** Esta hipótesis nace de la premisa P3, en la cual se establece la relación entre el contexto actual del usuario y su tendencia a consumir aplicaciones de ciertas categorías. En este sentido, se parte de la idea general que si existen categorías “relevantes” dependiendo del contexto, aquellos usuarios expertos en dichas categorías que hacen parte de la red social del usuario, ofrecerán recomendaciones más apropiadas en tales situaciones teniendo en cuenta la influencia que su cercanía social representa y su misma experticia. Después de la revisión bibliográfica y en el mejor conocimiento del grupo de trabajo del proyecto, no se conocen trabajos que aborden esta aproximación.

## 1.4. Objetivos

Una vez expuesto el enfoque de este proyecto, se presenta los objetivos de investigación, mediante los cuales se persigue explorar las hipótesis previamente presentadas.

### 1.4.1. Objetivo general

Especializar el número de resultados relevantes en el descubrimiento de aplicaciones móviles de acuerdo a la información contextual y social del usuario.

### 1.4.2. Objetivos específicos

1. Modelar la información contextual de consumo de aplicaciones en entornos Móviles.
2. Adaptar técnicas de filtrado colaborativo basadas en la información contextual del usuario, relaciones sociales y el consumo de las aplicaciones.
3. Evaluar experimentalmente los algoritmos y técnicas propuestos, por medio de un prototipo en un entorno de prueba.

## 1.5. Contribuciones y Principales Resultados.

Las principales contribuciones de éste proyecto de grado son:

- **Análisis detallado de la captura de información del contexto del usuario.** Los principales resultados de éste análisis son: (a) métodos de captura de información del contexto físico del usuario a partir de la información entregada por los sensores del dispositivo móvil del usuario, (b) métodos de captura de la información del contexto social del usuario, a partir de las información recolectada en su dispositivo móvil y (c) una exploración sobre la captura explícita de información sobre el contexto cognitivo del usuario.
- **Definición de un modelo de información contextual y de consumo.** Este es uno de los más importantes aportes del presente proyecto de grado, debido a que esta definición estuvo soportada por i) una amplia exploración sobre las diversas definiciones de contexto existentes en la literatura, ii) consideraciones prácticas en el diseño de aplicaciones móviles y iii) un análisis de la utilidad potencial de los tipos de información considerados al interior de sistemas de recomendación de aplicaciones móviles.
- Definición de técnicas de inferencia de calificaciones basadas en consumo para aplicaciones móviles, enfocadas en sistemas de FC.
- Definición de técnicas de cálculo de vecindad de usuarios basada en similitud de consumos, contextos de uso, cercanía social y experticia de los usuarios.
- **Definición de una arquitectura modular y eficaz para sistemas de recomendación de aplicaciones móviles.** Este aporte es el resultado del análisis a nivel de ingeniería de varios trabajos en sistemas de recomendación, teniendo como objetivos principales la modularidad y efectividad del sistema propuesto.
- **Prototipo funcional para la recomendación, optimizado en un entorno real.** Este es uno de los principales aportes, debido a que pese al carácter académico del proyecto, el prototipo funcional, fruto de la implementación de las técnicas y algoritmos fue realizado con características de un sistema de producción; resaltándose su estética y funcionamiento; reflejándose en estos aspectos los fuertes fundamentos en ingeniería de los integrantes del equipo de trabajo. Es de resaltar que el prototipo tuvo varias modificaciones en términos técnicos provocadas por la realimentación recibida por parte de los usuarios del sistema.
- **Prototipo para la evaluación de técnicas de recomendación de aplicaciones móviles.** Esta es una extensión del prototipo funcional de la recomendación el cual permite el cálculo de las medidas de calidad del sistema propuesto, así como la eficiencia del mismo.
- Motivar el desarrollo y la investigación sobre sistemas de recomendación en la comunidad académica de la Universidad del Cauca, implementando un prototipo de recomendación de aplicaciones móviles el cual crea un precedente alrededor de éste tema, y despertar así el interés por estas tecnologías en la comunidad investigativa del país, debido al positivo panorama social y económico factible de obtener con su difusión. Además, a partir del proceso seguido en este proyecto de investigación, se logró fundamentar la nueva materia electiva del departamento de telemática “Web Social”.
- Este trabajo soporta el componente de captura de información contextual en la tesis de maestría: **Recomendación de aplicaciones móviles basada en el análisis del**

**comportamiento contextual del usuario**, desarrollada al interior de la Universidad del Cauca.

## 1.6. Contenido de la Monografía.

La presente monografía se encuentra formada por tres partes que en su orden corresponden a introducción, propuesta y experimentación.

### Parte I. Introducción

Esta primera parte está dedicada a la descripción del entorno de investigación del presente proyecto de grado, los principales conceptos relacionados y los trabajos más significativos en el área de recomendación de aplicaciones móviles basada en información social y contextual. Esta parte está conformada por los siguientes capítulos.

- **Capítulo I.** Introducción  
Este capítulo está dedicado a presentar el entorno de investigación considerado en el presente proyecto de grado.
- **Capítulo II.** Estado del Arte  
Presenta una visión general sobre los trabajos relacionados y los conceptos que giran en torno a los SR y ARS.

### Parte II. Propuesta: modelos, técnicas y algoritmos

La segunda parte de esta monografía, está enfocada en la presentación de los avances propios realizados al interior del presente proyecto de investigación.

- **Capítulo III.** Modelo de información contextual y Calificación por análisis de Consumo  
En este capítulo se presenta las consideraciones para la captura de la información contextual a partir de las facilidades provistas por los dispositivos móviles, además se introduce el modelo de información contextual que servirá de referencia para el planteamiento de las técnicas y algoritmos desarrollados al interior de este proyecto de investigación. Posteriormente se enfatizará en el análisis del consumo de aplicaciones del usuario y la manera en que se relaciona con el cálculo automático de calificaciones. La motivación para la formulación de las técnicas propuestas, es la generación de un método implícito de captura de la apreciación del usuario, basado en información “completa” sobre el consumo de aplicaciones.
- **Capítulo IV.** Vecindad, predicción y recomendación  
Este capítulo presenta las diferentes aproximaciones basadas en FC propuestas en este proyecto soportadas por las técnicas de análisis de la información de consumo, contextual y social del usuario.

### Parte III. Experimentación

Corresponde a la última parte de la monografía, en la cual se presentan los elementos utilizados para la evaluación de los modelos, técnicas y algoritmos propuestos en la segunda parte.

- **Capítulo V.** Vanilla: prototipo y arquitectura



En este capítulo es presentada la arquitectura del prototipo de recomendación utilizado para probar los diferentes algoritmos de recomendación generados al interior de esta propuesta.

- **Capítulo VI.** Experimentación y Evaluación.  
El capítulo de experimentación y evaluación, contiene los fundamentos teóricos de la evaluación de sistemas de recomendación, así como el plan de pruebas y resultados, fruto de este proceso.
- **Capítulo VII.** Conclusiones y trabajos futuros

Presenta las conclusiones y algunas perspectivas de la investigación, analizando los resultados del trabajo realizado y generando un conjunto de recomendaciones importantes para el desarrollo de trabajos futuros.

## Capítulo II, Estado Del Arte

Esta sección está dedicada a la descripción de los principales conceptos y trabajos de investigación relacionados con el presente proyecto. Así, se presenta las bases teóricas que permitirán un profundo análisis de los tipos de sistemas de recomendación más importantes, exponiendo sus ventajas y desventajas. Finalmente, como producto del análisis de los trabajos relacionados, se presentará las brechas existentes con los principales problemas abordados en el presente proyecto de grado.

### 2.1. Contexto General

Esta sección contiene conceptos relevantes para la comprensión y posterior análisis del problema central del presente trabajo de grado. Entre los conceptos más importantes se destacan los relacionados con sistemas de recomendación, filtrado colaborativo y contexto del usuario.

#### 2.1.1. Sistemas de recomendación

Originalmente, los sistemas de recomendación fueron definidos como plataformas donde “los usuarios proveen recomendaciones (entradas), que son redirigidas por el sistema, a los destinatarios más adecuados (salidas)” (Resnick, y otros, 1997). Sin embargo, en la actualidad, el alcance de la definición de este tipo de sistemas alcanza una mayor connotación, abarcando a todo sistema capaz de producir recomendaciones **individuales** y **personalizadas**, o capaz de dirigir al usuario a objetos/ítems útiles de forma **personalizada**, a través de un amplio conjunto de posibilidades (Burke, 2002). De esta manera, los SR se distinguen por su enfoque hacia la “individualidad” y “personalización”, el cual los diferencia notoriamente de los *Sistemas de recuperación de información*<sup>9</sup>.

Un sistema de recomendación cuenta con tres componentes generales: (i) información de fondo o “Background data” que se describe como la información en el sistema antes de empezar el proceso de recomendación, (ii) información de entrada o “input data” que se identifica como la información que el usuario solicitante debe proveer al sistema, y (iii) algoritmos de agregación, encargados de combinar los dos tipos de información mencionados, con el fin de generar una recomendación (Sami, y otros, 2009). Teniendo en cuenta lo anterior, es posible establecer diferentes técnicas de recomendación que pueden clasificarse de acuerdo a las fuentes de información y al tipo de procesamiento utilizado, sobresaliendo las siguientes: recomendación por contenido, recomendación colaborativa, recomendación por utilidad, recomendación por demografía y recomendación por conocimiento (Resnick, y otros, 1997) (Schafer, y otros, 1999) (Terveen, y otros, 2001).

##### 2.1.1.1. Técnicas basadas en contenido

Son técnicas que presentan una fuerte dependencia con el dominio de recomendación. Generalmente, son asociadas a una especie de filtrado de información (Belkin, y otros, 1992), donde el principal objeto de estudio es el perfil del ítem a recomendar. Usualmente, en este tipo de sistemas, el perfil del usuario se encuentra compuesto por información

---

<sup>9</sup> **Sistemas de recuperación de información:** También conocidos como motores de búsqueda, son aquellos sistemas que retornan la información más apropiada frente a una solicitud de búsqueda e.g. “Matching” (Burke, 2002).

demográfica y sus preferencias asociadas. Estas preferencias están fuertemente relacionadas con las características utilizadas para la descripción de los ítems a recomendar, y son inferidas utilizando métodos de aprendizaje como árboles de decisión, redes neuronales o representaciones basadas en vectores. El proceso de recomendación utilizado en este tipo de técnicas considera el cálculo de una correlación entre ítems con el fin de encontrar a aquellos que guarden un alto grado de similitud con las preferencias del usuario, siendo éstos los más aptos a ser recomendados (Burke, 2002).

### **2.1.1.2. Técnicas basadas en utilidad**

Este tipo de técnicas se basa en el supuesto que la aceptación de un ítem por parte de un usuario, se encuentra estrechamente relacionada con la *utilidad*. Así, lo que se pretende es realizar recomendaciones teniendo en cuenta la *utilidad* de cada objeto según la necesidad del usuario, empleando técnicas de emparejamiento o “matching” sobre perfiles que representen tanto las necesidades y limitaciones del usuario, como las características y requerimientos de los ítems.

En este orden de ideas, el sistema estima la relevancia de la recomendación empleando *técnicas de satisfacción de restricciones* o “*constraint satisfaction techniques*”: Estas técnicas en Inteligencia Artificial se encargan de encontrar una solución teniendo en cuenta un conjunto de restricciones o “constraints” que imponen condiciones que las variables de la solución deben resolver. La solución es entonces, un vector de variables que satisfacen todas las restricciones para identificar el mejor acierto o “match” (Burke, 2002).

### **2.1.1.3. Técnicas basadas en conocimiento**

Los enfoques basados en conocimiento (*Knowledge-based approaches*), se distinguen del resto al tener conocimiento representado a través de estructuras, que en el caso de los sistemas de recomendación, son perfiles que ayudan a inferir que tan oportuna es una recomendación de acuerdo a las necesidades del usuario. Google por ejemplo, toma la solicitud del usuario cómo la única estructura de conocimiento y de acuerdo a la información de los links entre sitios web, puede determinar cuáles son los sitios más recomendables de acuerdo a su popularidad. Otros sistemas emplean técnicas basadas en el razonamiento de casos (*case-based reasoning*) para realizar recomendaciones, en donde el conocimiento es obtenido a través de encuestas u otras interfaces (Burke, 2002).

### **2.1.1.4. Técnicas demográficas**

Estos sistemas categorizan al usuario de acuerdo a sus características y atributos personales, realizando recomendaciones basados en “Clases Demográficas” (estereotipos) las cuales describen grupos o categorías de personas con características comunes (Burke, 2002). Generalmente, para la construcción de estas categorías son utilizadas diferentes técnicas de clasificación y categorización. Estas técnicas por su naturaleza pueden ser manuales o automáticas. Ejemplo de técnicas manuales de clasificación son los estudios de mercadeo, los cuales producen como resultado “grupos demográficos” que representan nichos para la comercialización de productos. Otro ejemplo de estas técnicas es el emparejamiento entre perfiles de usuarios y estereotipos previamente definidos, que como resultado producen un número de grupos de usuarios igual al número de estereotipos utilizados para la clasificación (Krulwich, 1997). Por otro lado, las técnicas automáticas optan por el uso de máquinas de aprendizaje alimentadas por la información que los usuarios brindan a través de encuestas, entre otros. En este orden de ideas, las técnicas de

recomendación basadas en información demográfica, forman correlaciones entre usuarios similares a las que se ven en los sistemas de filtrado colaborativo, sin embargo, uno de los beneficios de estos sistemas frente a otros, es que no necesitan de ningún tipo de historial de calificaciones o consumo.

### **2.1.1.5. Técnicas colaborativas o Filtrado Colaborativo (FC)**

Es probablemente la más madura de las técnicas de recomendación implementadas hasta el momento (Burke, y otros, 2011). Permite clasificar y valorar un determinado conjunto de ítems de forma colectiva, realizando recomendaciones con base en el historial de calificaciones y la similitud entre usuarios (Ekstrand, y otros, 2011) (Melville, y otros, 2010). Las calificaciones hechas por los usuarios sobre ciertos ítems, son utilizadas para predecir las calificaciones del resto de los elementos de acuerdo a la similitud entre usuarios (Galán, 2007) (Ekstrand, y otros, 2011) (Herlocker, y otros, 2000) otorgando mayor relevancia a ciertas valoraciones con el fin de lograr una mayor afinidad entre las predicciones y los gustos del usuario (Ekstrand, y otros, 2011). En la mayoría de los casos se deja a la comunidad la tarea de evaluar cada elemento y se utilizan algoritmos de minería de datos para establecer relaciones entre usuarios de una manera más precisa. Existen dos enfoques del filtrado colaborativo: Filtrado Colaborativo basado en memoria vecinos cercanos y el Filtrado colaborativo basado en modelos (Ekstrand, y otros, 2011) (Herlocker, y otros, 2000) (Su, y otros, 2009).

#### **El Filtrado Colaborativo basado en memoria (usuario-usuario)**

Trata de encontrar usuarios con perfil, comportamiento e historial de consumo similar, para ponderar sus preferencias y generar la lista de recomendación (Ekstrand, y otros, 2011) (Su, y otros, 2009) (Galán, 2007). Existen diferentes tipos de FC basado en memoria entre los cuales se destacan: el “*ítem – ítem*” y el “*usuario – usuario*”. El primero es uno de los algoritmos más implementados, y en su dinámica descarta la similitud entre usuarios y utiliza la similitud entre patrones de ratings (Ekstrand, y otros, 2011). Por ejemplo, si dos ítems tienden a obtener los mismos ratings de los mismos usuarios, entonces estos son similares. Por otro lado el FC de tipo usuario-usuario, tiene un enfoque más cercano a la filosofía social del FC. La dinámica de éste consiste en predecir el rating de un ítem no consumido por el solicitante, teniendo en cuenta los ratings que otros usuarios, con gustos afines, han definido para dicho ítem.

#### **El Filtrado Colaborativo basado en modelos**

Infiere un modelo derivado del historial de calificaciones, el cual es utilizado para realizar recomendaciones. Este tipo de recomendadores implementan varias técnicas de aprendizaje incluyendo redes neuronales (Jennings, y otros, 1993), redes bayesianas (Condliff, y otros, 1999) o indexación semántica latente (Foltz, 1990).

### **2.1.2. Comparación de técnicas de recomendación**

Después de la descripción de las principales técnicas de recomendación encontradas en la literatura, a continuación se presenta una tabla resumen, que expone las principales propiedades de las mismas. Seguido, se expone una segunda tabla que describe las ventajas y desventajas de cada una de ellas, analizadas con respecto a un conjunto de características relevantes dentro del dominio de los sistemas de recomendación.

**Tabla 2.** Descripción de los componentes de cada técnica de recomendación.

Técnica	Información de fondo o “Background”	Entrada (input data)	Algoritmo
<b>Colaborativa</b>	Ratings de <b>U</b> de ítems <b>I</b> .	Ratings de <b>u</b> de los ítems <b>I</b>	Identifica a los usuarios en <b>U</b> similares a <b>u</b> , y los extrapola de acuerdo al rating de <b>i</b> (si existe para los dos).
<b>Basada en Contenido</b>	Características de ítems <b>I</b>	Ratings de <b>u</b> de los ítems <b>I</b>	Genera un clasificador que ajusta el comportamiento de <b>u</b> (inferido por sus ratings) y lo usa en <b>i</b> .
<b>Demográficas</b>	Información demográfica de <b>U</b> y sus ratings de los ítems en <b>I</b> .	Información demográfica de <b>u</b> .	Identifica a los usuarios que son demográficamente similares a <b>u</b> y los extrapola de acuerdo al rating de <b>i</b> .
<b>Basada en Utilidad</b>	Características de los ítems en <b>I</b> .	Función de utilidad sobre los ítems en <b>I</b> que describan las preferencias de <b>u</b> .	Aplica la función a los ítems y determina un ranking de <b>i</b> .
<b>Basada en Conocimiento</b>	Características de los ítems en <b>I</b> y conocimiento sobre como esos ítems coinciden con las necesidades del usuario.	Descripción de las necesidades e intereses de <b>u</b> .	Infiere una coincidencia o “match” entre <b>i</b> y la necesidad de <b>u</b> .

- **I** Conjunto de ítems sobre los cuales la recomendación se puede hacer.
- **U** Conjunto de usuarios de los que se conocen sus preferencias.
- **u** Usuario al que se le va a recomendar.
- **i** Ítem por el cual es necesario predecir las preferencias de **u**.

**Tabla 3.** Comparativa técnicas de recomendación.

Técnica	Ventajas	Desventajas
<b>Filtrado Colaborativo</b>	a, b, c, d	i, j, l, m
<b>Basado en Contenido</b>	b, c, d	i, l, m

<b>Demográficas</b>	a, b, c	i, l, m, n
<b>Basada en utilidad</b>	e, f, g	o, p
<b>Basadas en conocimiento</b>	e, f, g, h	k

### Ventajas

- a) Las recomendaciones pueden ser multi-dominio (e.g. música, libros, películas, etc.).
- b) No se requiere conocimiento o Información respecto al dominio (no se refiere a la información de los ítems).
- c) Adaptativos: La calidad se mejora a medida que pasa el tiempo
- d) Pueden soportar retroalimentación o “Feedback” implícitamente.
- e) Inmunidad ante problemas de “ramp-up” (inserción de nuevos ítems o usuarios) de usuarios y de ítems.
- f) Sensibles a cambios en cuanto a las preferencias.
- g) Incluyen características externas a las de los usuario en la recomendaciones
- h) Capacidad de mapear de las necesidades del usuario, a productos o ítems

### Desventajas

- i) Inconvenientes en la inserción de nuevos usuarios (La precisión de las recomendaciones que reciba el usuario, es proporcional a los usuarios en su vecindad, se lo llama “ramp-up” de usuarios).
- j) Inconvenientes en la inserción de nuevos ítems (la probabilidad de que un ítem sea recomendado, es proporcional al número de calificaciones que tenga, se lo llama “ramp-up” de ítems).
- k) Ingeniería de conocimiento requerida.
- l) La calidad y precisión en la recomendaciones, depende del tamaño o del nivel de dispersión del *dataset* (historial de calificaciones) con el que se cuente.
- m) Insensibilidad a los cambios dinámicos de las preferencias de los usuarios (“Estabilidad vs. Plasticidad”)
- n) Recolección de información demográfica, necesaria.
- o) El usuario debe establecer la “función de utilidad” que le corresponde.
- p) Recomendaciones estáticas (independientes del tiempo y experiencia del sistema).

Partiendo de la comparación presentada, se puede concluir que debido al grado de madures en su desarrollo, el filtrado colaborativo es la técnica que más ventajas presenta sobre las demás técnicas de recomendación expuestas. Esto se debe en parte a que sus problemas asociados han sido atacados y minimizados por numerosos trabajos de investigación. Otro de los puntos a favor de esta técnica es el soporte a recomendaciones en múltiples dominios, crucial para su adaptación en la recomendación de aplicaciones móviles.

### 2.1.3. Filtrado colaborativo

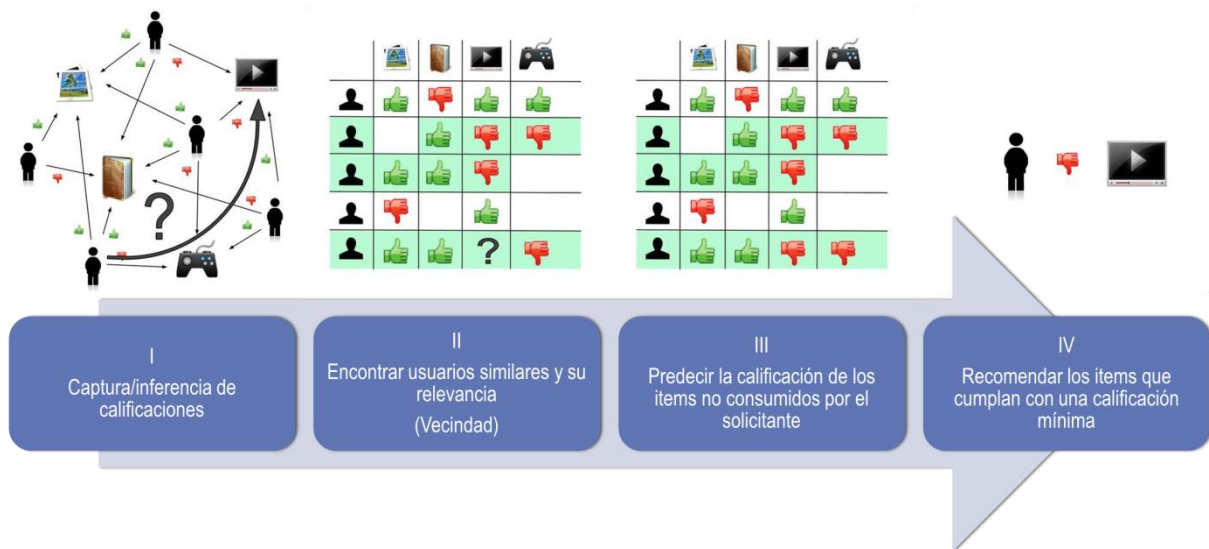
En esta sección se explicará los principales fundamentos teóricos de filtrado colaborativo, enfatizando en el proceso general que esta técnica sigue para calcular sus recomendaciones. Como fue mencionado anteriormente, esta técnica permite predecir la apreciación de ítems de manera colectiva, utilizando el historial de calificaciones generado por una comunidad.

El proceso de recomendación basado en FC está conformado principalmente por cuatro etapas descritas a continuación y presentadas en la figura 1.

#### 2.1.3.1. Captura/inferencia de Calificaciones

Esta etapa es la encargada de recolectar las calificaciones realizadas explícitamente por el usuario, con el fin de brindarle la información de “background” necesaria al sistema. Sin embargo, no siempre es posible obtener dichas calificaciones de forma óptima, y por tal razón es necesario implementar técnicas de inferencia de estos valores, basadas en el comportamiento del consumo del usuario, como en el trabajo de Yan (Yan, y otros, 2011). Una vez superado este inconveniente, la información recolectada en esta etapa puede ser matemáticamente representada como un grafo bipartito, en el cual los nodos corresponden a ítems y usuarios, siendo las aristas la evidencia de una transacción entre ellos que generalmente corresponde a la apreciación después de su consumo (rating), tal y como puede observarse en la primera sección de la figura. La información recolectada también puede ser representada como una matriz de  $N \times M$ , donde  $N$  corresponde al número de usuarios,  $M$  al número de ítems considerado y  $R_{j,i}$  es el rating concedido por un usuario  $j$  al ítem  $i$ .

En este orden de ideas, se define a la información de entrada del sistema, como el historial personal de los ratings otorgados por el usuario solicitante de la recomendación.



**Figura 1.** Proceso de recomendación de sistemas de Filtrado Colaborativo.

### 2.1.3.2. Definición de la vecindad de usuarios:

La premisa principal del filtrado colaborativo podría ser resumida en la siguiente afirmación “usuarios con gustos similares tienden a consumir ítems similares”. Por tal razón, la segunda etapa del proceso consiste en la identificación de los usuarios más similares al solicitante. En este sentido, el cálculo de la similitud entre los actores mencionados, se realiza utilizando la información sobre el historial de ratings a través de técnicas como la similitud Jaccard, cercanía de cosenos o correlación de Pearson (Sami, y otros, 2009). La medida o valor resultante de cualquiera de estas técnicas, oscila entre 0 (para los usuarios disimiles) y 1 (para los totalmente similares). A partir de este valor se elige un umbral que definirá el conjunto de usuarios pertenecientes a la **vecindad** de recomendación. Por ejemplo en la figura 1 se puede observar resaltados con verde, a aquellos usuarios pertenecientes a la vecindad del solicitante. Esta vecindad corresponde entonces a una lista de parejas (usuario, similitud) que servirá de referencia al momento de predecir el valor del rating de nuevos ítems.

### 2.1.3.3. Predicción de Calificaciones

La importancia de esta etapa radica en la manera cómo a partir de la información de la vecindad de usuarios, se puede predecir el rating de un ítem desconocido por el solicitante. Para tal fin, usualmente es utilizada la información relacionada a la vecindad del usuario más el historial de ratings de cada uno de los miembros de ella. Así, el nuevo rating predicho dependerá de los ratings asignados por los miembros de la vecindad, los cuales son ponderados de acuerdo a su grado de similitud con el solicitante.

### 2.1.3.4. Presentación de la recomendación:

Como último paso en el proceso de recomendación se encuentra la presentación de los resultados al solicitante. Para esta fase generalmente se organiza los ítems de acuerdo al valor de rating predicho en la etapa anterior. Sin embargo, es usual mirar que este último paso dependa de las características del dominio de recomendación, por ejemplo, en sistemas de recomendación de música, es inusual recomendarle al usuario una misma canción más de dos veces en un determinado periodo de tiempo.

## 2.1.4. Contexto

El contexto es un concepto multifacético que ha sido estudiado a lo largo de diferentes disciplinas, incluyendo las ciencias de la computación, ciencias cognitivas, lingüística, filosofía, psicología y ciencias organizadas (Adomavicius, 2005). Cada una de ellas tiende a referirse al contexto adaptando su definición genérica “Condiciones o circunstancias que afectan a algo” (McKechinie, 1989) a su campo de acción. Por tal razón existen tantas definiciones de contexto como número de disciplinas y sub-disciplinas que hacen uso de él.

En el área de *computación consiente del contexto*<sup>10</sup> se pueden encontrar múltiples definiciones. Por ejemplo, en los primeros trabajos Schilit y Theimer (Schilit, 1994), se define el contexto como “La localización y la identidad de las personas y objetos cercanos”. Esta definición fue ampliada en (Schilit, 1994) donde se define que: “*El Contexto abarca más que solo la ubicación del usuario, debido a que otras cosas de interés son móviles y cambiantes. El contexto incluye iluminación, nivel de ruido, la conectividad de red, los costos de*

---

<sup>10</sup> Conocida por su término en Inglés “Context-Aware computing”



*comunicación, ancho de banda de comunicación e incluso la situación social*". Varios años después Dey adapta la definición de contexto proponiendo una más abstracta (Dey, 2001): *"El contexto es una pieza de información que puede ser utilizada para caracterizar la situación de una entidad. Una entidad es una persona, lugar u objeto que se considera relevante para la interacción entre un usuario y una aplicación"*. Esta definición, es probablemente la primera definición de contexto que se adoptó ampliamente en las ciencias de la computación (Baltrunas, 2011). En sistemas de recomendación el término entidad se refiere usualmente al usuario, ítem y la experiencia que el usuario está evaluando. De esta manera para el presente trabajo, el contexto se define como el **conjunto de variables o piezas de información del usuario, que influyen en el uso de una aplicación en un determinado tiempo, como pueden ser su entorno o estado actual**. De la misma forma se ha determinado un conjunto de clases de contexto que pueden servir para describir los tipos de información relacionada con el contexto del usuario (Luis Rojas-Potosi, 2012) (Jung, 2009), las cuales serán descritas a continuación.

#### **2.1.4.1. Información de Contexto físico**

Este tipo de información describe las condiciones ambientales y geográficas del usuario. En el presente proyecto la fuente más importante de este tipo de información son los sensores del dispositivo móvil del usuario. Una descripción detallada sobre la información que puede ser capturada por medio de los sensores de los dispositivos móviles, se puede encontrar en la página de soporte al desarrollador, que tiene el sistema operativo Android<sup>11</sup>. Otra fuente de información de esta categoría son los servicios web dedicados a la entrega de información ambiental geo-referenciada.

#### **2.1.4.2. Información de Contexto Cognitivo**

Corresponde a la información relacionada con todo el espectro de los datos del contexto cognitivo, el cual incluye: preferencias, estados mentales, tareas y afinidades sociales de los usuarios humanos (Jung, 2009). La información de este tipo implica que la entidad a analizar es el usuario.

#### **2.1.4.3. Información social o Contexto social**

La información social corresponde a los datos asociados a la interacción del usuario con otros usuarios. Dentro de esta información se puede identificar dos campos de interés: en primera instancia se encuentra el perfil del usuario, que corresponde a un conjunto de atributos que caracterizan y representan de manera digital la identidad de una persona App (Appelquist, y otros, 2010) (Viswanath, y otros, 2010). En segunda instancia se encuentran las relaciones sociales, que se definen como aquellas interacciones<sup>12</sup> que se encuentran reguladas por normas sociales (Hogg, y otros, 2008) (Pabjan, 2005), como es el caso de las relaciones de amistad, familiares, sentimentales, laborales, etc (Hogg, y otros, 2008).

En este orden de ideas, y de acuerdo a la teoría de redes sociales (Adamic, 2012), este tipo de estructuras puede representarse a través de grafos, cuyos componentes principales corresponden a nodos y aristas, o perfiles y relaciones respectivamente; donde dependiendo del dominio o tipo de red, un nodo puede ser un vértice, un sitio, un actor, etc. y una arista

---

<sup>11</sup> [http://developer.android.com/guide/topics/sensors/sensors\\_overview.html](http://developer.android.com/guide/topics/sensors/sensors_overview.html)

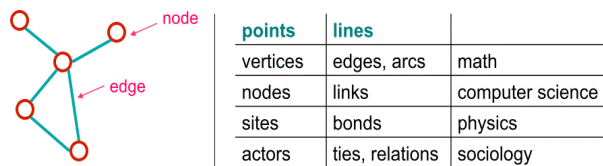
<sup>12</sup> **Interacción:** Acción que se ejerce recíprocamente entre dos o más individuos (personas en este caso). e.g. Llamada, Sms, mensaje, encuentro, etc. [16]

puede ser un lazo, una relación, un enlace, un link, etc., tal como se puede observar en la figura 2.

Debido al carácter genérico de la definición de información social, existen varios tipos de redes que pueden ser catalogadas bajo esa categoría. En este sentido, se resalta el trabajo de Aharony (Aharony, 2012), donde una de sus principales contribuciones fue la generación del data-set de información social más completo recolectado hasta el 2012, el cual muestra resultados muy interesantes que evidencian la influencia de diferentes tipos de redes sociales sobre las decisiones que las personas toman en diversos dominios que van desde la salud, hasta el comportamiento de consumo en diferentes mercados. Su experimento consistió en la observación de en una comunidad seleccionada de familias jóvenes durante un periodo de 15 meses, donde los autores identificaron y analizaron el mayor número de redes sociales presentes y observables mediante la información de sus dispositivos móviles. Entre las redes analizadas por el trabajo de Aharony se encuentran:

### Redes de proximidad física

- Red de proximidad a través de la detección de pares Bluetooth.
- Red de proximidad a través de puntos de acceso wi-fi similares.
- Red de proximidad a través de torres de celular (células) similares.
- Red de proximidad con la distancia predefinida por coordenadas de GPS.



"Network" ≡ "Graph"

**Figura 2.** Representación gráfica de un grafo.

### Redes sociales basadas en la actividad de sitios de redes sociales

- Red de contactos en Facebook.
- Red definida por las interacciones que se dan en Facebook (e.g. mensajes).
- Red definida por co-ubicación en Facebook (e.g. dos usuarios establecen un vínculo si ambos asisten al mismo evento, o hacen parte de un grupo específico).

### Redes auto-percibidas

- Red definida por los reportes de cercanía hechos por los mismos integrantes de la red.
- Red definida por respuestas similares de encuestas.

En la investigación de Aharony (Aharony, 2012) se hizo una comparación exhaustiva de los diferentes tipos de redes y su relación con diversos fenómenos sociales. Siendo el resultado más significativo para el presente proyecto, el análisis realizado sobre la influencia de los

distintos tipos de redes en la instalación de aplicaciones móviles. Dicho estudio, determinó que las redes más influyentes en este fenómeno son aquellas que consideran relaciones de cercanía física; contrario a la percepción común que dicta que las personas pertenecientes a las *redes sociales auto-percibidas o basadas en la actividad de sitios de redes sociales* tienen mayor influencia al momento de sugerir nuevas aplicaciones móviles.

## 2.2. Trabajos relacionados

Existen diferentes trabajos que abordan los núcleos temáticos del presente proyecto, en ellos se han presentado propuestas interesantes alrededor de diferentes aspectos involucrados en la recomendación de aplicaciones móviles. A continuación se presenta un análisis de los mismos, exponiendo su aporte y aplicación al momento de desarrollar este proyecto de grado. Para tal fin se ha identificado las siguientes áreas relevantes: Consumo de aplicaciones móviles, Filtrado Colaborativo basado en contexto y consideración del contexto social en sistemas de recomendación.

### 2.2.1. Consumo de aplicaciones móviles

Tal como fue mencionado en la introducción de esta monografía, numerosos estudios han demostrado la correlación existente entre el contexto y el consumo de aplicaciones. Por ejemplo, existen resultados que demuestran un incremento en el uso de apps en situaciones donde los usuarios realizan actividades como mirar televisión o están en su hogar. Por otro lado mientras se realiza actividades como leer un libro, estar en la escuela o en vehículos de transporte público, se presenta una fuerte tendencia a la abstinencia relacionada con el consumo de aplicaciones móviles (Böhmer, y otros, 2011) (Falaki, y otros, 2010) (Ipsos MediaCT - The Media, 2012).

En este sentido se destacan los primeros trabajos que analizaron las implicaciones prácticas de la consideración de eventos relacionados con el uso de dispositivos móviles y necesidades del usuario en una situación determinada. Por ejemplo, Church y Smyth (Church, y otros, 2008) al analizar las necesidades del usuario en un entorno móvil, concluyeron que el contexto (definido para ellos como localización y tiempo), es un factor importante para tener en cuenta la búsqueda web en entornos móviles. Así también Verkasalo (Verkasalo, 2009) demostró que el uso de ciertos tipos de servicios móviles depende del contexto actual del usuario.

Con relación a los trabajos mencionados, a finales de la primera década del siglo XXI, se generó un gran interés por conocer más sobre el contexto del usuario y los diferentes fenómenos de consumo que se ejecutan en él; motivándose la investigación en *captura de información contextual en entornos móviles*. Es en esta área, donde Froehlich y otros (Froehlich, y otros, 2007) presentaron un sistema de recolección de datos reales de uso de teléfonos móviles, considerando un conjunto de más de 140 tipos de eventos. Dicho trabajo brindó un método para el muestreo de la experiencia móvil y describió un sistema para reunir esos datos desde el dispositivo del usuario. Así mismo, el objetivo de Demumieux y Losguin (Demumieux, y otros, 2005) fue recolectar datos objetivos sobre el uso e interacción del usuario con teléfonos móviles para incorporar dichos datos en su proceso de diseño. Su framework fue capaz de seguir las funcionalidades de alto nivel en dichos dispositivos, por ejemplo, llamar, jugar, y descargar programas externos. Sin embargo uno de los problemas relacionados con estos estudios se dio en las limitantes de tiempo (máximo 28 días) y número de usuarios empleados (máximo 16). En respuesta a este tipo de problemas, Falaki utilizó información obtenida de aplicaciones móviles en Android y Windows recolectada por

un tiempo entre 7 y 21 semanas, dependiendo del usuario enfocada en el uso de aplicaciones móviles, determinando una estrecha relación entre el consumo y características demográficas y contextuales (Falaki, y otros, 2010).

Sin embargo, el avance más significativo en materia de captura de datos relacionados con el consumo y comportamiento del usuario, fue realizado por Aharony y otros quienes presentaron la metodología “SocialFMRI” (Aharony, 2012), con la cual construyeron el dataset más completo sobre este tipo de información, el cual fue capturado por medio del estudio “Friends and Family” el cual fue llevado a cabo durante 15 meses con la participación de 140 individuos. A pesar del carácter privado del dataset generado, este trabajo publicó el código fuente de “Funf” (Aharony, y otros, 2011), la herramienta que implementa la metodología de “SocialFMRI” enfocada en el análisis del comportamiento del usuario con una granularidad muy fina. Por último, en el trabajo de Aharony se experimentó sobre la relación entre la descarga de nuevas aplicaciones y la influencia de los diferentes tipos de redes sociales en dicho fenómeno (Pan, y otros, 2011). Producto de tal estudio se determinó que en la práctica se puede predecir la instalación de una nueva aplicación a partir de la información contenida en las redes de cercanía física.

Los trabajos anteriores permiten afirmar que es posible la captura controlada tanto de información contextual y de consumo de aplicaciones móviles a partir del dispositivo del usuario. Esta ventaja ha permitido a diversos trabajos centrarse específicamente en el fenómeno de análisis del **consumo** de aplicaciones móviles y su utilización en sistemas de recomendación. La justificación para la utilización de este tipo de información, se da debido al problema de falta de interacción entre el usuario y los métodos explícitos de captura de rating para las aplicaciones móviles. Este fenómeno se refleja de manera general en las tiendas de aplicaciones móviles. Por ejemplo, al mirar las estadísticas entre descargas y apreciaciones en Google Play, se puede encontrar que el número de apreciaciones es menor al 10% de las descargas realizadas. Por tal razón en lugar de utilizar métodos de captura explícita sobre la apreciación del usuario, diferentes trabajos de investigación se han enfocado en inferir dicha apreciación a partir del análisis de la información de consumo del usuario.

De esta manera, uno de los primeros trabajos que utilizó esta aproximación es el de Woerndl (Woerndl, 2007), el cual relacionó directamente la información contextual del usuario (básicamente localización) con el fenómeno de instalación de aplicaciones móviles. Posteriormente el trabajo de Girardello (Girardello, y otros, 2010) utilizó además de la información de instalación, la de desinstalación de aplicaciones, para el cálculo de la apreciación del usuario. Sin embargo, el trabajo de Yan y Chen: AppJoy (Yan, y otros, 2011) mostró que la información sobre instalación y desinstalación no es un indicio suficiente para determinar la apreciación del usuario sobre una aplicación. En dicho trabajo Yan y Chen utilizaron una aproximación basada en el modelo de análisis de mercado RFD<sup>13</sup>, el cual tiene en cuenta el carácter reciente del uso de una aplicación, su frecuencia y periodo, para determinar que tanto el usuario “gusta” de una aplicación. En este sentido para determinar dicha información sobre el consumo, Böhmer presentó un modelo para el ciclo de vida de las aplicaciones móviles, detectando 6 eventos y 4 estados que son de suma utilidad para la inferencia de este tipo de información (Böhmer, y otros, 2011). Así, para el conocimiento del presente proyecto, las aproximaciones presentadas por Yan y Chen, en conjunto con el trabajo de Bohmer, son las más avanzadas en el campo de la inferencia de calificaciones basadas en información del consumo de aplicaciones. Sin embargo y a pesar de sus

---

<sup>13</sup> Por las siglas en Inglés de ‘Recency, Frequency and Duration’

ventajas no han sido probadas teniendo en cuenta consideraciones contextuales para el cálculo de dichas apreciaciones

### 2.2.2. Filtrado Colaborativo Basado en Contexto

En esta área la principal premisa dicta que la recomendación de aplicaciones móviles, depende en gran medida del contexto actual del solicitante, al igual que las apreciaciones emitidas dependen del contexto en el cual fueron realizadas. Adoptando este postulado, al interior de la bibliografía han sido desarrollados diversos trabajos de investigación, entre los cuales se destacan los siguientes:

**Capturing, Analyzing and Utilizing Context-Based Information about User Activities on Smartphones:** En el trabajo de Woerndl y Schulze (Woemdl, y otros, 2011) se realiza un modelado de un usuario móvil teniendo en cuenta la información contextual del mismo. Este proceso se realiza en tres pasos: el primero es el diseño de un FrameWork que registra todos los sensores hardware y software de un Smartphone; entre ellos se registra la localización, llamadas telefónicas, redes, mensajes y multimedia. En el segundo, se analiza la información recolectada para identificar las actividades del usuario, principalmente a partir de su localización, y en el tercer paso se adapta el acceso a la información según el modelo de usuario.

**Context-Based Splitting of Item Ratings in Collaborative Filtering:** En el trabajo de Baltrunas y Ricci (Ricci, y otros, 2009) se analiza la influencia de las condiciones contextuales al momento de calificar un ítem, de esta manera se introduce una nueva técnica de filtrado colaborativo llamada Ítem Splitting, la cual asume que ciertos ítems deben tener diferentes ratings en diferentes contextos. Para esto se generan dos ítems virtuales a partir del ítem original de acuerdo a la variable del contexto y al nivel de influencia que esta tenga sobre la calificación del mismo. En este orden de ideas se evaluó la técnica mencionada con múltiples tipos de filtrado colaborativo, como el basado en usuarios, basado en modelo y una recomendación no personalizada, demostrando que existe una disminución considerable en el índice de MAE (Mean Absolute Error) a medida que las variables contextuales son más influyentes en la percepción que el usuario tiene sobre los ítems.

**Context-Aware Collaborative Filtering System: Predicting the User's Preferences in Ubiquitous Computing:** En el trabajo de Chen (Chen, 2005) se propone la implementación de un sistema capaz de predecir las preferencias del usuario de acuerdo a experiencias pasadas y al contexto en el que se encuentre. En este sentido las preferencias de cualquier usuario en su contexto actual, pueden ser predefinidas por lo que otros usuarios han hecho en contextos similares. Para esto, se implementa un algoritmo de FC modificado, que incluye una nueva variable que indica la similitud que hay entre un contexto y otro.

**Climbing the app wall: enabling mobile app discovery through context-aware recommendations:** En el trabajo de Karatzoglou (Karatzoglou, y otros, 2012) se presenta a Frappe: Un sistema basado en algoritmos de filtrado colaborativo que considera el contexto del usuario junto con la frecuencia de uso de cada aplicación. Sin embargo, su alcance se limita a considerar únicamente la frecuencia de uso de cada aplicación, relevando a un segundo plano el concepto de consumo propuesto en AppJoy (Yan, y otros, 2011).

### 2.2.3. Consideración del contexto social en SR.

Como fue expuesto anteriormente, existen varias clases de información contextual: información sobre el contexto físico, cognitivo y social (Luis Rojas-Potosi, 2012). Así, en secciones anteriores fueron presentados trabajos relacionados con el análisis de las implicaciones de considerar el contexto físico y cognitivo dentro de las recomendaciones en entornos móviles. Por esta razón, en esta sección se presenta un análisis sobre trabajos que han considerado la utilización del contexto social del usuario en investigaciones relacionadas con sistemas de recomendación en entornos móviles y servicios:

**Composite Social Network for Predicting Mobile Apps Installation:** En el trabajo de Pan, Aharony y Pentland (Pan, y otros, 2011) se realiza un estudio que busca determinar qué tipo de red es la más influyente en la elección e instalación de aplicaciones móviles en dispositivos Android. Este estudio se realizó con una muestra de 55 participantes en un periodo de 5 meses para cuatro tipos de redes: la red generada por el registro de llamadas, red de proximidad con Bluetooth, red de amistad y la red de afiliación. En cada uno de los casos, se registró la instalación de aplicaciones de cada usuario ignorando las desinstalaciones, donde además, se generó una nueva red teniendo en cuenta nivel de correlación entre los usuarios de acuerdo a las aplicaciones que estos instalan. Finalmente, como un resultado de este trabajo se afirma que el tipo de red más influyente en la instalación de aplicaciones móviles, es la red de proximidad con Bluetooth. Este trabajo fue fundamentado por un proyecto de investigación doctoral denominado “Social fMRI: Measuring and Designing Social Mechanisms using Mobile Phones” desarrollado por Nadav Aharony (Aharony, 2012). Dicho proyecto presenta a “*Social Funcional Mechanism-design and Relationship imaging*” o “*Social fMRI*”: un enfoque que mejora metodologías computacionales relacionadas con las ciencias sociales, entrelazando estrategias de recolección de información con intervenciones experimentales. Las principales contribuciones de este trabajo son: i) Una metodología capaz de combinar enfoques de recolección de información, con intervenciones experimentales en grupos de personas (lo menos intrusivas posible).ii) Un *framework* que implementa la metodología mencionada. iii) Un *dataset* armado a través de la información recolectada por su sistema, el cual hasta la fecha, en uno de los más grandes de su género. iv) Un amplio conjunto de hallazgos experimentales que contribuyen al entendimiento de importantes preguntas de investigación en ciencias sociales computacionales.

**Developing Compelling Social-Enable Applications with context-based Social Interaction Analysis:** En el trabajo de Skraba, Beauvais y Maaradji (Skraba, y otros, 2009) se plantea el uso de las redes sociales para asistir a los usuarios en sus necesidades de comunicación, esto se realiza mediante un sistema de análisis de tres capas denominado SIAFrameWork (Social InteractionAnalysis), donde se infiere una red social mediante la recolección de información de fuentes como llamadas telefónicas, sms y correos electrónicos. En las tres capas del sistema, se realizan funciones de recolección de datos, análisis de los datos almacenados para ponderar cada interacción social y presentación de la red social en forma de un grafo, el cual se utiliza para el desarrollo de aplicaciones sociales tales como Social Helper, cuyo propósito es mostrar los contactos del usuario más cercanos en las últimas 24h en casos de emergencia.

**Use Social Network Information to enhance collaborative filtering performance:** En el trabajo de Liu y Lee (Liu, y otros, 2010), busca mejorar la precisión de las recomendaciones que provee, a través de la detección de la mejor vecindad del usuario. Para esto, se experimentó con 4 diferentes tipos de vecindades: (i) vecinos más cercanos, que corresponden a los usuarios que por lo menos tienen una calificación en común, (ii) red

social o usuarios que mantienen una relación entre sí, (iii) combinación entre las dos vecindades previas y (iv) especialización de los vecinos más cercanos teniendo en cuenta el los mensajes que el usuario ha enviado. Los resultados de este trabajo demuestran que la vecindad más adecuada para el cálculo de recomendaciones en sistemas de FC es aquella que combina a los vecinos más cercanos y la red social.

### 2.3. Brechas existentes

A partir del análisis de los anteriores trabajos se determina que los avances y brechas existentes se dan en las siguientes áreas:

**Captura de información contextual y de consumo:** con relación a este tema se determinó que el trabajo más adelantado es el realizado por Aharony (Aharony, 2012). Sin embargo, a pesar de la riqueza de la información entregada y en virtud del objetivo de estudio de dicha investigación (comportamiento contextual del usuario con granularidad fina), se encontró que su herramienta fue diseñada para dispositivos dedicados, con elevadas capacidades de procesamiento y almacenamiento. Esta afirmación es fundamentada por la experimentación propia realizada por los integrantes de este proyecto de grado, quienes al desplegar “funF” en un dispositivo “Samsung galaxy S3”, observaron un desempeño aceptable, con un consumo de batería demasiado elevado, interrupción de tareas simultaneas y sobrecarga de la misma aplicación provocando interrupciones. Por tal razón se justifica el primer objetivo de investigación del presente proyecto *“Modelar la información contextual de consumo de aplicaciones en entornos Móviles”* ya que a pesar de trabajar con una técnica de filtrado colaborativo genérica; la consideración del contexto implica el desarrollo de técnicas de modelado y captura que permitan la recolección de los datos necesarios sin afectar la experiencia del usuario. A su vez se resalta que el presente proyecto en su etapa de pruebas, considera la instalación de una aplicación de captura de datos sobre los dispositivos propiedad de voluntarios; contrario al experimento de Aharony donde para efecto de pruebas, el proyecto de investigación dotó a sus participantes con 140 dispositivos Android para la captura de datos.

**Calculo de calificaciones basadas en consumo:** Como fue presentado en la sección dedicada al análisis de consumo, para el conocimiento del presente proyecto, las aproximaciones presentadas por Yan y Chen (Yan, y otros, 2011), en conjunto con el trabajo de Bohmer (Böhmer, y otros, 2011), son las más avanzadas en el campo de la inferencia de calificaciones basadas en información del consumo de aplicaciones. Sin embargo y a pesar de sus ventajas no han sido probadas teniendo en cuenta consideraciones contextuales para el cálculo de dichas apreciaciones.

**Definición de vecindad basada en cercanía social:** Después del análisis de los trabajos de filtrado colaborativo basado en contexto físico, cognitivo y social; se determinó que en el momento, para el conocimiento de este proyecto, no existe una aproximación que utilice redes sociales para la definición de una vecindad de usuarios en el dominio de la recomendación de aplicaciones móviles. Por tal razón este proyecto pretende incluir funciones para el cálculo de la vecindad basadas en similitud calculada a partir de la función de correlación de Pearson y en redes sociales que representen altos niveles de cercanía y confianza, que de acuerdo a los trabajos presentados, influyen significativamente en la instalación de aplicaciones móviles. El objetivo es comparar cuál de las funciones propuestas, presenta mejores resultados.

**Filtrado colaborativo de aplicaciones móviles basado en contexto físico y cognitivo:**

En cuanto a los contextos físico y cognitivo, se sugiere un modelo basado en dimensiones y estados que pretende superar en cierta medida la dispersión presente en modelos contextuales tradicionales; aspecto reforzado por el procesamiento de datos primitivos y algoritmos de agrupamiento para el manejo de datos continuos, con el objetivo de obtener información contextual más coherente.



## **Capítulo III, Modelo de información contextual y Calificación por análisis de Consumo**

El presente capítulo tiene como objetivo describir los tipos y fuentes de información empleadas para definir el modelo contextual propuesto y el cálculo de la calificación de las aplicaciones móviles en función de su consumo. De esta manera, esta sección contiene una descripción detallada de las fuentes de información empleadas, su clasificación de acuerdo al tipo de información recolectada y del modelo utilizado para su abstracción. Seguidamente se define el consumo a partir de un análisis detallado de las variables de frecuencia y duración de uso de una aplicación móvil, aplicando el modelo contextual propuesto y exponiendo finalmente el cálculo de las categorías relevantes para la recomendación del usuario en un contexto dado.

### **3.1. Fuentes de Información**

En el dominio de la computación web o móvil, existen diversas fuentes de información que describen al usuario y su situación, permitiendo generar perfiles contextuales que detallan su entorno, comportamiento y estado a través del tiempo. En este tipo de entornos, uno de los principales retos es la recolección de la información contextual, debido a la alta movilidad del usuario y la cantidad de datos que implican su cambio dinámico de situaciones.

Algunos trabajos como (Hua, 2005), en pro de superar dicho reto, utilizan una aproximación ubicua, instalando sensores en puntos fijos (por lo menos, en su campo de experimentación o trabajo) para la recolección de información contextual. Sin embargo, generar ambientes propicios para un monitoreo óptimo de información contextual utilizando este tipo de dispositivos, resulta sumamente costoso y complicado.

Gracias a las nuevas tendencias marcadas por el uso masivo de Smartphones y teniendo en cuenta que este tipo de dispositivos permite realizar un seguimiento constante del usuario gracias a su portabilidad; se puede prescindir de enfoques como el mencionado anteriormente, utilizando a los dispositivos móviles como una fuente importante de información del entorno del usuario. En este orden de ideas, se realiza una clasificación de la información contextual teniendo en cuenta su origen o fuente de información dentro del Smartphone: vía sensores, explícitamente por el usuario (interfaces), u obtenida a través del consumo de servicios web ejecutados en el mismo dispositivo (Figura 3.).

#### **Información provista por el Usuario**

Comprende toda la información que el usuario pueda proporcionar de forma explícita, como datos personales, localización relativa, estado de ánimo actual, etc. Para su recolección, es necesario brindar los medios adecuados que permitan su ingreso de forma cómoda, rápida e intuitiva; entre los cuales pueden encontrarse formularios, interfaces gráficas, entre otros.

#### **Información asociada al dispositivo móvil**

Corresponde a la información que puede ser abstraída de los sensores del Smartphone. Para la recolección de este tipo de información, no se necesita la participación directa del

usuario, por el contrario, se puede detectar de manera implícita y transparente. Para propósitos de este proyecto se utiliza Teléfonos inteligentes con el sistema operativo Android, debido a su popularidad y fácil desarrollo en esta plataforma.

### Información obtenida a través de Servicios Web

Información obtenida a través del consumo de servicios web relacionados con información contextual del usuario, como servicios de telemetría y variables ambientales. Dependiendo del tipo de servicio, en algunos casos, se necesitan parámetros de entrada como la localización del usuario.

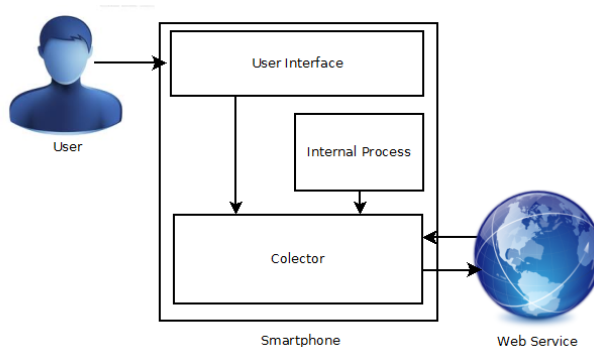


Figura 3. Fuentes de información.

## 3.2. Tipos de información

A partir de las fuentes citadas anteriormente, se clasifica la información que el sistema recolecta en tres grupos o categorías, i) información contextual, ii) información de dispositivo e iii) información de aplicaciones móviles.

### 3.2.1. Información Contextual

Corresponde a toda información que describa la situación contextual del usuario, es decir, su entorno y su estado a través del tiempo. Esta se clasificada en tres categorías contextuales: contexto físico, contexto cognitivo y contexto social (Luis Rojas-Potosi, 2012) (Figura 4).

<b>Processed variables</b>		User movement, User position		Social ties
<b>Basic variables</b>	<b>Implicit</b>	Mobile sensors, Services Weather, Services Temperature		Contact List, Call list, SMS list
	<b>Explicit</b>	Relative location	Mood	
		<b>Physical Context</b>	<b>Cognitive Context</b>	<b>Social Context</b>

Figura 4. Categorías de información Contextual.

#### 3.2.1.1. Contexto Físico

Describe el estado de las variables del entorno físico del usuario como posición, movimiento, nivel de luminosidad, etc. (Baltrunas, 2011) (Aharony, 2012). Dichos estados pueden ser

detectados a través de los sensores<sup>14</sup> del Smartphone (Sensors Overview), servicios web o de forma explícita (información que proporciona el usuario respecto a su entorno).

**Sensores de entorno:** Sensores que miden diversos parámetros ambientales, entre los cuales se incluyen barómetros, fotómetros y termómetros (Sensors Overview) (Weather), representados en la Tabla 4.

**Tabla 4.** Sensores de Entorno en Dispositivos Android.

Sensor	Unidades	Descripción de Datos
Temperatura Ambiente	Grados Centígrados (C°)	Temperatura Ambiente
Luz	Lux (lx)	Iluminancia
Presión	hPa o mbar	Presión Ambiente
Humedad Relativa	Porcentaje (%)	Humedad Relativa del ambiente
Temperatura	Grados Centígrados (C°)	Temperatura del dispositivo

**Sensores de movimiento:** Sensores que miden las fuerzas de aceleración y las fuerzas de rotación en los tres ejes (x, y, z) (Sensors Overview) (Tabla 5).

**Tabla 5.** Sensores de Movimiento en Dispositivos Android.

Sensor	Unidades	Descripción de Datos
Acelerómetro	Metros sobre segundo al cuadrado (m/s <sup>2</sup> )	Fuerza de aceleración en los ejes X,Y y Z
Gravedad	Metros sobre segundo al cuadrado (m/s <sup>2</sup> )	Fuerza de gravedad en los ejes X,Y y Z
Giroscopio	Radianes sobre segundo (rad/s)	Tasa de rotación en los ejes X,Y y Z
Aceleración Lineal	Metros sobre segundo al cuadrado (m/s <sup>2</sup> )	Fuerza de aceleración en los ejes X,Y y Z
Vector de Rotación	Sin Unidad	Componente del Vector de Rotación a lo largo de los ejes X,Y y Z

<sup>14</sup> **Sensores:** Componentes capaces de detectar magnitudes físicas o químicas y transformarlas en variables eléctricas para su posterior recolección y análisis. Pueden clasificarse en sensores de entorno, sensores de movimiento y sensores de posición, de acuerdo a las capacidades de los Smartphones (Sensors Overview).

**Sensores de posición:** Sensores que miden la posición física del dispositivo, entre los cuales se incluyen magnetómetros y sensores de orientación (Tabla 6).

**Tabla 6.** Sensores de posición en el dispositivo Android.

Sensor	Unidades	Descripción de Datos
Campo Magnético	Micro Tesla ( $\mu T$ )	Fuerza del campo magnético en los ejes X,Y y Z
Orientación	Grados	Ángulo de inclinación en los ejes X, Y y Z
Proximidad	Centímetros (cm)	Distancia del dispositivo a un objeto (También puede expresarse en unidades binarias cerca-lejos)

**Servicios Web de información ambiental y atmosférica:** Servicios RSS que proporcionan información detallada sobre el ambiente del usuario según su localización global (Weather)(Tabla 7).

**Tabla 7.** Descripción del Servicio RSS de Yahoo Weather.

Tipo	Elemento	Unidades	Descripción
Viento	Sensación Térmica	Grados Centígrados (C°)	Sensación termina del viento
	Dirección	Grados	Dirección del viento
	Velocidad	Metros o kilómetros por hora (mph, kph)	Velocidad del viento
Atmosfera	Humedad	Porcentaje	Humedad del Ambiente
	Visibilidad	Kilómetros o millas (km, mi)	Distancia de visibilidad
	Presión	Presión en in o mb	Presión Atmosférica
Astronomía	Amanecer	Hora	Hora exacta de la entrada del sol
	Atardecer	Hora	Hora exacta de la salida del sol
Clima	Condición	Sin Unidad	Descripción textual de la condición climática
	Temperatura	Grados Centígrados (C°)	Temperatura ambiente

**Localización relativa:** Información brindada por el usuario para describir su localización relativa de forma explícita. Esta puede ser categorizada en localización de interiores o localización de exteriores.

- **Interiores:** Localización descrita dentro de una estructura o edificio cerrado, como puede ser sala, estudio, cuarto, cocina, comedor, etc.
- **Exteriores:** Localización que describe un lugar o una estructura, como puede ser parque, universidad, hospital, biblioteca, cafetería, restaurante, etc.

**Información procesada:** A partir de las variables anteriormente descritas y un complejo procesamiento de las mismas, es posible inferir mayor cantidad de información como puede ser la movilidad del usuario (Barkley, 2000) (Ruth E. Mayagoitia, 2002) y su posición, entre otros (Tabla 8).

**Tabla 8.** Información Procesada

Información	Fuente	Descripción
Podómetro	Acelerómetro	Determina si un usuario se encuentra en movimiento o no.
Posición	Orientación y Pantalla	Determina la posición del usuario (de pie, sentado, acostado), de acuerdo a la orientación del Smartphone y al estado de la pantalla

### 3.2.1.2. Contexto Cognitivo

Describe el estado cognitivo del usuario, entre las cuales se incluyen las preferencias, estados mentales y afinidades entre seres humanos. Sin embargo, para fines de este trabajo, solo se tienen en cuenta los estados de ánimo, debido a la gran correlación que existe entre esta variable y el consumo de aplicaciones (Falaki, y otros, 2010).

**Estado de Ánimo:** Estados causados por emociones (Cheng-Yu, 2004) (Mayer JD, 1991) que no se remiten necesariamente a causas específicas y que no se pueden controlar ni elegir (Mayer JD, 1991) (Tabla 9).

**Tabla 9.** Principales Tipos de Estado de Ánimo (Cheng-Yu, 2004).

Tipos de Estado de Animo	Ejemplo de emociones
Ira	Rabia, enojo , irritación

Miedo	Temor, timidez, nervios
Alegría	Felicidad, euforia, placer
Dolor	Tristeza, angustia, melancolía
Aceptación	Amor, afecto, tolerancia
Rechazo	Odio, aversión, aburrimiento
Sorpresa	Asombro, sorpresa, extrañeza
Expectativa	Interés, atención, calma

### 3.2.1.3. Contexto Social

Describe toda la información relacionada con la red social del usuario. Para fines de este trabajo se capturan tres redes sociales, la red de contactos, red de llamadas y la red de mensajes de texto (Tabla 10).

**Tabla 10.** Redes sociales empleadas.

Red Social	Descripción
Contactos	Lista de contactos en el Smartphone
Llamadas	Llamadas realizadas y recibidas en la cual no se graba la conversación solo se registra el origen y destino
SMS	SMS enviados y recibidos, no se registra el contenido solo el origen y destino

**Vínculos Sociales:** A través del análisis de las tres redes sociales, se genera una macro red social, la cual agrega el valor de los vínculos sociales de las redes descritas anteriormente. Una descripción detallada de la conformación de esta red se encuentra en el Capítulo 5.

### 3.2.2. Información de Dispositivo

Corresponde a toda la información que describa el estado y características del Smartphone, tanto en hardware como en Software (Tabla 11).

**Tabla 11.** Información del dispositivo móvil.

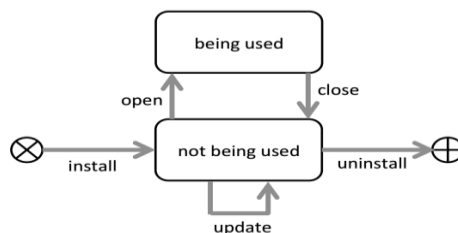
Tipo	Información	Descripción

Hardware	Pantalla	Características de tamaño y densidad de pantalla Tiempos en los que la pantalla se encuentra activa
	Batería	Consumo y nivel de batería
	Redes	Disponibilidad de Redes y periodos de conexión
	Fabricante	Empresa fabricante del dispositivo
	Modelo	Modelo del dispositivo dado por el fabricante
Software	Versión SO	Nombre de la versión del sistema operativo actual
	Versión SDK	Versión del SDK

### 3.2.3. Información de Aplicaciones Móviles

Información relacionada con el consumo y descripción de las aplicaciones móviles. Tal como se expuso en el estado del arte, para el análisis del consumo de aplicaciones móviles, se toma en cuenta el registro de eventos de su ciclo de vida (Böhmer, y otros, 2011) (Figura 5). Por otro lado, la información general de cada aplicación es extraída de la página de Google Play.

#### 3.2.3.1. Consumo de Aplicación Móvil



**Figura 5.** Ciclo de vida de una Aplicación Móvil (Böhmer, y otros, 2011).

Como se puede observar en la Figura 5, después de instalar una aplicación móvil, esta entra en un estado de **no uso** (*not being used*). Posteriormente al ejecutarse cambia de estado a **uso** (*being used*), de donde finalmente se puede regresar al estado de anterior o simplemente puede cerrarse. Para desinstalarla o actualizarla, necesariamente el estado debe ser de no uso (*not being used*).

En este orden de ideas, se detectan los eventos de instalación, desinstalación y actualización. Adicionalmente se registra cuando una aplicación se abre y se cierra, con el fin de detectar e inferir información como la *frecuencia* de uso (Número de veces que se ejecuta una aplicación) y su *duración* (*periodo* de tiempo en el que se usa la aplicación), datos necesarios para el análisis del consumo.

#### 3.2.3.2. Descripción de Aplicaciones Móviles

Esta información es necesaria al momento de entregar la recomendación al usuario solicitante. Ésta se obtiene a partir del id o nombre de paquete de la aplicación móvil en la página de Google Play (Google Play Store) (Tabla 12).

**Tabla 12.** Información de aplicaciones móviles obtenida en Google Play.

Información	Descripción
Nombre	Nombre de la aplicación como se muestra en el Google Play
Categoría	Categoría a la que pertenece la aplicación (juegos, productividad, entretenimiento, etc.)
Descripción	Descripción corta de la aplicación móvil
Rating	Calificación general de la aplicación móvil
Icono	Icono de la aplicación, como se muestra en el Google Play
Banner	Banner de la aplicación, como se muestra en el Google Play, no todas las aplicaciones contienen Banner

**NOTA:** La descripción, rating y el nombre de una aplicación en Google Play; pueden variar según la región y el idioma en el cual se realiza la consulta.

### 3.3. Modelo Contextual

Con el fin de identificar y analizar los datos capturados en cierto instante, se definen los términos *dimensión contextual* y *estado contextual*.

#### 3.3.1. Dimensión contextual

La *dimensión contextual* es definida como una variable que experimenta diferentes estados, donde  $V$  es el conjunto de  $n$  dimensiones consideradas en el sistema de recomendación de aplicaciones móviles propuesto.

$$V = \{v_i\}_{1 \leq i \leq n} \quad (1)$$

Por Ejemplo:

$$V = \{\text{Localización Relativa, Estado de ánimo, Clima, etc.}\}$$

#### 3.3.2. Estado Contextual

Corresponde al estado de una dimensión contextual en un instante de tiempo, donde  $S_i$  es el conjunto de  $m$  estados para la variable  $v_i$ .



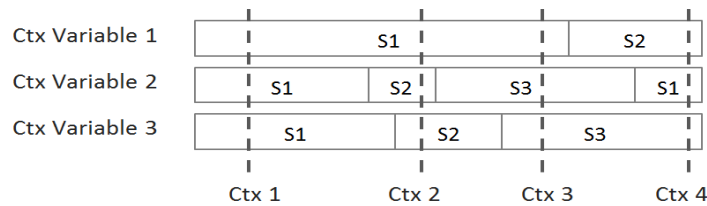
$$S_i = \{s_{i,j}\}_{1 \leq j \leq m} \quad (2)$$

Por Ejemplo para la *dimensión contextual*  $v_i = \text{Clima}$ :

$$S_i = \{\text{Soleado}, \text{lluvioso}, \dots, \text{parcialmente nublado}\}$$

Posteriormente el contexto del usuario es definido como un vector formado por los estados actuales de las  $n$  dimensiones contextuales, como se observa en la figura 6.

$$C = \{s_{i,*}\}_{1 \leq i \leq n} \quad (3)$$



**Figura 6.** Modelo Contextual del Usuario.

### 3.4. Calificación por análisis de Consumo

La calificación de un ítem en un sistema de recomendación permite conocer cuál es la apreciación del usuario en términos de su gusto o utilidad percibida. Consecuentemente, este factor puede ser expresado de muchas maneras tales como: la calificación binaria (like / dislike, utilizada en plataformas sociales como Facebook y YouTube); la calificación numérica (las cinco estrellas utilizadas por Amazon<sup>15</sup>) la calificación ordinal y la unitaria (Sami, y otros, 2009) (Ricci, y otros, 2011).

En el dominio de las aplicaciones móviles, generalmente se utilizan calificaciones numéricas empleando estrellas para comunicar la apreciación general. Este hecho puede observarse en plataformas o mercados como Google Play, AppStore, OVI store, entre otros. Sin embargo, tal como fue mencionado en el estado del arte, existen muchas dificultades al momento de obtener y recolectar dichas apreciaciones por parte del usuario, debido a que en la mayoría de los casos, este no está dispuesto a tomarse el trabajo de calificar las aplicaciones que consume (Yan, y otros, 2011), entorpeciendo el funcionamiento de este tipo de sistemas cuya precisión y eficacia disminuyen ante la ausencia de información necesaria.

Con el objetivo de solucionar este inconveniente, se propone inferir la calificación de una aplicación a partir del comportamiento del usuario, en este caso, asociado al consumo, que también provee información acerca de las tendencias o tipos de categorías que el usuario prefiere en determinados contextos (Yan, y otros, 2011) (Karatzoglou, y otros, 2012) (Woerndl, 2007).

<sup>15</sup> <http://www.amazon.com>

### 3.4.1. Análisis del Consumo

El consumo de aplicaciones móviles depende principalmente del tipo de aplicación y del tipo de usuario, es decir, existen aplicaciones cuyo uso está determinado según su función y utilidad en una situación específica. A su vez, existen usuarios que utilizan su Smartphone para propósitos particulares y en diferente medida, comparado con otros usuarios, ya sea por sus hábitos, necesidades o actividades.

Por otro lado, es posible que los gustos del usuario varíen a través del tiempo, debido a factores como las actualizaciones que algunas aplicaciones pueden sufrir generando problemas o mejoras en la experiencia inducida al usuario. Estos factores pueden conducir que el usuario modifique su consumo ya sea buscando alternativas diferentes o intensificando el uso de la aplicación actualizada según sea el caso. Esta variación, también puede presentarse por cambios en los hábitos/actividades del usuario o por el proceso natural de descubrimiento, en donde el usuario encuentra aplicaciones de su agrado o utilidad.

De acuerdo a lo anterior, con el objetivo de inferir la calificación de una aplicación móvil de forma implícita a partir del *consumo*, es necesario tener en cuenta la naturaleza y la **forma** del consumo actual del usuario, analizando factores como *frecuencia* de uso, *duración* de uso y la *fecha* de uso, como una medida de actualidad.

### 3.4.2. Tipos de Aplicaciones Móviles

Generalmente, las aplicaciones móviles son diseñadas para cumplir funciones específicas, orientadas a solventar necesidades particulares, ya sean de entretenimiento, sociales, información, etc. En este orden de ideas, su consumo está en función de su propósito o su naturaleza, reflejado en los factores de *frecuencia* y *duración* mencionados anteriormente. De esta forma, se determina los siguientes patrones de consumo:

#### **Aplicaciones de Uso Frecuente**

Son aplicaciones que se utilizan varias veces durante intervalos de tiempo relativamente cortos, es decir, la duración del uso de la aplicación es pequeña, pero se usa de manera muy repetitiva. Algunas aplicaciones de este tipo son aplicaciones de chat o mensajería, aplicaciones de lectura de RSS, etc.

#### **Aplicaciones de Uso Prolongado**

Son aplicaciones que se consumen durante tiempos prolongados pero pocas veces, es decir, el uso de la aplicación se realiza en tiempos largos pero no de manera repetitiva. Algunos ejemplos de este tipo pueden ser reproductores de video, editores de imágenes, juegos, lectura de libros digitales, etc.

Un elevado valor de alguno de estos dos factores, refleja un alto grado de consumo ya sea en el primer caso, donde el usuario usa la aplicación más veces respecto al segundo caso, o por el contrario, donde el consumo del usuario toma más tiempo en cada ejecución de la aplicación respecto al primer caso.

### 3.4.3. Calificación General por Consumo (Score<sup>16</sup>)

Para la inferencia de calificaciones (scores) a partir del *consumo*, se toman los eventos registrados del ciclo de vida de cada aplicación consumida, lo que permite inferir la información de *frecuencia* y de *duración* de acuerdo a las veces y el tiempo en que una aplicación permanece activa en pantalla.

$$Score(a) = d_a + f_a \quad (4)$$

Teniendo en cuenta la (Ecuación 4), La calificación para la aplicación  $a$  está dada por la suma entre  $da$  y  $fa$ , donde  $da$  representa la duración total en la que ha sido consumida (la suma de los tiempos de cada evento de ejecución) y  $fa$  representa el número total de veces que ha sido usada (número de eventos de cada ejecución). Los eventos del ciclo de vida de la aplicación que se toman para este cálculo, se filtran de acuerdo a la fecha de su ocurrencia descartando los que se hayan dado antes de los  $N$  días previos al momento de ejecutar este cálculo. Este filtro es el que determina los gustos o preferencias **actuales** del usuario, y a la vez evidencia los cambios del consumo en el tiempo.

Por otro lado, debido a la variación que puede existir en la forma como un usuario consume aplicaciones respecto a otros, es necesario normalizar la expresión mencionada anteriormente. De esta forma se tiene:

$$GeneralConsumptionScore(a) = \frac{d_a MAX(F) + f_a MAX(D)}{MAX(F) MAX(D)} \quad (5)$$

En la (Ecuación 5),  $F$  representa el vector de frecuencias de aplicaciones consumidas por el usuario y  $D$  el vector de duración de aplicaciones consumidas por el usuario. Normalizando respecto a los valores máximos dentro de los vectores mencionados, se obtendrá una calificación del consumo más coherente y acorde para cada usuario. De esta manera, se especifica el proceso de cálculo de la calificación general por consumo mediante el (Algoritmo 1). En dicho algoritmo se especifica que los datos necesarios corresponden al registro del consumo del usuario y su objetivo es encontrar la calificación general por consumo para todo el conjunto de aplicaciones previamente consumidas por él.

---

#### Algorithm 1: GeneralConsumptionScore

---

1. **INPUTS:** User's general consumption record  $R_u$  /\* Where each register on the general consumption record corresponds to  $\{time, app, appLifeCycleEvent\}$  \*/
2. **OUTPUTS:** set  $GCS_u$  /\* set of the generalConsumptionScore for the user  $u$  \*/
3. **BEGIN**
4. Set  $consumedApps \leftarrow LookUpAppsSet(R_u)$  /\* this function look up for non-repeated apps list from the consumption record \*/
5. Set  $D$  /\* Set of apps Duration \*/
6. Set  $F$  /\* apps Frequency \*/
7. Set  $GCS$  /\* Set of Apps' General Consumption Score \*/
8. **for** each App  $a_i$  **in**  $consumedApps$  **do**

---

<sup>16</sup> **Score:** Calificación inferida a partir del consumo, siendo la “calificación” un término diferente del “rating”.

```

9.   di ← appDuration(ai, Ru)
10.  fi ← appFrequency(ai, Ru)
11.  end for
12.  for each ai in consumedApps do
13.   gcsi ← GeneralConsumptionScore(ai) =  $\frac{da_iMAX(F)+fa_iMAX(D)}{MAX(F)MAX(D)}$ 
14.  end for
15.  Return GCSu
16.  END

```

La calificación general por consumo obtenida para cada aplicación, oscila entre 0 y 2, donde 2 representa a la aplicación más consumida, en este caso la aplicación de mayor frecuencia y de mayor duración, y cero corresponde a las aplicaciones que no han sido consumidas.

### 3.4.4. Calificación Contextual por Consumo.

Como fue expuesto en el estado del arte, el consumo de una aplicación móvil está fuertemente relacionado con la situación en la que tiene lugar. Sin embargo la ecuación presentada anteriormente considera una calificación basada en consumo independiente del contexto. Por tal razón, con el fin de tener en cuenta la relación consumo-contexto se propone la siguiente ecuación que determina la apreciación de una aplicación, basada en el análisis de su consumo en un estado contextual determinado.

$$Score(a, s) = \frac{d_{a,s}MAX(F_s) + f_{a,s}MAX(D_s)}{MAX(F_s)MAX(D_s)} \quad (6)$$

En la (Ecuación 6),  $Score(a, s)$  representa la calificación de la aplicación móvil  $a$  en el estado contextual  $s$ ,  $d_{a,s}$  y  $f_{a,s}$  corresponden a la duración y la frecuencia total de la aplicación  $a$  en el estado contextual  $s$ , respectivamente.  $D_s$  y  $F_s$  son vectores de duración y frecuencia de todas las aplicaciones consumidas en el estado  $s$ . La ecuación planteada tiene un rango de 0-2 como se ha explicado anteriormente, generando aplicaciones hipotéticas donde el  $score(a, s)$  más alto solo lo obtendrá la aplicación que tenga la máxima frecuencia y la máxima duración, la cual no siempre puede existir. Para solucionar este problema se realiza una escala del  $score(a, s)$  en cada estado contextual  $s$  (Ecuación 7).

$$Scorek(a, s) = \frac{score(a,s)}{MAX(score(A,s))} k \quad (7)$$

El rango de  $Scorek(a, s)$  iría de 0 a  $k$ , donde  $k$  representa el valor máximo de la calificación, por ejemplo una calificación de 0 a 5, la cual corresponde a la aplicación más consumida en el estado contextual  $s$ .  $A$  corresponde al vector de aplicaciones para el estado contextual  $s$ .

De acuerdo a lo planteado en el modelo contextual y en su respectiva expresión (Ecuación 3), se puede observar que el consumo del usuario en cada contexto depende del estado de sus dimensiones contextuales, en lugar de totalidad del contexto. Esto permite asignar un factor prioridad a cada dimensión contextual que representa el grado de influencia que esta tiene sobre el consumo de aplicaciones móviles. Consecuentemente el consumo total de una aplicación dado, se define de la siguiente forma:

$$CtxScore(a, c) = \sum_{s_{i,*} \in C} w_i Scorek(a, s_{i,*}) \quad (8)$$

En la (Ecuación 8),  $CtxScore(a, c)$  representa la calificación de la aplicación móvil  $a$  en el contexto  $c$ ,  $S_{i,*}$  es el estado actual de la variable contextual  $i$ , y  $w_i$  es la prioridad de cada dimensión contextual  $i$ , presente en el contexto  $c$ . La prioridad ( $w$ ) de cada variable contextual puede ser determinada manualmente por el usuario de acuerdo a la percepción que este tenga sobre la influencia que cada dimensión tenga sobre su consumo (puede presentarse el caso de que a pesar de los constantes cambios del clima, el consumo de aplicaciones móviles no se ve alterado, pero para el caso de cambios en la localización del usuario, sí).

---

**Algorithm 2: ContextualConsumptionScore**


---

1. **INPUTS:** User's contextual consumption record  $R_u$  /\* Where each register on the general consumption record corresponds to  $\langle time, app, appLifeCycleEvent, State\ of\ Contextual\ Dimention\ 1, \dots, State\ of\ Contextual\ Dimention\ I \rangle$  \*/, User's current context  $C_u$
2. **OUTPUTS:** set  $CCS_u$  /\* set of the Contextual Consumption Scores for the user  $u$  \*/
3. **BEGIN**
4. Set  $consumedApps \leftarrow LookUpAppsSet(R_u)$  /\* this function look up for non-repeated apps list from the consumption record\*/
5. Set  $k \leftarrow getTopScore()$  /\* this function return the  $k$  value\*/
6. Set  $SCSK$  /\* Set of Apps' Contextual State Consumption Score scaled with  $k$  \*/
7. Set  $CCS$  /\* Set of Apps' Contextual Consumption Score \*/
8. **for** each State  $s_i$  **in**  $C_u$  **do**
9.   Set  $D_i$  /\* Set of apps Duration for the State  $s_i$  \*/
10.   Set  $F_i$  /\* Set of apps Frequency for the State  $s_i$  \*/
11.   Set  $SCS_i$  /\* Set of Apps' Contextual State  $s_i$  Consumption Score \*/
12.   Set  $consumedContextualStateApps \leftarrow LookUpContextualStateAppsSet(s_i, R_u)$  /\* this function look up for non-repeated apps list from the contextual state consumption record\*/
13.   **for** each App  $a_j$  **in**  $consumedContextualStateApps$  **do**
14.      $d_{i,j} \leftarrow durationByContextualState(a_j, s_i, R_u)$  /\* this function get the duration for the app  $a_j$  in the contextual state  $s_i$  \*/
15.      $f_{i,j} \leftarrow frequencyByContextualState(a_j, s_i, R_u)$  /\* this function get the frequency for the app  $a_j$  in the contextual state  $s_i$  \*/
16.   **end for**
17.   **for** each App  $a_j$  **in**  $consumedContextualStateApps$  **do**
18.     set  $scs_{i,j} \leftarrow ContextualStateConsumptionScore(a_j, s_i) = \frac{d_{a_j, s_i} MAX(F_{s_i}) + f_{a_j, s_i} MAX(D_{s_i})}{MAX(F_{s_i}) MAX(D_{s_i})}$
19.   **end for**
20.   **for** each App  $a_j$  **in**  $consumedContextualStateApps$  **do**
21.      $scsk_{i,j} \leftarrow ContextualStateConsumptionScore(a_j, s_i) = \frac{scs_{i,j}}{MAX(sc s_i)} k$
22.   **end for**
23. **end for**
24. **for** each App  $a_i$  **in**  $consumedApps$  **do**
25.   Set  $S_i \leftarrow currentContextualStateByConsumedApp(a_i, C_u, R_u)$  /\* this function get an array of contextual states for the app  $a_i$ , where this application has been consumed \*/
26.    $csc_i \leftarrow ContextualConsumptionScore(a_i, S_i) = \sum_{j \in S_i} w_j scsk_{j,i}$
27. **end for**
28. **Return**  $CSC_u$
29. **END**

### 3.4.5. Detección de categoría por contexto

Tal como fue presentado en el estado del arte, diversos trabajos mostraron una estrecha relación entre características contextuales y categorías de aplicaciones consumidas (Falaki, y otros, 2010). En este sentido, esta sección está dedicada al análisis de las preferencias del usuario en términos de las categorías de aplicaciones que tiende a consumir en situaciones particulares. Para dicho análisis, se tiene en cuenta los pesos de las dimensiones contextuales  $w_{v_n}$  definidos anteriormente.

El proceso a seguir, en primera instancia, determina la categoría predominante por cada estado en el vector del contexto actual del usuario. Posteriormente, se procede a armar un nuevo vector cuyas componentes corresponden a los valores de  $w$  por categoría predominante determinada (Ecuación 9).

$$\text{Categories}(c) = \{w_1 \text{category}_a, w_2 \text{category}_b, w_3 \text{category}_a, \dots, w_i \text{category}_x\} \quad (9)$$

En la (Ecuación 9),  $\text{Categories}(c)$  es el vector de categorías relevantes por estado contextual en el contexto  $c$  y  $w_i$  es la prioridad de cada dimensión contextual.

Debido a que una categoría dada puede ser predominante en más de un estado, como se puede apreciar en el vector dado por la anterior expresión (en ese caso la categoría  $a$ , es predominante en los estados 1 y 3), se calcula la suma de pesos o prioridades  $w_i$  que tengan categorías comunes. Seguido, se calcula la prioridad total del contexto con la sumatoria de las prioridades de cada dimensión contextual (Ecuación 10), utilizadas en la (Ecuación 8).

$$W = \sum_{S_{i,*} \in c} w_i \quad (10)$$

Según la (Ecuación 10),  $W$  representa la prioridad total del contexto  $c$ ,  $w_i$  la prioridad de cada dimensión contextual  $S_{i,*}$  perteneciente al contexto  $c$ . Con el fin de encontrar la revelación de cada categoría frente a otras, se normaliza respecto a  $W$  (Ecuación 10-b).

$$\text{Categories}(c) = \left\{ \frac{\sum_{n=1}^{N1} w_{V_n, C1}}{W}, \frac{\sum_{n=1}^{N2} w_{V_n, C2}}{W}, \dots, \frac{\sum_{n=1}^{Nn} w_{V_n, Cm}}{W} \right\} \quad (10-b)$$

Cada componente o dimensión del vector de categorías, será un valor entre 0 y 1. El mismo representa la **relevancia** de una categoría frente a otras en el contexto dado y representa el indicador del número de aplicaciones que se deben recomendar por categoría respectivamente. De esta forma se define el número de aplicaciones recomendadas de la categoría predominante  $m$ , como el producto entre su relevancia y el número total de aplicaciones que se muestran (Ecuación 11).

$$NumAppsRec_{C_m} = NumAppsRecSocial. \left( \frac{\sum_{n=1}^{N1} W_{V_{n,C_m}}}{W} \right) \quad (11)$$

---

**Algorithm 3: ContextPredominantCategories**


---

1. **INPUTS:** Requester User  $u_i$ , Requester User current context  $Ctx_n \{S_{i,*}\}_{1 \leq i \leq n}$ , set ASR /\* ASR is the set of the score records from the target Users, the registers have the form {User  $u_i$ , App  $a_j$ , Score  $r_{i,j,S_j}$ } \*/
2. **OUTPUTS:** Set *predominantCategories* in context  $Ctx_n$ .
3. **BEGIN**
4. Set *UserPredominantCategoryTemp* /\* Set of the most consumed category in each contextual state  $S_i$  \*/
5. Set *UserPredominantCategories* /\* Set of the most consumed categories in the current context  $Ctx_n$  \*/
6. **for** each state  $S_i$  **in**  $Ctx_n$  **do**
7.     Set  $AppCount_{S_i} \leftarrow CountApplications(S_j)$  /\* This function detects the number of applications which have been consumed in state  $S_i$  \*/
8.     Set  $PredominantCategory_{S_j} \leftarrow highestNumberOfAppsCountedByCategory(AppCount_{S_j})$  /\*This function detects de maximum counting of Apps belonging to a specific category. This array has the form {category<sub>i</sub>,  $W_{V_{S_j}}$ } (position 1 = category's name, position 2 = category's weight), where  $W_{V_{S_j}}$  represents the weight assigned to the contextual dimension instantiated by the corresponding contextual State  $S_j$  \*/
9.     Set  $booleanFlag \leftarrow findCategory(PredominantCategory_{S_j}, UserPredominantCategories)$  /\* This function detects whether the  $PredominantCategory_{S_j}$  coincides with an another predominant category in other contextual state already saved in  $UserPredominantCategories$ , returning true if the current category already exists on the set \*/
10.     **if**  $booleanFlag = true$  **then**
11.         Set  $W_{total_{PredominantCategories_{S_j}}} \leftarrow PredominantCategory_{S_j}[2] + UserPredominantCategory$  /\*Operation which sums the weights  $W_{V_{S_j}}$  of equal predominant categories in different contextual states, in order to infer their relevance context \*/
12.         *UserPredominantCategories*  $\leftarrow removeCategory(PredominantCategory_{S_j})$  /\*Delete the previous weight corresponding to the current category  $PredominantCategory_{S_j}[1]$ , in order to replace it \*/
13.         *UserPredominantCategories*  $\leftarrow add(\{W_{total_{PredominantCategories_{S_j}}}, PredominantCategory_{S_j}[1]\})$  /\*Save the new Weight  $W_{total_{PredominantCategories_{S_j}}}$  as the relevance of the current category  $PredominantCategory_{S_j}[1]$  \*/
14.     **else**
15.         *UserPredominantCategories*  $\leftarrow add(PredominantCategory_{S_j})$
16.     **end if**
- 17.
18. **end for**
19. **return** *UserPredominantCategories*
20. **END**

### 3.5. Privacidad y Seguridad

La información recolectada puede ser muy sensible para el usuario, esta se almacena de manera local y centralizada con una encriptación basada en claves simétricas (Aharony, 2012). En la base de datos local se borran los datos en cada sincronización evitando el uso inapropiado de esta por otras aplicaciones y previniendo el uso inadecuado de espacio de memoria en el dispositivo. Además, se utiliza el número de celular único para identificar cada usuario, de igual manera se mantiene un nivel anonimato debido a la encriptación de la identidad y a que no se la vincula con un dato personal que identifique al usuario como tal. Para la recolección de los datos se utiliza un esquema de términos y condiciones donde se especifica que fin va a tener la información recolectada (Anexo E).

### 3.6. Resumen

En este capítulo se exploraron varias fuentes de información, que van desde los usuarios (información que este pueda brindar de forma explícita), hasta el Smartphone (información obtenida de forma automática, a través de sensores o consumo de servicios web). Teniendo en cuenta lo anterior, se clasificó información que se desea detectar en inf. Contextual, inf. Del dispositivo móvil e inf. De aplicaciones móviles, que corresponde a todo tipo de información que pueda ser útil para describir la situación contextual del usuario (contexto físico, contexto cognitivo y contexto social), al consumo y características no funcionales (metadatos) de cada aplicación.

Como uno de los principales aportes de este trabajo, se planteó un modelo contextual que incluye conceptos como dimensión contextual (variable que experimenta diferentes estados contextuales), estado contextual (corresponde al estado que una dimensión contextual puede experimentar en un instante determinado) y situación contextual (como un conjunto de estados contextuales por una dimensión). Esto con el fin de mejorar la percepción del contexto real del usuario y así mitigar los problemas de dispersión presentes en modelos contextuales cuyos estados tienden a ser continuos, cumpliendo con el primero objetivo específico definido en este trabajo.

Una vez definido el modelo contextual, se abordó el análisis del consumo de aplicaciones móviles, inferido a través de los eventos detectados del ciclo de vida de las mismas en determinados contextos, con el fin de generar una calificación automática. Para esto se clasificaron las aplicaciones móviles en aplicaciones de uso frecuente y de uso prolongado, según sea el patrón de consumo. Seguidamente se expresó el cálculo del consumo tomando en cuenta la frecuencia y la duración total de cada aplicación, esto, durante el periodo de tiempo más reciente para monitorear el cambio del consumo, por ende, de las preferencias de cada usuario. Como uno de los aportes de este trabajo, se aplicó el modelo contextual en la ecuación de consumo, con el fin de expresar el consumo en función del contexto tomando en cuenta cada estado contextual y el peso de cada dimensión. Por último se determinó la detección de la categoría o categorías relevantes de aplicaciones móviles dado un contexto con el fin de personalizar la recomendación sobre las preferencias del usuario representadas a través de categorías.



## Capítulo IV, Vecindad, Predicción y Recomendación

Como fue expuesto en el estado del arte, las ventajas del filtrado colaborativo sobre otros tipos de sistemas de recomendación, son significativas en aspectos como el rendimiento, versatilidad, adaptabilidad, simplicidad, entre otros. Lo anterior ha generado una gran aceptación de este tipo de enfoque, que gracias a sus beneficios, se ha convertido en una de las técnicas de recomendación más implementadas hoy en día (Ekstrand, y otros, 2011). Esta tendencia ha incentivado la investigación en esta área, buscando la optimización de la técnica mediante diversas aproximaciones que involucran desde aspectos sociales (Liu, y otros, 2010) (Golbeck, y otros, 2006), hasta aspectos contextuales (Baltrunas, 2011) (Hua, 2005) (Böhmer, y otros, 2010) (Chen, 2005) (Ricci, y otros, 2009).

Por definición: FC es una técnica de recomendación basada en comunidad (Ricci, y otros, 2011), que soporta sus recomendaciones en la agregación del registro de apreciaciones de un conjunto de usuarios similares al solicitante de la recomendación. En términos generales, el FC considera 4 fases en su proceso: i) captura/inferencia de calificaciones, ii) Identificación de la vecindad de usuarios, iii) predicción de la calificación de ítems y iv) recomendación. De esta manera, a lo largo de esta monografía, el (Capítulo 3) fue orientado al análisis y adaptación de técnicas enfocadas a la solución de problemas relacionados con la primera fase, en el dominio de la recomendación de aplicaciones móviles. Así, como parte final de la presente propuesta, en esta sección se exponen las aproximaciones para el cálculo de las vecindades propuestas y todo lo relacionado con la segunda fase del proceso de recomendación. Posteriormente se expondrán las técnicas y algoritmos relacionados con la tercera y cuarta fase del proceso de FC, empezando con una descripción de las generalidades sobre predicción y recomendación en FC, seguida por la presentación de los enfoques propuestos para la predicción y recomendación de aplicaciones móviles basada en información social y contextual del usuario.

Los enfoques presentados a continuación fueron diseñados con el fin de comprobar las dos hipótesis guías de la presente propuesta<sup>17</sup>, lo cual corresponde al **segundo objetivo específico** del presente proyecto: “*Adaptar técnicas de filtrado colaborativo basadas en la información contextual del usuario, relaciones sociales y el consumo de las aplicaciones*”. Por tal razón, a continuación se presentará tres enfoques los cuales representan: i) una aproximación al filtrado colaborativo tradicional, la cual será utilizada como línea base para la evaluación, ii) una aproximación basada en técnicas contextuales, cuyos resultados serán utilizados para ser comparada con la primera aproximación, con el fin de explorar la hipótesis H1 iii) una aproximación basada en la relevancia otorgada a expertos dentro de la red social del usuario según categoría, ésta al ser comparada con la primera aproximación permitirá el análisis de la hipótesis H2.

---

<sup>17</sup> **H1.** El uso de calificaciones inferidas teniendo en cuenta el consumo y el contexto físico del usuario generan una mejor calidad en los ARS.

**H2.** Otorgar una mayor relevancia a la apreciación de usuarios expertos dentro de la red social del usuario por cada categoría de aplicaciones, incrementa la calidad de las recomendaciones de los ARS basados en FC.

## 4.1. Generalidades sobre predicción y recomendación en FC

En esta sección se presenta las prácticas más comunes utilizadas en las fases de cálculo de vecindad, predicción y recomendación en filtrado colaborativo.

### 4.1.1. Vecindad

Uno de los aspectos críticos en los sistemas de FC *usuario-usuario*, es el mejoramiento del rendimiento y precisión a través de la optimización en el cálculo de la vecindad del usuario. En esta sección se presentan tres técnicas de cálculo de vecindad basadas en: la vecindad tradicional, información contextual e información social.

#### 4.1.1.1. Vecindad tradicional

Tal como fue expuesto en el estado del arte, los sistemas de FC *usuario-usuario* tradicionales definen a la vecindad como el conjunto de usuarios que guardan un alto grado de similitud con el solicitante de la recomendación. Esta se calcula: i) Seleccionando un conjunto de vecinos potenciales (Filtro para reducir el espacio de acción del cálculo de la *similitud*), y ii) Calculando el *factor de similitud* con cada uno de estos. Esta técnica de cálculo de vecindad considera un filtro preliminar antes del cómputo del factor de similitud entre usuarios, como se presenta a continuación.

#### Filtro por aplicaciones consumidas e instaladas

Para la estimación del conjunto de vecinos potenciales, algunos trabajos incluyendo el del grupo Lens (**Ekstrand, y otros, 2011**), catalogan a todos los usuarios registrados en el sistema como potenciales vecinos. Hecho que afecta la escalabilidad en el caso de contar con un número elevado de usuarios en registrados en el sistema. Con el fin de superar este inconveniente, se propone la implementación de un filtro de aplicaciones consumidas e instaladas el cual consiste en seleccionar los usuarios que más aplicaciones hayan consumido, por ende calificado, en común con el usuario en cuestión.

#### Factor de similitud

Este factor es el que determina el grado de similitud entre usuarios. Generalmente se calcula a través del *factor de correlación de Pearson*, debido a su incidencia positiva tanto en precisión como en eficiencia (**Ekstrand, y otros, 2011**).

#### Correlación de Pearson

Este método calcula la correlación estadística entre dos usuarios de acuerdo a los ratings comunes que existan entre ellos y al promedio de calificación de cada uno, siguiendo lo expuesto en la (Ecuación 12).

$$s(u, u') = \frac{\sum_{i \in I_u \cap I_{u'}} (r_{u,i} - \bar{r}_u)(r_{u',i} - \bar{r}_{u'})}{\sqrt{\sum_{i \in I_u \cap I_{u'}} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_u \cap I_{u'}} (r_{u',i} - \bar{r}_{u'})^2}} \quad (12)$$

En la ecuación especificada,  $r_{u,i}$  y  $r_{u',i}$  representan el rating (calificación inferida a partir del consumo en este caso) del ítem  $i$  del usuario y del vecino respectivamente; de igual forma, tanto  $\bar{r}_u$  como  $\bar{r}_{u'}$ , representan los promedios de calificaciones de cada uno.

Sin embargo, este factor de similitud tiende a ser impreciso cuándo existen pocos ítems comunes, por ejemplo, cuándo solo existe un ítem calificado entre dos usuarios, el valor de similitud resultante es alto. Esto puede ser solucionado con el uso de umbrales o rangos que garanticen que el número de ítems consumidos entre usuarios sea el adecuado (se sugiere que sea por encima de 50 (Ekstrand, y otros, 2011)). Esto implica problemas de arranque en frío, al no contar con la información suficiente para superar dichos umbrales. El proceso seguido para la identificación de una vecindad teniendo en cuenta los anteriores parámetros es descrito por el (Algoritmo 4).

---

**Algorithm 4: TraditionaNeighborhoodDetection**


---

1. **INPUTS:** Requester User  $u_r$ , set ASR /\* ASR is the set of the score records of the target Users, the registers have the form  $\{User\ u_i, App\ a_j, Score\ r_{i,j}\}$ \*/, double *threshold*, integer *topN*
2. **OUTPUTS:** Set *Neighborhood* /\* It is a set of pairs  $\{user, similarity\}$  \*/
3. **BEGIN**
4. Set  $U \leftarrow LookUpUserSet(ASR)$  /\* It is the set of all target users \*/
5. Set *UserSimilarity* /\*The set of tuples:  $\{requester\ user\ u_r, target\ user\ u_i, user's\ similarity\ userSimilarity_i\}$ \*/
6. set  $A_r \leftarrow lookUpApps(u_r, ASR)$  /\* this function looks up the consumed apps for the requester\*/
7. **for** each user  $u_i$  **in**  $U$  **do**
8.      $A_i \leftarrow lookUpApps(u_i, ASR)$
9.      $A \leftarrow A_i \cap A_r$  /\*conjunto de aplicaciones en común consumidas por el solicitante y el usuario candidato  $u_i$  \*/
10.      $userSimilarity_i \leftarrow s(u_r, u_i) = \frac{\sum_{a_j \in A} (r_{r,j} - \bar{r}_r)(r_{i,j} - \bar{r}_i)}{\sqrt{\sum_{a_j \in A} (r_{r,j} - \bar{r}_r)^2} \sqrt{\sum_{a_j \in A} (r_{i,j} - \bar{r}_i)^2}}$
11. **end for**
12. *UserSimilarity*  $\leftarrow similarityRanking(UserSimilarity)$
13. **for** each *userSimilarity<sub>i</sub>* **in** *UserSimilarity* **do**
14.     **if** *userSimilarity<sub>i</sub>* > *threshold* **then**
15.          $neighborhood_i \leftarrow userSimilarity_i$
16.     **end if**
17.     **if** *iteration* > *topN* **then**
18.         **Break for**
19.     **end if**
20. **end for**
21. **Return** *Neighborhood*
22. **END**

El proceso de identificación de vecindad culmina con la selección de los usuarios más similares al solicitante, mediante el uso de un valor de umbral (threshold) y un límite de miembros (topN).

#### 4.1.1.2. Vecindad contextual

Esta técnica sigue la dinámica expuesta anteriormente para la vecindad tradicional, diferenciándose por el hecho que para el cálculo de su vecindad se emplea un factor de similitud dependiente del contexto

## Filtro de aplicaciones consumidas e instaladas por contexto

Para la estimación del conjunto de vecinos potenciales en un contexto dado, se escogen a los usuarios que más ítems en común hayan consumido con el solicitante, en un contexto determinado.

### Factor de similitud

El *factor de similitud* utilizado en esta técnica, al igual que en la vecindad tradicional, el *factor de correlación de Pearson*, con la única diferencia de que las calificaciones o ratings que se utilizan, corresponden un contexto específico.

### Correlación de Pearson por contexto

Se representa a través de la misma expresión estadística dada anteriormente, pero con las modificaciones en cuanto a ratings y promedios sugeridas por el contexto.

$$s(u, u')_{Ctx_n} = \frac{\sum_{i \in I_u \cap I_{u'}} (r_{u,i,Ctx_n} - \bar{r}_{u,Ctx_n})(r_{u',i,Ctx_n} - \bar{r}_{u',Ctx_n})}{\sqrt{\sum_{i \in I_u \cap I_{u'}} (r_{u,i,Ctx_n} - \bar{r}_{u,Ctx_n})^2} \sqrt{\sum_{i \in I_u \cap I_{u'}} (r_{u',i,Ctx_n} - \bar{r}_{u',Ctx_n})^2}} \quad (13)$$

En la (Ecuación 13),  $r_{u,i,Ctx_n}$  y  $r_{u',i,Ctx_n}$  representan el rating (calificación inferida a partir del consumo en este caso) del ítem  $i$  del usuario y del vecino respectivamente, en el contexto  $Ctx_n$ ; de igual forma, tanto  $\bar{r}_{u,Ctx_n}$  como  $\bar{r}_{u',Ctx_n}$ , representan los promedios de calificaciones de cada uno.

---

### Algorithm 5: Contextual NeighborhoodDetection

---

1. **INPUTS:** Requester User  $u_r$ , Requester user Current Context  $Ctx_n$ , set ASR /\* ASR is the set of the score records of the target Users, User the registers have the form  $\{User u_i, App a_j, Score r_{i,j,Ctx_n}\}$  based on contextual scores (Algorithm 4) \*/, double *threshold*, integer *topN*
2. **OUTPUTS:** Set *Neighborhood* /\* It is a set of pairs  $\{user, similarity\}$  \*/
3. **BEGIN**
4. Set  $U \leftarrow \text{LookUpUserSet}(ASR)$  /\* It is the set of all target users \*/
5. Set *UserSimilarity* /\*The set of tuples:  $\{requester\ user\ u_r, target\ user\ u_i, user's\ similarity\ userSimilarity_i\}$ \*/
6. set  $A_r \leftarrow \text{lookUpApps}(u_r, ASR)$  /\* this function looks up the consumed apps for the requester in his current physical context\*/
7. **for** each user  $u_i$  **in**  $U$  **do**
8.      $A_i \leftarrow \text{lookUpApps}(u_i, ASR)$
9.      $A \leftarrow A_{i,Ctx_n} \cap A_{r,Ctx_n}$  /\*Set of common consumed apps by the candidate user  $u_i$  and the requesting user  $Ctx_n$  \*/
10.      $userSimilarity_i \leftarrow s(u_r, u_i)_{Ctx_n} = s(u_r, u_i) = \frac{\sum_{j \in A} (r_{r,j,Ctx_n} - \bar{r}_{r,Ctx_n})(r_{i,j,Ctx_n} - \bar{r}_{i,Ctx_n})}{\sqrt{\sum_{j \in A} (r_{r,j,Ctx_n} - \bar{r}_{r,Ctx_n})^2} \sqrt{\sum_{j \in A} (r_{i,j,Ctx_n} - \bar{r}_{i,Ctx_n})^2}}$
11. **end for**
12. *UserSimilarity*  $\leftarrow \text{similarityRanking}(UserSimilarity)$
13. **for** each  $userSimilarity_i$  **in** *UserSimilarity* **do**
14.     **if**  $userSimilarity_i > \text{threshold}$  **then**
15.          $neighborhood_i \leftarrow userSimilarity_i$

```

16. end if
17. if iteration > topN then
18.     Break for
19. end if
20. end for
21. Return Neighborhood
22. END

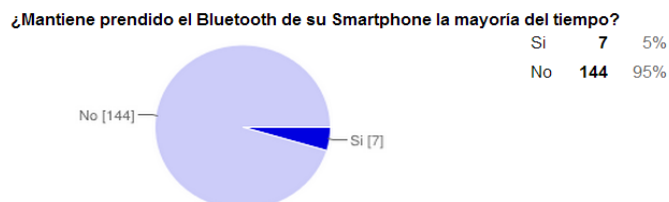
```

#### 4.1.1.3. Vecindad basada en redes sociales

Como fue mencionado en el estado del arte, hasta el momento no existe una aproximación en recomendación de aplicaciones móviles que tenga en cuenta la información del contexto social del usuario para la generación de una vecindad. Sin embargo, de acuerdo a los resultados expuestos en el trabajo de (Aharony, 2012), las redes sociales de interacciones *cara-cara* son las que mayor influencia tienen sobre la difusión y en la instalación de aplicaciones móviles.

Algunas aproximaciones a este tipo de redes, son las redes sociales de proximidad que brindan una significativa probabilidad de detectar interacciones que se dan de forma directa o *cara-cara*. Sin embargo, las redes que utilizan “finger prints” en wi-fi, coordenadas de GPS o celdas para determinar la proximidad entre dos usuarios, además de estar sujetas a procesamientos altos y a altos consumo de ancho de banda, no garantizarían que una interacción directa se haya dado entre estos dos (e.g. la distancia entre dos desconocidos conectados a una red wi-fi en un centro comercial, puede ser muy pequeña, pero por más cerca que estén, no necesariamente estos dos van a establecer una interacción). En este sentido, la opción más adecuada entre las redes mencionadas en el trabajo de Aharony (Aharony, 2012) es la red social basada en la sincronización entre pares Bluetooth, la cual garantiza que se ha establecido por lo menos una interacción directa entre dos personas.

Sin embargo, el entorno de funcionamiento del presente proyecto, sugiere que la probabilidad de contar con una red de usuarios totalmente sincronizada a través de esta tecnología, es mínima, tal y como se demuestra en una encuesta realizada para observar el uso del Bluetooth en la posible población de prueba. Como puede apreciarse en la Figura 7. Dicho estudio reveló que 1 de 20 usuarios, mantienen encendido este dispositivo la mayor parte del tiempo. La justificación de estos resultados está en razones como “prolongar la vida de la batería del Smartphone”, “Seguridad”, “por qué se considera que el Bluetooth es una tecnología poco práctica”, entre otras.



**Figura 7.** Encuesta sobre el uso de Bluetooth, dirigida a la comunidad de la Universidad de Cauca (publicada a través de las cuentas en "Facebook" y "Twitter" de esta institución), y al grupo de personas compuesto por los contactos en "Facebook" y seguidores en "Twitter", de los tesisas y asesor.

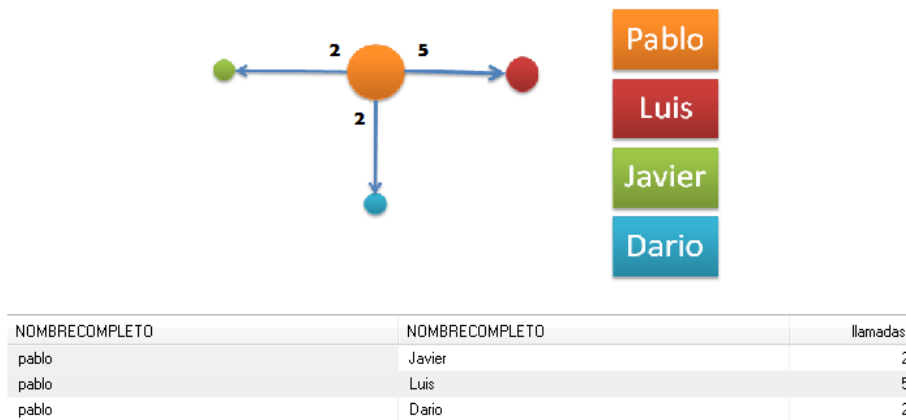
En este orden de ideas, es necesario abordar otras posibilidades para la selección de un tipo de red social cuya probabilidad de que por lo menos se haya tenido alguna interacción *cara-cara*, sea alta. De esta manera se opta por una red social basada en los contactos del celular, partiendo de la premisa que personas que se comunican constantemente tienen mayor probabilidad de encontrarse físicamente.

Sin embargo, en la realidad, uno de los problemas que se presentan es que algunos de los contactos que se guardan son irrelevantes y no tienen ningún tipo de relación con el usuario. Generalmente este tipo de contactos son almacenados de forma ocasional, y su presencia en la lista de contactos representa una relación estrecha. Es así como la detección de registros de llamadas y SMS pueden ayudar a superar estos inconvenientes, actuando como un filtro frente a los contactos irrelevantes. En este orden de ideas, dichos eventos son referenciados temporalmente, lo que permite filtrar la novedad de las interacciones que se registran, permitiendo definir que usuarios son los más cercanos en el presente o quienes dejan de serlo.

Teniendo en cuenta lo anterior, se define la vecindad basada en la red social de contactos del usuario, enriquecida con llamadas y SMS. Donde los vecinos más cercanos corresponden a los contactos con los cuales se ha establecido el mayor número de interacciones.

### Estructura de la red social de contactos

La estructura de la red social seleccionada, se describe tomando a los usuarios como los nodos, en donde se establece una relación unidireccional<sup>18</sup> (arista) cuándo un usuario tiene registrado a otro como contacto en su Smartphone; de la misma forma, el peso de una arista entre dos nodos, corresponde la suma del número de SMS y al número de llamadas que se han hecho entre estos, tal y como se muestra en la Figura 8.



**Figura 8.** Red social de llamadas (en el caso de que exista cierto número mensajes entre pablo y Darío, este será sumado al número de llamadas para definir el peso de la relación).

<sup>18</sup> **Relaciones unidireccionales:** se denominan unidireccionales, a aquellas relaciones en donde no hay reciprocidad por parte de uno de los individuos, en este caso, si el usuario A tiene en su lista de contactos al usuario B, no quiere decir que el usuario B tenga registrado al usuario A (Adamic, 2012).

### Factor de similitud

A diferencia de las otras vecindades descritas previamente, el factor de similitud entre usuarios utilizado en esta técnica, se calcula teniendo en cuenta las relaciones e interacciones presentes en la red social, tal y como se muestra en la (Ecuación 14).

$$s(u, u') = \text{Min} \left\{ 1, \left( 0,2 + 0,2 \cdot Fv + \frac{Nc_{u,u'} + Ns_{u,u'}}{Nc_{total} + Ns_{total}} \right) \right\} \quad (14)$$

Donde,  $Nc$  indica el número de llamadas,  $Ns$  el número de mensajes, el valor 0,2 representa el valor de cercanía por defecto si el usuario tiene a otro en su lista de contactos. Por otra parte el factor  $0,2 \cdot Fv$  indica la asignación de un peso extra si el solicitante tiene marcado como favorito al contacto en cuestión. Finalmente, la expresión establece como regla, escoger el valor mínimo entre 1 y el factor de similitud social, una vez este sea calculado. De manera similar a trabajos como el de Liu y Lee (Liu, y otros, 2010), lo que se busca son fuentes de información alternas, que aumenten la probabilidad de contar con más datos.

---

#### Algorithm 6: SocialNeighborhoodDetection

---

1. **INPUTS:** Requester User  $u_r$ , set USN /\* USN is the set of attributes from user's social network, including  $\{User\ u_i, Nc_{i,j}, Ns_{i,j}, Nc_{total}, Ns_{total}, Fv\}$  \*/
2. **OUTPUTS:** Set *Neighborhood* /\* It is a set of pairs  $\{user, similarity\}$  \*/
3. **BEGIN**
4. Set  $U \leftarrow \text{LookUpUserSet}(ASR)$  /\* It is the set of all target users \*/
5. Set *UserSimilarity* /\*The set of tuples:  $\{\text{requester user } u_r, \text{target user } u_i, \text{user's similarity } userSimilarity_i\}$  \*/
6. **for** each user  $u_i$  **in**  $U$  **do**
7.      $userSimilarity_i \leftarrow s(u_r, u_i) = \text{Min} \left\{ 1, \left( 0,2 + 0,2 \cdot Fv + \frac{Nc_{u_r, u_i} + Ns_{u_r, u_i}}{Nc_{total} + Ns_{total}} \right) \right\}$
8. **end for**
9. *UserSimilarity*  $\leftarrow \text{similarityRanking}(UserSimilarity)$
10. **for** each *userSimilarity<sub>i</sub>* **in** *UserSimilarity* **do**
11.      $neighborhood_i \leftarrow userSimilarity_i$
12. **end for**
13. **Return** *Neighborhood*
14. **END**

#### 4.1.1.4. Usuarios expertos

Una vez definida la vecindad basada en la red social de contactos, llamadas y SMS, se identifican a los usuarios expertos según las categorías relevantes en el contexto actual del solicitante. Los usuarios expertos son definidos como los contactos o las personas pertenecientes a la red social del usuario, que registran un consumo elevado de diversas aplicaciones de una categoría dada, tal como lo presenta la Figura 9. En otras palabras los expertos son los usuarios que más experiencia tienen en el consumo de aplicaciones en cierta categoría, y por lo tanto, más argumentos para sugerir aplicaciones. En este orden de ideas, con el fin de comprobar la hipótesis 2 (H2 – Capítulo 1, **Otorgar una mayor**

**relevancia a la apreciación de usuarios expertos por cada categoría de aplicaciones dentro de la red social del usuario, incrementa la calidad de las recomendaciones de los ARS basados en F)** se define que las apreciaciones asociadas a este tipo de usuarios tienen una relevancia proporcional con su “grado de experticia”, relacionado con su nivel de consumo de aplicaciones de una categoría determinada.



**Figura 9.** Usuarios expertos de la red social de contactos del usuario.

---

**Algorithm 7: ExpertUsersDetection**

---

1. **INPUTS:** set  $U$  /\* set of registered users in the system \*/, set  $AS$  /\* is the set of all registered applications in the system \*/, set  $ARS$  /\* is the set of the score records of the Users ( $U$ ), the registers have the form  $\{User\ u_i, App\ a_j, Score\ r_{i,j,S_j}\}$  \*/
2. **OUTPUTS:** Experts users per category /\* This set represents the detected experts per category, which are whose form cis  $\{category_i, \{user_1, user_2, user_3, \dots, user_n\}\}$ . \*/
3. **BEGIN**
4. Set  $Categories \leftarrow LookUpRegisteredCategories(AS)$  /\* Function that detects all categories analyzing the registered applications on the system \*/
5. Set  $ExpertsUsersByCategory$  /\*Collection that contains the set of expert users, indexed by categories\*/
6. **for** each  $category_j$  **in**  $Categories$  **do**
7.     **for** each user  $u_i$  **in**  $U$  **do**
8.          $CategoriesCountByUser_{u_i} \leftarrow countConsumedAppByCategory(u_i, category_j)$  /\*Function that returns a list composed by  $\{user_i, numberOfApps_{user_i}\}$  \*/
9.          $CategoriesCount_j \leftarrow add(CategoriesCountByUser_{u_i})$
10.     **end for**
11.     Set  $ExpertsUsersByCategory_j \leftarrow rankCounting(CategoriesCount_j)$  /\* This function returns the set of users sorted by the highest number of consumed apps in  $category_i$ , to the lowest. This set has the form  $\{user_1, user_2, user_3, \dots, user_n\}$  \*/
12.      $ExpertsUsersByCategory \leftarrow add(category_j, ExpertsUsersByCategory_j)$  /\* Function based on adding and indexing experts users by  $category_i$  \*/
13. **end for**
14. **Return**  $ExpertsUsersByCategory$
15. **END**



### 4.1.2. Predicción

Una de las implementaciones del filtrado colaborativo más sencillas para la predicción de ratings, es el *promedio ponderado* o “*Weighted averaging*”<sup>19</sup> el cual funciona muy bien en la práctica y es abierto respecto a la integración con enfoques sociales (Pennok, y otros, 2000). De esta forma, para generar las predicciones utilizando el *promedio ponderado*, es necesario calcular la vecindad  $N$  del usuario  $u$ , para lo cual se puede utilizar las aproximaciones abordadas en secciones anteriores. Una vez  $N$  ha sido calculada, el sistema combina los ratings de los usuarios pertenecientes a  $N$  del ítem  $i$  (ítem sobre el cuál se hará la predicción) y genera el rating del ítem  $i$  para el usuario  $u$  utilizando el *factor de similitud* (o  $s(u, u')$ ) adoptado. De esta manera el rating predicho para el ítem  $i$  y usuario  $u$  corresponde a:

$$P_{u,i} = \bar{r}_u + \frac{\sum_{u' \in N} s(u, u') (r_{u',i} - \bar{r}_{u'})}{\sum_{u' \in N} |s(u, u')|} \quad (15)$$

En la (Ecuación 15), los ponderados de los ratings ( $\bar{r}_u$  y  $\bar{r}_{u'}$ ) son los factores que compensan las diferencias que existen en las escalas de ratings que cada usuario tiende a utilizar, ya que algunos usuarios tienden a calificar con ratings más altos que otros.

### 4.1.3. Recomendación

La recomendación es la fase final del proceso de FC. Ésta es responsable de la definición de los ítems a recomendar al usuario solicitante, a partir de los resultados de calificaciones predichos anteriormente. Existen dos técnicas utilizadas en esta fase que pueden complementarse al momento de generar la lista final de ítems a recomendar: recomendación top- $n$  y recomendación por umbral (Sami, y otros, 2009). Ambas técnicas parten de la existencia de una lista de ítems ordenada de acuerdo al valor de rating predicho. La técnica top- $n$ , toma los  $n$  primeros ítems que presentan los ratings más altos para ser recomendados al usuario. Por otro lado la técnica basada en umbral, selecciona todos los ítems cuyo rating predicho sobrepase un valor  $t$ , definido por el diseñador del SR. Usualmente estas dos técnicas son combinadas de tal manera que la recomendación corresponde al conjunto de los  $n$  ítems con las mayores calificaciones, cuyo valor sobrepasa un umbral  $t$ .

### 4.1.4. Ejemplo de implementación (Ekstrand, y otros, 2011)

Para un mejor entendimiento de las generalidades con respecto a la predicción de ratings, se plantea el siguiente ejemplo. Considerando la matriz  $N$  (*matriz usuario vs ítem*), se desea calcular o predecir el rating del usuario  $C$  sobre el ítem “*Equilibrium*” ( $p_{C, equilibrium}$ ), utilizando el factor de correlación de Pearson y la predicción por promedio ponderado.

<sup>19</sup> En la literatura se encuentran aproximaciones más complejas, como las implementaciones de la función *regresión multi-variable* [Ref. 137 lens] sobre los usuarios de la vecindad o un *promedio no ponderado* [Ref 62 Lens]),

	<i>Batman Begins</i>	<i>Alice in Wonderland</i>	<i>Dumb and Dumber</i>	<i>Equilibrium</i>
User A	4	?	3	5
User B	?	5	4	?
User C	5	4	2	?
User D	2	4	?	3
User E	3	4	5	?

**Figura 10.** Matriz  $N$ , aplicación vs. Usuario.

Como puede apreciarse en la matriz  $N$ , únicamente los usuarios A y D han calificado dicho ítem; de esta manera estos se toman como la vecindad del usuario C y se calcula la similitud que cada uno mantiene con C a través del factor de correlación de Pearson (Ecuación 12).

$$s(C, A) = \frac{(4 - 4)(5 - 3,667) + (3 - 4)(2 - 3,667)}{\sqrt{(5 - 3,667)^2 + (2 - 3,667)^2} \sqrt{(4 - 4)^2 + (3 - 4)^2}} = 0,832$$

$$s(C, D) = \frac{(2 - 3)(5 - 3,667) + (4 - 3)(4 - 3,667)}{\sqrt{(5 - 3,667)^2 + (4 - 3,667)^2} \sqrt{(2 - 3)^2 + (4 - 3)^2}} = -0,515$$

Con los factores de similitud calculados, es posible aplicar la predicción por *promedios ponderados*, tal como fue expuesta en la (Ecuación 15).

$$p_{C, equilibrium} = \bar{r}_C + \frac{s(C, A)(r_{A, equilibrium} - \bar{r}_A) + s(C, D)(r_{D, equilibrium} - \bar{r}_D)}{|s(C, A)| + |s(C, D)|}$$

$$p_{C, equilibrium} = 3,667 + \frac{0,832(5 - 4) + (-0,515)(2 - 3)}{0,832 + 0,515|} = 4,667$$

Suponiendo que el umbral para la recomendación  $t$  es 4.0, se tiene que el ítem “equilibrium” es apto para su recomendación, al sobrepasar el valor de  $t$ . Nótese que la predicción de la calificación para el ítem “Equilibrium” estuvo soportada por los ratings brindados por dos usuarios. Por otro lado los demás ítems cuentan con la calificación de al menos 3 usuarios. Este hecho se vuelve crítico a medida que el banco de usuarios crece, debido a que la falta de calificaciones incide negativamente sobre la calidad de la predicción del rating del ítem candidato ser recomendado; siendo ésta una de las principales causas de la imprecisión en las recomendaciones que se proveen mediante FC. Este problema se acentúa aún más en sistemas de *FC usuario–usuario* basados en contexto, debido a que los registros de ratings son segmentados dependiendo de los estados de las dimensiones contextuales presentes al momento de la calificación. Ante este problema fueron presentadas las aproximaciones para el cálculo de vecindad relacionadas en las anteriores secciones, las cuales brindan un método para mitigar la ausencia de ratings en recomendación contextual y q a la vez permitan realizar cálculos sobre un conjunto manejable de valores que no afecten la eficiencia del sistema.

## 4.2. Propuesta

Como fue dicho anteriormente, con el fin de cumplir el objetivo general del presente proyecto y así comprobar las hipótesis H1 y H2, se propone **especializar el número de aplicaciones relevantes en el descubrimiento de las mismas**, proponiendo algoritmos que agreguen diversas técnicas para la inferencia de calificaciones y vecindades teniendo en cuenta factores sociales, contextuales y de consumo. Por último en la presente propuesta se define un algoritmo de construcción de recomendación que agrega los resultados de los tres algoritmos de predicción propuestos, el cual tiene como finalidad brindar una recomendación robusta y soportar la comparación de la eficacia de los algoritmos de predicción propuestos.

### 4.2.1. Predicción

A continuación se presentarán tres enfoques para la recomendación de aplicaciones móviles utilizando información social y contextual del usuario. Estos enfoques fueron propuestos para validar las hipótesis guía de este proyecto, y se basan en *filtrado colaborativo usuario-usuario de promedio ponderado* o *Weighted averaging*. Así, estos enfoques se definen como: **Tradicional**, **Contextual** y **Social-contextual** (siendo este, el enfoque principal). Cada una de ellos utiliza diferentes algoritmos “pre-filtrado” o algoritmos de adaptación que dependen de las fuentes de información que se tienen en cuenta para cada uno (procesamiento de la información de entrada o “input data” (Capítulo 2).

Una de las ventajas de esta “diversidad” en algoritmos es la posibilidad de una configuración “fallback” de los mismos, la cual permite utilizar un algoritmo que requiera menor información de entrada, ante la eventual falla de un algoritmo de orden superior que requiera mayor información. A continuación se dará una descripción detallada de los tres enfoques mencionados.

#### 4.2.1.1. Enfoque tradicional

El enfoque tradicional parte de las bases del filtrado colaborativo tradicional presentado en un ejemplo anterior. Como parámetros de entrada tiene:

- La información del usuario solicitante ( $u_r$ ).
- Una vecindad de usuarios (*neighborhood*) definida utilizando el (Algoritmo 4), orientado a la detección de vecinos similares basado en las calificaciones genéricas de ellos. Esta vecindad es detectada siguiendo un proceso el cual se encarga de filtrar el conjunto total de personas registradas en el sistema, con el objetivo de generar la vecindad del solicitante de acuerdo al número de aplicaciones que han sido consumidas en común (la intersección entre el conjunto de aplicaciones consumidas por el solicitante, con el conjunto de su vecino  $u$  De acuerdo a lo que sugieren trabajos como el de Liu y Lee (Liu, y otros, 2010), el tamaño máximo de la vecindad debe ser de 40 (entre 30 y 50 aproximadamente)
- El conjunto de las calificaciones generales de aplicaciones por usuario de acuerdo al consumo (ASR), las cuales son inferidas utilizando el (Algoritmo 1). Cada registro de este conjunto contiene la siguiente información  $\{User\ u_i, App\ a_j, Score\ r_{i,j}\}$ ; donde,  $u_i$  corresponde al usuario asociado a la calificación,  $a_j$  es la aplicación calificada y  $r_{i,j}$  es la calificación inferida a la aplicación  $a_j$  por el usuario  $u_i$ .
- Un número máximo de aplicaciones a recomendar ( $topN$ )

A partir de estos datos de entrada se ejecuta el (Algoritmo 8), con el fin de encontrar una lista de las aplicaciones móviles más relevantes.

---

**Algoritmo 8: TraditionalPrediction**

---

1. **INPUTS:** Requester  $u_r$ , Traditional Neighborhood  $neighborhood$  (Algorithm 4), set ASR /\* ASR is the set of the score records of the target Users, the registers have the form {User  $u_i$ , App  $a_j$ , Score  $r_{i,j}$ } \*/ ,integer  $topN$
  2. **OUTPUTS:** Set  $PredictedRatings$  /\* each register  $predictedRating$  of the list  $PredictedRatings$  has the form {Application  $a_i$ , Rating  $r_i$ } \*/
  3. **BEGIN**
  4. set  $A^r \leftarrow \text{lookUpApps}(u_r, ASR)$  /\* this function looks up the consumed apps for the requester\*/
  5. set  $A^N \leftarrow \text{lookUpApps}(neighborhood, ASR)$  /\* this is the set of the apps consumed by all the neighborhood \*/
  6.  $A \leftarrow A^N - A^r$  /\* This is the applications' set those have not been consumed by the requester \*/
  7. Set  $R$  /\* Defined as the array of predicted ratings \*/
  8. **for** each  $a_i \in A$  **do**
  9.  $U' \leftarrow \text{lookUpUsersFromNeighborhood}(neighborhood, ASR, a_i)$
  10.  $r_i \leftarrow P_{U',i} = \bar{r}_u + \frac{\sum_{u' \in U'} s(u_r, u') (r_{u',i} - \bar{r}_{u'})}{\sum_{u' \in N} |s(u_r, u')|}$
  11.  $PredictedRatings \leftarrow \text{add}(\{a_i, r_i\})$  /\* This function indexes the set of predicted ratings in each iteration by the corresponding app \*/
  - 12.
  13. **end for**
  14.  $PredictedRatings \leftarrow \text{rankRatings}(PredictedRatings)$  /\* This function ranks the apps by their predicted ratings, from higher to lower \*/
  15. **return**  $PredictedRatings \leftarrow \text{getTopNApps}(topN)$
  16. **END**
- 

El proceso descrito en el (Algoritmo 8) empieza por identificar las aplicaciones que la vecindad de usuarios han consumido pero que el solicitante no (líneas 4,5 y 6). Seguido se predice el rating  $r$  para cada una de las aplicaciones que no han sido consumidas por el solicitante, utilizando la ponderación expuesta en la sección anterior. Así, primero se identifica que usuarios han consumido la aplicación relacionada (línea 9), para a partir de sus apreciaciones predecir la calificación (línea 10).

#### 4.2.1.2. Enfoque contextual

El enfoque contextual parte de las bases del filtrado colaborativo presentado en un ejemplo anterior, junto con el modelo de contexto y calificaciones contextuales definidas en el capítulo 3. Como parámetros de entrada tiene:

- La información del usuario solicitante ( $u_r$ ),
- El contexto físico actual del solicitante ( $Ctx_n$ ).
- Vecindad de usuarios ( $neighborhood$ ) definida utilizando el (Algoritmo 5), orientado a la detección de vecinos similares basado en las aplicaciones consumidas en el contexto dado. Esta vecindad es detectada siguiendo un proceso el cual se encarga de filtrar el conjunto total de personas registradas en el sistema, con el objetivo de generar la

vecindad del solicitante de acuerdo al número de aplicaciones que han sido consumidas en común en el mismo contexto (la intersección entre el conjunto de aplicaciones consumidas por el solicitante, con el conjunto de su vecino  $u$ ). De acuerdo a lo que sugieren trabajos como el de Liu y Lee (Liu, y otros, 2010), el tamaño máximo de la vecindad debe ser de 40 (entre 30 y 50 aproximadamente).

- El conjunto de las calificaciones de las aplicaciones por usuario de acuerdo al consumo en cada estado contextual  $S_j$  ( $ASR_{ctx}$ ), las cuales son inferidas utilizando el (Algoritmo 2). Cada registro de este conjunto contiene la siguiente información  $\{User\ u_i, App\ a_j, Score\ r_{i,j,S_i}\}$ ; donde,  $u_i$  corresponde al usuario asociado a la calificación,  $a_j$  es la aplicación calificada y  $r_{i,j,S_i}$  es la calificación inferida a la aplicación  $a_j$  por el usuario  $u_i$  en el estado  $S_j$  perteneciente al contexto actual  $Ctx_n$ .
- Un número máximo de aplicaciones a recomendar ( $topN$ ).

A partir de estos datos de entrada se ejecuta el (Algoritmo 9), con el fin de encontrar una lista de las aplicaciones móviles más relevantes.

---

**Algoritmo 9: ContextualPrediction**

---

1. **INPUTS:** Requester  $u_r$ , Current user Context  $Ctx_n$ , Contextual Neighborhood  $neighborhood_{ctx}$  calculated though Algorithm 5 (Capítulo 5), set  $ASR_{ctx}$  /\*  $ASR_{ctx}$  is the set of the score records per contextual state from the target Users, the registers have the form  $\{User\ u_i, App\ a_j, Score\ r_{i,j,S_j}\}$  \*/ , integer  $topN$
  2. **OUTPUTS:** Set *PredictedRatings* /\* each register *predictedRating* of the list *PredictedRatings* has the form  $\{Application\ a_i, Rating\ r_{i,ctx_n}\}$  \*/
  3. **BEGIN**
  4. set  $A^r \leftarrow \text{lookUpApps}(u_r, ASR, Ctx_n)$  /\* this function looks up the consumed apps for the requester in his current context  $Ctx_n$  and calculates the contextual score for each one taking the score from each State  $S_j$  (Algorithm 2). This set has the form  $\{Application\ a, Contextual\ Score\ Sc_i^r\}$  \*/
  5. set  $A^N \leftarrow \text{lookUpApps}(neighborhood, ASR, Ctx_n)$  /\* this function looks up the set of the apps consumed by all the neighborhood in the given context, and calculates the contextual score for each one taking the score from each State  $S_j$  (Algorithm 2). This set has the form  $\{Application\ a, Contextual\ Score\ Sc_i^N\}$  \*/
  6.  $A \leftarrow A^N - A^r$  /\* This is the applications' set those have not been consumed by the requester \*/
  7. **for** each  $a_i \in A$  **do**
  8.  $U' \leftarrow \text{lookUpUsersFromNeighborhood}(neighborhood_{ctx}, ASR_{ctx}, a_i)$
  9.  $r_{i,ctx_n} \leftarrow P_{U',i} = \bar{r}_{u,ctx_n} + \frac{\sum_{u' \in U'} s(u_r, u')_{ctx_n} (r_{u',i,ctx_n} - \bar{r}_{u',ctx_n})}{\sum_{u' \in N} |s(u_r, u')_{ctx_n}|}$
  10. *PredictedRatings*  $\leftarrow \text{add}(\{a_i, r_{i,ctx_n}\})$  /\* This function indexes the set of predicted ratings in each iteration by the corresponding app\*/
  11. **end for**
  12. *PredictedRatings*  $\leftarrow \text{rankRatings}(\textit{PredictedRatings})$  /\* This function ranks the apps by their predicted ratings, from higher to lower \*/
  13. **return** *PredictedRatings*
  14. **END**
- 

El proceso descrito en el Algoritmo 8 empieza por identificar las aplicaciones que la vecindad de usuarios ha consumido en el contexto  $Ctx_n$  pero no el solicitante (líneas 4,5 y 6). Seguido se predice el rating  $r$  para cada una de las aplicaciones correspondientes, utilizando

la ponderación expuesta en la sección anterior. Así, primero se identifica que usuarios han consumido la aplicación relacionada (línea 9), para a partir de sus apreciaciones en el contexto dado predecir la calificación (línea 10).

#### 4.2.1.3. Enfoque Social-contextual

El enfoque social-contextual parte de las bases del filtrado colaborativo tradicional presentado en un ejemplo anterior, junto con la red social de contactos del usuario:

- La información del usuario solicitante ( $u_r$ ),
- El contexto físico actual del solicitante ( $Ctx_n$ )
- Conjunto de usuarios expertos por categoría (*ExpertsUsers*), definidos a través del algoritmo 7 (Capítulo 5).
- Una vecindad de usuarios (*neighborhood*) definida utilizando el (Algoritmo 6), orientado a la detección de vecinos similares basado en la red social de contactos enriquecida con llamadas y sms's.
- El conjunto de las calificaciones generales de aplicaciones por usuario de acuerdo al consumo (*ASR*), las cuales son inferidas utilizando el (Algoritmo 1). Cada registro de este conjunto contiene la siguiente información  $\{User\ u_i, App\ a_j, Score\ r_{i,j}\}$ ; donde,  $u_i$  corresponde al usuario asociado a la calificación,  $a_j$  es la aplicación calificada y  $r_{i,j}$  es la calificación inferida a la aplicación  $a_j$  por el usuario  $u_i$ .
- Un número máximo de aplicaciones a recomendar (*topN*).

A partir de estos datos de entrada se ejecuta el (Algoritmo 10), con el fin de encontrar una lista de las aplicaciones móviles más relevantes.

---

#### Algoritmo 10: Social-ContextualPrediction

---

1. **INPUTS:** Requester  $u_r$ , User's Current Context  $Ctx_n$ , Social Neighborhood *neighborhood* calculated through Algorithm 6 (Capítulo 5), Set of social network experts *ExpertsUsers* calculated through Algorithm 7, set ASR /\* ASR is the set of the score records from the target Users, the registers have the form  $\{User\ u_i, App\ a_j, Score\ r_{i,j,s_j}\}$  \*/, integer *topN*
2. **OUTPUTS:** Set *PredictedRatings* /\* each register *predictedRating* of the list *PredictedRatings* has the form  $\{Application\ a_i, Rating\ r_i\}$  \*/
3. **BEGIN**
4. set  $A^r \leftarrow \text{lookUpApps}(u_r, ASR)$  /\* this function looks up the consumed apps for the requester. This set has the form  $\{Application\ a, Score_i\}$  \*/
5. set *PredominantCategories*  $\leftarrow \text{detectMostConsumedCategoriesByContextualState}(u_i, Ctx_n)$  /\* function based on algorithm 3 (Capítulo 4), which detects the categories the user tends to consume in his current context \*/
6. set  $A^N \leftarrow \text{lookUpApps}(\textit{ExpertsUsers}, \textit{PredominantCategories}, ASR)$  /\* this function looks up the set of the apps consumed by all the neighborhood according to the defined categories *PredominantCategories* and their relevance (a higher relevance of a category, means a higher number of apps belonging to the same category). This set has the form  $\{Application\ a, Score_i\}$  \*/
7.  $A \leftarrow A^N - A^r$  /\* This is the applications' set those have not been consumed by the requester \*/
8. **for** each  $a_i \in A$  **do**
9.  $U^? \leftarrow \text{lookUpUsersFromNeighborhood}(\textit{neighborhood}_{social}, ASR, a_i)$
10.  $r_i \leftarrow P_{U^?,i} = \bar{r}_u + \frac{\sum_{u' \in U^?} s(u_r, u')_{social} (r_{u',i} - \bar{r}_{u'})}{\sum_{u' \in N} |s(u_r, u')_{social}|}$

11.  $PredictedRatings \leftarrow add(\{a_i, r_i\})$  /\* This function indexes the set of predicted ratings in each iteration by the corresponding app\*/
12. **end for**
13.  $PredictedRatings \leftarrow rankRatings(PredictedRatings)$  /\* This function ranks the apps by their predicted ratings, from higher to lower \*/
14. **return**  $PredictedRatings$
15. **END**

---

El proceso descrito en el (Algoritmo 10) empieza por identificar el tipo de aplicaciones o categorías que el usuario tiende a consumir en su contexto actual. Seguidamente, se identifican los contactos expertos de la red social del usuario, quienes son, los que se considera tienen mayor experiencia, por ende, un mejor criterio para sugerir nuevas aplicaciones pertenecientes a las categorías predominantes definidas (H2). Posteriormente, se seleccionan las aplicaciones pertenecientes a dichas categorías que los expertos han consumido (pero que el usuario no), cuya cantidad estará limitada por la relevancia de cada categoría (Capítulo 3). Finalmente se procede a ejecutar la predicción de los ratings de las aplicaciones descubiertas, utilizando la vecindad basada en la red social de contactos, llamadas y sms del usuario.

#### 4.2.2. Recomendación final

Al observar los algoritmos anteriores, se tiene que la salida de todos ellos corresponde a un conjunto de registros de la forma {Aplicación  $a_i$ , Calificación  $r_i$ }. A partir de dichos resultados, se debe definir la recomendación entregada al usuario, la cual debe tener un orden que represente el grado de relevancia predicho para el ítem a recomendar. Para concertar un criterio de organización de la recomendación final basada en las predicciones realizadas por los distintos algoritmos, es necesario considerar los fines académicos del presente proyecto, orientados a la comparación de los enfoques presentados.

De esta manera, las listas de ítems a organizar tienen la siguiente forma.

$$TraditionalList = [\{a^T_1, r^T_1\}, \dots, \{a^T_n, r^T_n\}] \quad (16)$$

$$ContextualList = [\{a^C_1, r^C_1\}, \dots, \{a^C_m, r^C_m\}] \quad (17)$$

$$SocialContextualList = [\{a^S_1, r^S_1\}, \dots, \{a^S_o, r^S_o\}] \quad (18)$$

De acuerdo a las (Ecuaciones 16, 17, 18),  $TraditionalList$  es la lista de aplicaciones dadas por el proceso tradicional;  $ContextualList$  es la lista de aplicaciones, resultante del proceso contextual para el solicitante  $u_r$  en el contexto  $c$ , y  $SocialContextualList$  es la lista de aplicaciones provistas por el proceso Social-Contextual para el usuario  $u$  en el contexto  $c$ .

Siguiendo los lineamientos propuestos por el objetivo general del proyecto, con el fin de brindar un óptimo descubrimiento especializando el número de aplicaciones relevantes, se seleccionan las 3 no repetidas con los ratings más altos en cada vector, contando con un número máximo de 9 aplicaciones en total, como lo presenta el (Algoritmo 11).

---

#### Algoritmo 11: RecommendationRanking

---

1. **INPUTS:** Set *TraditionalList*, Set *ContextualList*, Set *SocialContextualList* /\* Each one of the sets have registers {Application  $a_i$ , Score  $r_i$  } \*/,  
Integer *rNumber* /\* maximum number of applications for each technique \*/,  
Integer *depth* /\* maximum number of applications for each technique to consider \*/  
Integer *appPerPage* /\* Numer of applications per recommendation page \*/
2. **OUTPUTS:** Set *RankedAppsList*
3. **BEGIN**
4. Set A  $\leftarrow$  selectRecommendedApp(*SocialContextualList*, *depth*, *rNumber*) /\* Organizes the list of applications and selects rNumber applications out of depth, that randomly \*/
5. Set B  $\leftarrow$  selectRecommendedApp(*ContextualList*, *depth*, *rNumber*) /\* Organizes the list of applications and selects rNumber applications out of depth, that randomly \*/
6. Set C  $\leftarrow$  selectRecommendedApp(*TraditionalList*, *depth*, *rNumber*) /\* Organizes the list of applications and selects rNumber applications out of depth, that randomly
7. Set nPage  $\leftarrow$  (A.size + B.size + C.size) / *appPerPage*
8. **for** i=0; i < nPage; i++ **do**
9.     **if** A.size > 0
10.         rankedAppList<sub>j</sub>  $\leftarrow$  a<sub>j</sub>
11.     **else if** B.size > 0
12.         rankedAppList<sub>j</sub>  $\leftarrow$  b<sub>j+1</sub>
13.     **else if** C.size > 0
14.         rankedAppList<sub>j</sub>  $\leftarrow$  c<sub>j+1</sub>
15.     **else**
16.         **break for**
17.     **end if**
- 18.
19.     **if** B.size > 0
20.         rankedAppList<sub>j+1</sub>  $\leftarrow$  b<sub>j</sub>
21.     **else if** C.size > 0
22.         rankedAppList<sub>j+1</sub>  $\leftarrow$  c<sub>j+1</sub>
23.     **else if** A.size > 0
24.         rankedAppList<sub>j+1</sub>  $\leftarrow$  a<sub>j+2</sub>
25.     **else**
26.         **break for**
27.     **end if**
- 28.
29.     **if** C.size > 0
30.         rankedAppList<sub>j+2</sub>  $\leftarrow$  c<sub>j+1</sub>
31.     **else if** A.size > 0
32.         rankedAppList<sub>j+2</sub>  $\leftarrow$  a<sub>j+2</sub>
33.     **else if** B.size > 0
34.         rankedAppList<sub>j+2</sub>  $\leftarrow$  b<sub>j+2</sub>
35.     **else**
36.         **break for**
37.     **end if**
- 38.
39. **end for**
- 40.
41. **return** *RankedAppsList*
42. **END**



La ubicación de cada aplicación en la interfaz de recomendación, dependerá de su rating y de la prioridad de cada técnica de predicción, determinada por el anterior algoritmo. En este caso se da una prioridad alta a la técnica Social-Contextual, media a la técnica Contextual y baja a la técnica Tradicional. En este sentido, el orden en cómo se muestran las aplicaciones esta dado de la siguiente forma.

$$Pag1 = \{as1, ac2, at2\} \quad (19)$$

$$Pag2 = \{ac1, as2, at3\} \quad (20)$$

$$Pag3 = \{at1, as3, ac3\} \quad (21)$$

Como se puede observar, en cada interfaz/página, el primer elemento es la aplicación con más alto rating provista por cada técnica de predicción utilizada, debido a que es el primer elemento que nota el usuario; el resto de elementos se posicionan según la prioridad de la técnica de predicción.

### 4.3. Resumen

En este capítulo se exploraron varias técnicas para la inferencia de la similitud entre usuarios, que permitirán identificar varios tipos de vecindades al interior del sistema de FC propuesto entre las cuales se encuentran: vecindad tradicional, vecindad contextual y vecindad basada en red social de contactos enriquecida con interacciones tales como llamadas y sms's. Las dos primeras vecindades mencionadas se basan en los algoritmos que plantean otros trabajos de FC relacionados con la inferencia de la similitud entre usuarios a través del factor de correlación de Pearson, y el factor de correlación de Pearson extendido contextualmente. Sin embargo, como uno de los aportes de este trabajo y en pro de cumplir de cumplir gran porcentaje el segundo objetivo específico de este trabajo (***Adaptar técnicas de filtrado colaborativo basadas en la información contextual del usuario, relaciones sociales y el consumo de las aplicaciones***) concentrándose en la parte social, se considera la última vecindad como la más importante en el descubrimiento y sugerencia de aplicaciones móviles (Aharony, 2012). De la misma forma con el fin de personalizar aún más dicho descubrimiento, se definen a usuarios expertos dentro de la misma red social, cuya opinión y consumo puede significar un mayor grado de influencia debido a la experiencia que estos usuarios puedan tener.

Posteriormente fueron presentados los distintos enfoques para la predicción de calificaciones propuestos en este proyecto. Las técnicas y tipos de información relacionadas al proceso necesario para ejecutar cada uno de los enfoques propuestos, pueden ser resumidos en la Tabla 13.

**Tabla 13.** Elementos y componentes de cada técnica de recomendación.

Característica	Tradicional	Contextual	Social-contextual
Datos de entrada	Consumo del solicitante	Contexto del usuario, Consumo contextual del solicitante	Contexto del solicitante, Consumo del solicitante
Datos de soporte	Consumo de los demás usuarios	Consumo contextual de los demás usuarios	Consumo contextual de los demás usuarios, categoría de las aplicaciones móviles.
Tipo de	General	Contextual	General

<b>calificación individual</b>			
<b>Vecindad</b>	Tradicional	Contextual	Social por categoría contextual
<b>Escala de calificación</b>	Más consumidas	Más consumidas por contexto actual.	Más consumidas por expertos en la red social del solicitante

En la anterior tabla se puede observar que el enfoque tradicional es el que menor información de entrada necesita, en cambio los dos enfoques restantes utilizan una gran cantidad de información para predecir la calificación de sus recomendaciones. En cuanto al cálculo de la calificación individual de las aplicaciones, se puede observar que el enfoque contextual es el que menos datos posee para dicho proceso, en comparación con los enfoques tradicional y social-contextual. Por otro lado el tipo de vecindad elegido para cada uno de los enfoques varía en su grado de complejidad, siendo el más simple el adoptado por el enfoque tradicional, siendo los más complejos, en este aspecto, los enfoques: contextual y social-contextual; resaltando que entre más simple sea el proceso se tiende a descartar menos miembros de la vecindad, produciendo una recomendación más general y menos precisa.

## Capítulo V, Vanilla: Arquitectura e implementación

Uno de los principales resultados del trabajo de investigación realizado en este proyecto es el diseño e implementación de “Vanilla”, un sistema de recomendación de aplicaciones móviles desarrollado teniendo en cuenta las principales necesidades presentes en un entorno<sup>20</sup> móvil altamente concurrente<sup>21</sup>. En este orden de ideas, el sistema implementa varias soluciones a nivel de ingeniería enfocado en aspectos propios del dominio de aplicación, en un entorno real como: bajo consumo de ancho de banda, alta disponibilidad, rápido procesamiento de las solicitudes, usabilidad, bajo consumo de energía en los dispositivos móviles, entre otras.

En el presente capítulo se aborda la descripción detallada del trabajo realizado alrededor del desarrollo software del sistema de recomendación propuesto, que implementa la captura de información (Capítulo 3), análisis del consumo (Capítulo 3), selección de la vecindad (Capítulo 4) e implementación de los algoritmos de predicción y recomendación (Capítulo 4). La estructura del capítulo consta de una descripción por medio de vistas de casos de uso y arquitectura funcional, relacionando al final las tecnologías utilizadas para la implementación.

Para la implementación del prototipo se adoptó el primer macro componente de La Estructura para la Descripción del Sistema y el Modelado del Proceso de Desarrollo, del Modelo de Construcción de Soluciones (Serrano, 2005), como referencia metodológica para soportar el proceso de desarrollo, en el cual, se involucra la especificación de artefactos esenciales, para el entendimiento adecuado de la funcionalidad y el comportamiento esperados del sistema/solución (Modelo de Casos de Uso, Arquitectura de referencia, Modelo de Diseño (Anexo A), Modelo de Navegabilidad (Anexo A), Modelo de Despliegue y Plan de Pruebas).

El prototipo desarrollado consta de un esquema cliente–servidor, donde el cliente corresponde a una aplicación móvil encargada de recolectar la información del contexto del usuario y presentar las recomendaciones resultantes. Por otro lado, el servidor procesa la información obtenida y ejecuta los algoritmos y técnicas propuestas para generar las respectivas recomendaciones.

A continuación se describen los casos de uso del sistema, la arquitectura y las consideraciones en la implementación del prototipo.

### 5.1. Casos de Uso del Sistema

La definición de los casos de uso del sistema, parte de la identificación de los requisitos funcionales del mismo.

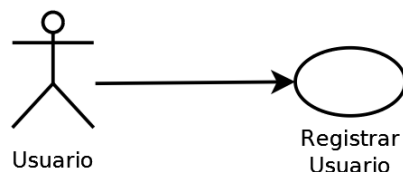
**R1. Recomendar aplicaciones móviles personalizadas a cada usuario en un contexto dado:** Se refiere a la capacidad que tiene el sistema de interpretar la información social, contextual y de consumo para brindar una recomendación personalizada a cada usuario.

#### Caso de Uso CU1: Registrar Usuario

---

<sup>20</sup> *Entorno:* Se utiliza entorno, para referirse al entorno de ejecución de la lógica de negocio

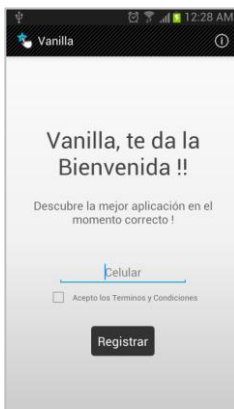
<sup>21</sup> *Concurrencia:* se refiere al suceso de varias conexiones o solicitudes en un mismo instante de tiempo.



**Diagrama 1.** Diagrama del caso de uso *Registrar usuario*.

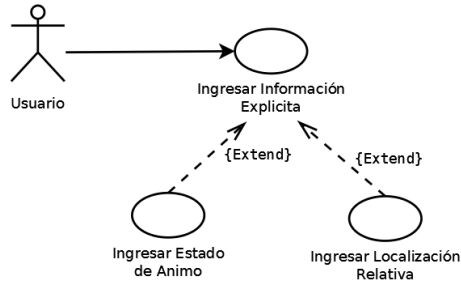
**Tabla 14.** Caso de uso *registrar usuario*.

Descripción	
<b>Actor</b>	Usuario
<b>Precondiciones</b>	Instalar Cliente del Sistema
<b>Propósito</b>	Registrar el usuario actual en el sistema de recomendación
<b>Resumen</b>	Vanilla registra a cada usuario con el fin de garantizar la identidad del móvil y la confiabilidad de la información que se captura (Requerimiento para la realización de recomendaciones).
<b>GUI</b>	Interfaz de registro de usuario (Figura 11).
Curso normal de Eventos CU1	
Acción de Actores	Respuesta del sistema
1. El usuario abre la Aplicación Cliente	2. Despliega la pantalla para el registro del usuario.
3. Se ingresa el número de teléfono para realizar el registro y se presiona el botón de registrar	4. Notifica que el registro ha sido exitoso
Curso Alterno de Eventos CU1	
Acción de Actores	Respuesta del sistema
	4. Error en conexión con el Servidor
	5. Notifica que el registro no ha sido exitoso



**Figura 11.** Registro de usuario

**Caso de Uso CU2: Ingresar Información Explícita**



**Diagrama 2.** Diagrama caso de uso *ingresar información explícita*.

**Tabla 15.** Caso de uso ingresar información explícita.

Descripción	
<b>Actor</b>	Usuario
<b>Precondiciones</b>	Agregar Widget de aplicación en la pantalla de inicio (Figura 12 (a)).
<b>Propósito</b>	Ingresar información explícita por parte del usuario.
<b>Resumen</b>	El usuario proporciona la información que necesariamente se tiene que capturar explícitamente para las dimensiones contextuales como la <i>localización relativa</i> y <i>Estado de ánimo</i> .
<b>GUI</b>	Widget (Figura 12 (a)) y selección del ingreso de los estados de la dimensión contextual (Figura 12 (b) (c)).
Curso normal de Eventos CU2	
Acción de Actores	Respuesta del sistema
1. El usuario selecciona el icono en el Widget de la información a ingresar.	2. Despliega las opciones correspondientes a la Dimensión contextual seleccionada.
3. Selecciona el estado actual correspondiente a la dimensión contextual	4. Almacena la información en la base de datos local
Curso Alternativo de Eventos CU2	
Acción de Actores	Respuesta del sistema
	4. Base de datos local no existente
	5. Crea la base de datos y almacena la información en la base de datos local.

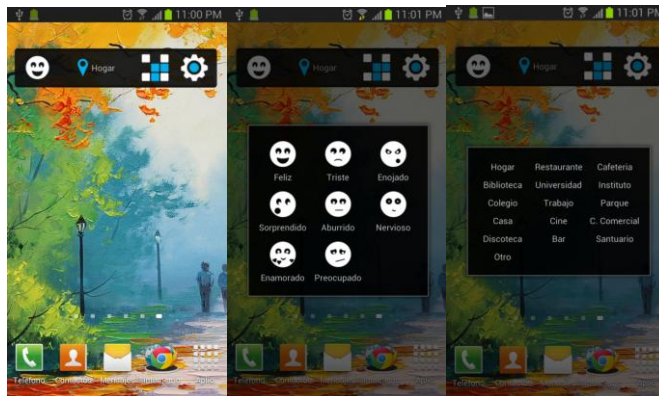


Figura 12. (a) Widget de cliente, (b) Ingreso del estado de ánimo (c) Ingreso de la localización relativa.

### Caso de Uso CU3: Ingresar Información Explícita

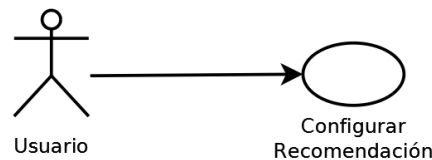


Diagrama 3. Diagrama caso de uso *Ingresar información explícita*.

Tabla 16. Caso de uso *Ingresar información explícita*.

Descripción CU3	
<b>Actor</b>	Usuario
<b>Precondiciones</b>	Abrir la ventana de configuración (Figura 12 (a)).
<b>Propósito</b>	Establecer la prioridad de cada dimensión contextual.
<b>Resumen</b>	En este caso de uso se proporciona la descripción de la configuración de la prioridad de las dimensiones contextuales, con el fin de que la recomendación se ajuste a las preferencias del usuario.
<b>GUI</b>	Interfaz de configuración de recomendación (Figura 13).
Curso normal de Eventos CU3	
Acción de Actores	Respuesta del sistema
1. El usuario desplaza la barra para fijar la prioridad al a dimensión contextual deseada.	2. Establece la configuración de la prioridad de cada dimensión contextual.



Figura 13. Configuración de la prioridad de las dimensiones contextuales.

#### Caso de Uso CU4: Configurar Muestreo

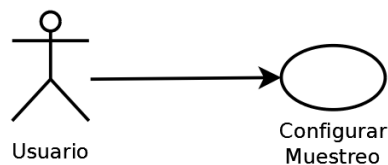


Diagrama 4. Diagrama de caso de uso *Configurar muestreo*.

Tabla 17. Caso de uso *Configurar muestreo*.

Descripción CU4	
<b>Actor</b>	Usuario
<b>Precondiciones</b>	Abrir la ventana de configuración (Figura 12 (a)).
<b>Propósito</b>	Configurar los tiempos de muestreo de cada Dimensión Contextual.
<b>Resumen</b>	El usuario puede configurar el tiempo de muestreo de las dimensiones contextuales, con el fin de que tenga mayor control sobre la precisión de la recomendación y el gasto de la batería.
<b>GUI</b>	Interfaz de configuración de recomendación (Figura 14).
Curso normal de Eventos CU4	
Acción de Actores	Respuesta del sistema
1. El usuario ingresa el tiempo de muestreo en la dimensión contextual seleccionada.	2. Establece la configuración de muestreo de cada dimensión contextual.

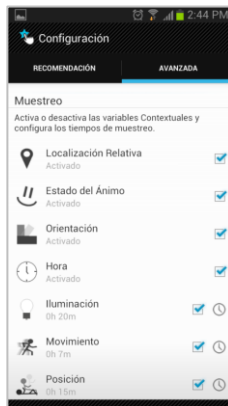


Figura 14. Configuración del muestreo de las dimensiones contextuales.

**Caso de Uso CU5: Sincronizar Información**

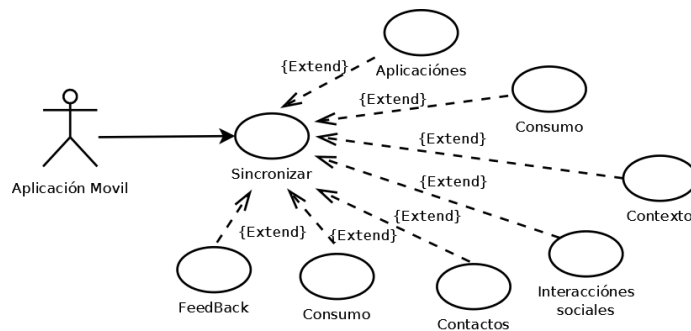


Diagrama 5. Diagrama caso de uso *Sincronizar información*.

Tabla 18. Caso de uso *Sincronizar información*.

Descripción CU5	
<b>Actor</b>	Aplicación móvil.
<b>Precondiciones</b>	Registrar usuario en sistema.
<b>Propósito</b>	Sincronizar todos los datos con el repositorio en el servidor central.
<b>Resumen</b>	Sincronización de la información correspondiente al consumo, contexto interacciones sociales, contactos, aplicaciones y el feedback de las recomendaciones.
<b>GUI</b>	No tiene una interfaz gráfica de usuario debido a que es una funcionalidad que se efectúa de forma transparente.
Curso normal de Eventos CU5	
Acción de Actores	Respuesta del sistema
1. Cliente envía información a la aplicación web, por medio de mensajes JSON.	2. Almacena la información y retorna un OK para confirmar que el almacenamiento se realizó con éxito.
3. Elimina la información de la base de datos local.	



Curso Alterno de Eventos CU5	
Acción de Actores	Respuesta del sistema
	2. Conexión con el servidor no disponible
3. Almacena el mensaje en pendientes y lo vuelve a enviar en cuanto exista conexión con el servidor.	
	2. Error al almacenar la información
3. Vuelve a armar el mensaje y lo envía nuevamente	

### Caso de Uso CU6: Solicitar Recomendación

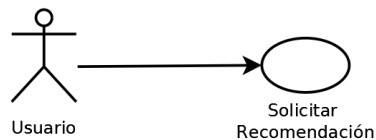


Diagrama 6. Diagrama caso de uso *Solicitar recomendación*.

Tabla 19. Caso de uso *Solicitar recomendación*.

Descripción CU6	
<b>Actor</b>	Usuario
<b>Precondiciones</b>	Registrar usuario en sistema.
<b>Propósito</b>	Solicitar una recomendación de aplicaciones móviles.
<b>Resumen</b>	Proceso que inicia el usuario a fin de solicitar una recomendación a través de la aplicación móvil.
<b>GUI</b>	interfaz gráfica de usuario, donde se lista las aplicaciones recomendadas
Curso normal de Eventos CU5	
Acción de Actores	Respuesta del sistema
1. Cliente Presiona el botón de recomendación en el Widget de la aplicación.	2. La aplicación genera el JSON con el contexto y peso de las dimensiones contextuales actuales.
	3. Se procesa la información y se arma el JSON con la aplicaciones a recomendar
	4. Se despliega la recomendación en la interfaz gráfica correspondiente
Curso Alterno de Eventos CU5	
Acción de Actores	Respuesta del sistema
	2. Información por dimensión contextual no disponible.
	3. Notifica al usuario que no es posible realizar una recomendación en ese momento.



Figura 15. Recomendación de aplicaciones móviles.

## 5.2. Arquitectura del Sistema

### 5.2.1. Aplicación Móvil

**Módulo de captura:** Encargado de la recolección y captura de la información correspondiente al contexto físico, social y cognitivo. Adicionalmente se detectan y almacenan los eventos relacionados al consumo de las aplicaciones móviles.

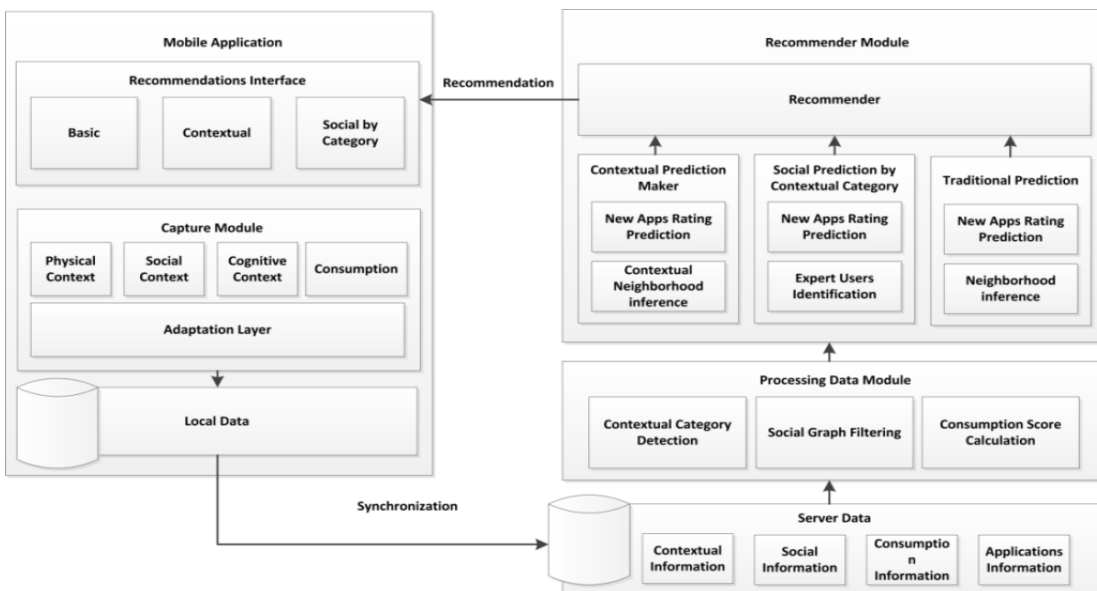


Figura 16. Arquitectura del Sistema de recomendación Propuesto, Vanilla.

**Datos Locales (Local Data):** Corresponde al repositorio en donde se almacena de manera temporal, la información capturada en el módulo anteriormente mencionado. Esta

información es borrada del móvil cada vez que se transmite al servidor, debido a las restricciones de memoria que pueden darse en los dispositivos móviles.

**Interfaz de recomendación (Recommendation interface):** Interfaz gráfica donde se despliegan las recomendaciones finales de cada técnica planteada en esta propuesta (capítulo 6).

### 5.2.2. Sincronización

Sincroniza los datos entre la base de datos local en la aplicación móvil y el repositorio central en el servidor, mediante un intercambio de mensajes, mostrados en la Tabla 20 y en el (Anexo D).

**Tabla 20.** Información/mensajes de sincronización.

Tipo de mensaje	Descripción
Registro	Información del Smartphone como modelo, fabricante, versión del sistema operativo y el número de celular del usuario
Aplicaciones	Eventos de instalación y desinstalación de las aplicaciones móviles
Consumo	Eventos correspondientes al uso de aplicaciones móviles.
Contexto	Cambio en los estados de las dimensiones contextuales.
Interacciones Sociales	Registro de llamadas y SMS
Contactos	Contactos disponibles en el Smartphone
FeedBack	Aplicaciones seleccionadas en la interfaz de recomendación y los tiempos de permanencia en el sistema de recomendación.

### 5.2.3. Servidor

**Repositorio central (Server data module):** Base de datos central del sistema.

- **Información contextual (contextual data):** Corresponde al conjunto de los posibles estados de las variables contextuales.
- **Información social (Social data):** Se define como el grafo que representa la red social basada en contactos, llamadas y sms's (Capítulo 5).
- **Información de consumo (Consumption information):** Se define como las puntuaciones de las aplicaciones en cada estado de todas las variables contextuales, calculadas a través de los eventos de consumo y de contexto de cada usuario (Capítulo 4).

- **Información de Aplicaciones (Applications Information):** Corresponde a toda la información abstraída del portal de Google Play, que describe a cada aplicación móvil.

**Módulo de procesamiento de información (Processing data module):** Encargado de adaptar toda la información contenida en el repositorio central, para la ejecución de los algoritmos de recomendación.

- **Detector de categoría contextual (Contextual category detection):** Encargado de determinar cuál es la categoría de consumo predominante en el contexto actual del usuario (Capítulo 4).
- **Calculo de la calificación de aplicaciones móviles (Consumption score calculation):** Calcula la calificación de la aplicación en el contexto actual y de forma general (Capítulo 4).
- **Filtrado del grafo social (Social graph filtering):** Se filtra el grafo social y se retornan los contactos del usuario en cuestión, junto con el número de llamadas y SMS que sucedieron entre este y todos sus contactos.

**Módulo de Recomendación:** Ejecuta los algoritmos del filtrado colaborativo con las entradas adaptadas para cada técnica planteada.

- **Predicción Contextual (Contextual Prediction):** Se realiza la predicción de la calificación de las aplicaciones, en función del contexto actual del usuario (Capítulo 6).
- **Predicción Social - Contextual (Social Prediction by Contextual Category):** Se realiza la predicción de la calificación de las aplicaciones, en función de usuarios expertos por categorías en el contexto actual del usuario (Capítulo 6).
- **Predicción Tradicional (Traditional Prediction):** Se realiza la predicción de la calificación de las aplicaciones móviles, tomando en cuenta la vecindad tradicional (Capítulo 6).
- **Recomendador (recommender):** Entrega de forma organizada, toda la información resultante de la ejecución de los tres algoritmos de predicción, al móvil.

### 5.3. Consideraciones de la Implementación del prototipo

A continuación se describe el alcance del prototipo, las tecnologías empleadas para su realización y su publicación:

#### 5.3.1. Tecnologías Empleadas

##### 5.3.1.1. Aplicación Móvil:

La aplicación móvil fue desarrollada en la plataforma ANDROID (Android), donde uno de los principales retos en este módulo, fue la implementación de una metodología eficiente para la captura de cambios en el contexto del usuario. A pesar de la existencia de herramientas y frameworks como “funF” (Aharony, 2012), orientados a la realización de esta tarea, el excesivo consumo de recursos (aproximadamente 23 servicios) ocasionaba el cierre esporádico de la aplicación, cuestión supremamente contraproducente en un ambiente real.

De esta forma se optó por implementar una metodología de captura propia orientada a eventos (*Broadcast Receivers*), que pretende superar los inconvenientes de este estilo.

En este orden de ideas, la captura de información contextual se hace sobre ocho dimensiones contextuales, donde se registra cada cambio de estado de acuerdo a un tiempo determinado de muestreo, a excepción de las dimensiones contextuales de estado de ánimo y localización relativa que requieren el ingreso de información del usuario a través de un Widget en la pantalla *home* del Smartphone (Figura 12 (a)).

**Tabla 21.** Dimensiones contextuales capturadas.

Dimensión Contextual	Fuente de Información	Tipo de Información
Localización Relativa	Usuario	Contexto Físico
Estado de Animo	Usuario	Contexto Cognitivo
Orientación	Smartphone	Contexto Físico
Hora	Smartphone	Contexto Físico
Iluminación	Smartphone	Contexto Físico
Movimiento	Smartphone	Contexto Físico
Posición	Smartphone	Contexto Físico
Clima	Servicio Web	Contexto Físico

De este modo, la aplicación es capaz de capturar toda la información del usuario que se considere pertinente para la realización de las recomendaciones de una forma eficiente (consumo, social y registros de su comportamiento, etc.). Por otro lado, el usuario podrá solicitar que se le recomienden aplicaciones que este no haya consumido y de la misma forma tendrá la posibilidad de personalizar y configurar las opciones que este crea convenientes como la activación de sensores, señalar algunos datos relacionados con su contexto actual (variables explícitas), entre otras.

### 5.3.1.2. Servidor Central

#### Aplicación web

Aplicación web en JAVA EE que se compone de Servlets principalmente. Entre sus procesos se encuentran la recepción y procesamiento de la información provista desde la aplicación móvil, la ejecución de los algoritmos de recomendación (expuestos en el Capítulo 6) que necesiten de datos en tiempo real del usuario (contexto actual del usuario para la *recomendación contextual*), entre otros.

## Proceso demonio

Con el objetivo de mejorar la eficiencia y balancear el procesamiento en el sistema, se desarrolló una aplicación JAVA de apoyo cuya ejecución se hará de forma periódica con intervalos de espera relativamente cortos (horas). Entre sus procesos se encuentran la actualización de la estructura de la red social, el cálculo de usuarios expertos, la ejecución de los algoritmos de recomendación que prescindan de los datos en tiempo real del usuario (*recomendación tradicional* y *recomendación social-contextual*, teniendo en cuenta que es posible reducir el número de veces de ejecución de dichos algoritmos debido a que la variación de los ratings calculados a través de estos algoritmos, es regular en un tiempo relativamente corto (horas)), entre otros.

### 5.3.2. Publicación

El prototipo desarrollado para validar el sistema de recomendación propuesto (Vanilla), se implementó con la intención de realizar un acercamiento a lo que una aplicación comercial se refiere. En este sentido, la misma se lanzó en Google Play<sup>22</sup> con el objetivo de generar medios adecuados para su descarga y tener un mayor control de errores y fallas a través de actualizaciones.

De acuerdo a lo anterior, la aplicación fue diseñada con el fin de satisfacer algunos requerimientos que un ambiente de producción implica, tales como bajo consumo de ancho de banda, eficiencia y rapidez en las transacciones, bajo consumo de batería, entre otros. A continuación se presentan las soluciones propuestas en orden de cumplir las necesidades mencionadas previamente.

#### Bajo consumo de ancho de banda

Sincronización de datos disparada por alertas a través de mensajes JSON entre la aplicación móvil y la aplicación web. Se utilizó JSON debido a que es un formato de documento mucho más simple, fácil de interpretar y liviano respecto a otros.

#### Procesamiento eficiente de la información (TIEMPO Y MEMORIA)

Utilización de *Java Servlets* para la recepción y procesamiento (parcial) de la información brindada desde los dispositivos móviles. Se resalta que *Java Servlets* es uno de los frameworks más robustos en el manejo de solicitudes HTTP<sup>23</sup>. Adicionalmente se utilizan procesos demonio (aplicación JAVA) para balancear el procesamiento en la comunicación entre el servidor y móvil.

#### Alta disponibilidad

Segmentación de la transmisión de la información por eventos, realizados en horas del día que se estiman puede ser menos concurrente en las conexiones. Optimización de algoritmos para consumir menos procesamiento y ofrecer mayor disponibilidad.

---

<sup>22</sup> <https://play.google.com/store/apps/details?id=edu.unicauca.android.vanilla>

<sup>23</sup> <http://www.techempower.com/benchmarks/>

### **Bajo consumo de batería en los dispositivos móviles**

Teniendo en cuenta que una aplicación móvil, también es atractiva por su consumo moderado de energía, Vanilla se ha desarrollado como una aplicación orientada a eventos a través de la utilización de *BroadCast receivers* en su mayoría (para detectar el consumo y cambios en el contexto es necesaria la implementación de servicios). La filosofía de estos componentes de Android, permite la realización de tareas complejas con procesos simples y eficientes, en pro de optimizar el consumo de energía.

### **Usabilidad**

Diseño llamativo en la aplicación móvil, ya que es un sistema comercial. Interfaces gráficas agradables, fáciles de entender y manejar.

## **5.4. Resumen**

En resumen, este capítulo trata todo lo relacionado con la implementación del prototipo experimental definido en el último objetivo específico. De esta manera se presentaron todos los elementos necesarios para la descripción del mismo incluyendo casos de uso, imágenes de la interfaces principales, la arquitectura del sistema, descripción del aplicativo móvil, tecnologías seleccionadas y consideraciones para la implementación del sistema en un entorno de producción.

## Capítulo VI, Experimentación y Evaluación

En esta sección se presenta evaluación de las aproximaciones abordadas en este proyecto; la cual está basada en la metodología seguida por López y otros (López, 2010), exponiendo dos secciones principales, orientadas a la descripción metodológica de la evaluación y a la ejecución de los experimentos.

### 6.1. Metodología

Esta sección está dedicada a la descripción metodológica de los experimentos realizados por el presente proyecto de investigación. Para tal fin se expondrá la planeación del experimento, seguida por la descripción de las métricas de evaluación utilizadas, terminando por el plan de pruebas adoptado.

#### 6.1.1. Planeación del experimento

Esta sección expone la planeación de los experimentos realizados, retomando y/o definiendo objetivos, hipótesis, selección de sujetos y objetos experimentales, entre otros.

##### 6.1.1.1. Definición del objetivo

El objetivo de evaluación de los experimentos realizados es “medir la calidad y desempeño de las aproximaciones presentadas en esta propuesta, haciendo uso de la plataforma “Vanilla”, contrastando sus resultados”.

##### 6.1.1.2. Formulación de Hipótesis

Como fue mencionado en secciones anteriores, el presente proyecto busca la comprobación o negación de dos hipótesis, relacionadas a continuación:

- H1n (Hipótesis nula) El uso de calificaciones inferidas a partir del consumo y el contexto físico del usuario **NO MEJORA** la calidad en los ARS
- H1a (Hipótesis Alternativa) El uso de calificaciones inferidas a partir del consumo y el contexto físico del usuario **MEJORA** la calidad en los ARS
- H2n (Hipótesis nula) Otorgar una mayor relevancia a la apreciación de usuarios expertos dentro de la red social del usuario por cada categoría de aplicaciones **NO INCREMENTA** la calidad de las recomendaciones de los ARS basados en FC.
- H2a (Hipótesis Alternativa) Otorgar una mayor relevancia a la apreciación de usuarios expertos dentro de la red social del usuario por cada categoría de aplicaciones **INCREMENTA** la calidad de las recomendaciones de los ARS basados en FC.

La comprobación o negación de las hipótesis enunciadas anteriormente se hace mediante la ejecución de un mismo experimento, el cual involucra la comparación simultánea de las tres aproximaciones tratadas al interior de este proyecto, basados en el trabajo de Hayes (Hayes, y otros, 2002).

##### 6.1.1.3. Selección de Variables

Ya que el objetivo de las dos hipótesis es medir la calidad y desempeño de dos aproximaciones para la recomendación de aplicaciones móviles, se utilizarán métricas de



calidad “on-line” y “off-line”, en conjunto con medidas de rendimiento; las cuales son descritas a profundidad en la sección de métricas. Por tal razón la experimentación consta de las siguientes partes

#### ***Experimentación de calidad “off-line”***

El objetivo de este experimento es realizar una evaluación de la predicción de calificaciones realizada por cada una de las aproximaciones o técnicas propuestas. Así las variables a utilizar son las siguientes:

Variables Independientes

- Registro del consumo del usuario.
- Calificaciones calculadas a partir del consumo del usuario.
- Aplicaciones disponibles en el mercado

Variables Dependientes

- Calificaciones predichas por las aproximaciones propuestas.

#### ***Experimentación de rendimiento “on-line”***

El objetivo de este experimento es realizar una evaluación de la utilidad de las recomendaciones realizadas por cada una de las aproximaciones propuestas. Así las variables a utilizar son las siguientes:

Variables Independientes

- Aplicaciones disponibles en el mercado

Variables Dependientes

- Registro de la interacción del usuario con las recomendaciones
- Registro del consumo del usuario.

#### ***Experimentación de rendimiento***

El objetivo de este experimento es determinar el rendimiento del SR propuesto, mediante medidas de eficiencia temporal.

Variables Independientes

- Capacidad de procesamiento (las medidas se hacen sobre un mismo entorno)

Variables Dependientes

- Tiempo de respuesta de los algoritmos de recomendación.
- Número de usuarios, calificaciones y aplicaciones consideradas.

#### **6.1.1.4. Selección de Sujetos**

Debido a las características del dominio de la recomendación de aplicaciones móviles descrito en la introducción de este documento, los sujetos de estudio debían cumplir con el siguiente perfil:

- Poseer un SmartPhone Android.
- Utilizar constantemente la aplicación “Vanilla”.

Así, a diferencia del trabajo de Aharony (Aharony, 2012), al no contar con los recursos económicos para proveer dispositivos a los participantes de este experimento, se solicitó la participación voluntaria de personas mediante redes sociales como Facebook y Twitter. De este modo se logró contar con un número de aproximadamente 50 personas que instalaron la aplicación, de las cuales alrededor de 12 la utilizaron constantemente. Dicha población en su mayoría corresponde a personas allegadas a los desarrolladores del este proyecto, radicadas en Popayán y Pasto principalmente.

#### **6.1.1.5. Objetos experimentales**

Tal como fue mencionado, el objetivo de la experimentación es determinar la calidad y rendimiento de las aproximaciones trabajadas en este proyecto. Para tal fin se implementó un prototipo funcional para la recomendación de aplicaciones móviles denominado “Vanilla” el cual fue descrito en secciones anteriores. Entre sus principales características, este prototipo permite:

- La recolección de datos sobre el consumo de aplicaciones por parte del usuario.
- El registro de información de interacción del usuario con la aplicación de recomendación.
- La agregación de la recomendación a partir de los resultados de las tres aproximaciones a evaluar.
- El registro de los tiempos de procesamiento de las peticiones.

De esta manera, por medio de Vanilla, se recolectan los datos para el cálculo de las métricas de evaluación “on-line”, “off-line” y de rendimiento presentadas en este documento. Por otra parte, tal como es sugerido por Hayes (Hayes, y otros, 2002), la interacción del usuario con la recomendación se realiza guardando el carácter “anónimo” sobre la procedencia de las recomendaciones.

#### **6.1.2. Métricas de evaluación**

Actualmente, muchas técnicas son utilizadas para la evaluación de sistemas de recomendación; varias de ellas involucran un análisis “off-line” empleando métodos de áreas como el aprendizaje automático y recuperación de información. Este tipo de técnicas de evaluación generalmente se basa en el análisis de un conjunto de datos reales, brindando una visión general del desempeño de un sistema de recomendación sin la necesidad de la respuesta del usuario ante la recomendación. Su gran ventaja se evidencia al comparar más de una aproximación a la recomendación, pudiendo utilizar los mismos datos como insumo para los obtener una visión objetiva del desempeño de varios algoritmos.

Sin embargo, a pesar de la utilidad de este tipo de técnicas, diversos estudios recomiendan el uso de técnicas “on-line” para medir de mejor manera el impacto de los SR (Hayes, y otros, 2002) (Böhmer, et al., 2013). Las técnicas “on-line”, a diferencia de las técnicas “off-line”, precisan la existencia de información sobre la respuesta del usuario ante recomendaciones reales, brindando una vista específica del desempeño de un algoritmo en especial. Este tipo de técnicas miden principalmente la utilidad del SR, con respecto a la apreciación del usuario, siendo su principal desventaja el carácter “exclusivo” de sus resultados, generalmente aplicables al análisis de un solo algoritmo a la vez. En este sentido, trabajos como el desarrollado por Hayes (Hayes, y otros, 2002), presentan alternativas para

la evaluación simultánea de varios algoritmos por medio del manejo de interfaces de usuario y agregación de resultados de las aproximaciones consideradas. Este tipo de aproximación fue utilizada para el diseño del algoritmo de agregación de recomendaciones tratado anteriormente en este documento.

Así, en esta sección se expone la evaluación de las aproximaciones presentes en esta propuesta, con respecto a distintas métricas de tipo “off-line” y “on-line”, además

Uno de los aspectos críticos para evaluar la eficiencia de este tipo de sistemas, es el establecimiento de medidas de desempeño adecuadas, con su respectivo plan de pruebas. En este sentido, el presente capítulo describirá las evaluaciones y plan de pruebas propuesto para el cálculo de las medidas que se consideran son relevantes para la evaluación de Vanilla. Finalmente se presentarán algunos resultados preliminares de las mismas.

### 6.1.2.1. Medida de eficacia (off-line)

La eficacia se define como una medida de relevancia y pertinencia al observar el número de aplicaciones seleccionadas sobre el número total de aplicaciones mostradas por cada recomendación que se solicita (Ecuación 22).

$$K_{u,\tau} = \frac{Ns_{u,\tau}}{Nt_{u,\tau}} \quad (22)$$

En la expresión en la parte superior,  $Ns_{u,\tau}$  corresponde al número de aplicaciones seleccionadas en la recomendación  $\tau$  por el usuario  $u$ , y  $Nt_{u,\tau}$  corresponde al número de aplicaciones mostradas o recomendadas en la misma. Generalmente, en este tipo de sistemas las solicitudes de recomendación se dan de manera periódica, por lo que es posible observar la evolución de esta métrica a medida de que el sistema recolecta más información y percibir si hay mejoras o no al transcurrir el tiempo.

Con el objetivo de facilitar la observación y el análisis de dicha evolución al interior del sistema de recomendación de este proyecto, se ha propuesto una nueva dimensión denominada  $Tao(\tau)$  que representa el orden sucesivo en que se solicita cada recomendación. En otras palabras, las recomendaciones se expresan en el orden en el que cada usuario las haya solicitado (Así, los valores de  $Tao(\tau)$  corresponden al orden de las recomendaciones solicitadas como “primera”, “segunda”, “tercera”, ..., “n-ésima”). De esta forma, la eficacia puede expresarse en función de  $\tau$ , como está descrito en la (Ecuación 23).

$$K(\tau)_u = \frac{Ns_u(\tau)}{Nt_u(\tau)} \quad (23)$$

Posteriormente, se define el promedio de eficacia de un usuario, ponderando los valores de la eficacia en cada recomendación dentro del rango que se defina (delta  $Tao(\tau)$ ) (Ecuación 24).

$$P_{k_u} = \frac{\sum_{\tau \in \Delta\tau} K_{u,\tau}}{\Delta\tau} \quad (24)$$

## Eficacia promedio general

Métrica que representa la eficacia promedio de un conjunto de usuarios en el sistema.

$$Pr_{total} = \frac{\sum_{i=1}^U P_{k_{ui}}}{U} \quad (25)$$

En la (Ecuación 25),  $P_{k_{ui}}$  el promedio de la eficacia del usuario  $ui$ , y  $U$ , el número de usuarios que se tienen en cuenta para calcular esta medida (generalmente, todos los usuarios del sistema).

## Evolución eficacia

Medida que pretende representar cómo evoluciona la eficacia promedio del sistema, respecto a  $Tao$  ( $\tau$ ).

$$Ev(\tau) = \frac{\sum_{i=1}^{u(\tau)} K_{u_i}(\tau)}{u(\tau)} \quad (26)$$

En la (Ecuación 26), por cada  $\tau$ , se calculará la sumatoria de las eficacias  $K_{u_i}(\tau)$ , de un conjunto de usuarios  $u(\tau)$  cuyo tamaño se define por la cantidad de usuarios que hayan solicitado la recomendación correspondiente al valor de  $\tau$ .

## Eficacia en los diferentes tipos de recomendación

Teniendo en cuenta las expresiones descritas previamente, cada una de ellas será aplicada para cada tipo de recomendación (tradicional, contextual, social-contextual), con el objetivo de comparar los resultados y determinar cuál es el que más aceptación tiene (por ende el que mejores resultados presenta), siendo esta prueba, una de las razones por la cual se definieron tres tipos de recomendación, para que cada uno sirva como referencia del otro.

### 6.1.3. Medida de *interés*, respecto al tiempo de observación de las interfaces de recomendación (on-line)

Existen otras medidas o métricas que pueden brindar evidencias sobre la aceptación y uso del sistema, entre las cuales se destacan el tiempo que el usuario gasta observando los ítems que se le recomiendan sobre las interfaces que los exponen (Baltrunas, 2011). Es lógico pensar que entre más tiempo se emplee en la observación de dichas interfaces, puede haber un mayor grado de *interés* sobre las recomendaciones que se realizan (Baltrunas, 2011) (independientemente si el usuario selecciona un ítem o no, esto se describe en la medida expuesta al principio de este capítulo).

### Tiempo empleado para la visualización de las interfaces de recomendación por usuario

Se define al *tiempo de observación por recomendación* de un usuario, como la suma de los tiempos de cada evento registrado en la base de datos referentes a la misma página, representado a través de la (Ecuación 27).

$$T_{p_k, \tau_i, u_o} = \sum_{j=0}^J t_{j, p_k, \tau_i, u_o} \quad (27)$$

En (Ecuación 27),  $t_{j,p_k,\tau,u_o}$ , corresponde al tiempo en el registro  $j$  (en la base de datos), y  $J$  al número total de registros referentes a la recomendación  $\tau$ , de la página  $p_k$ .

### **Evolución de tiempo empleado para la visualización de las interfaces de recomendación, por página**

Una vez definido el factor *tiempo de observación por recomendación* y teniendo en cuenta la dimensión *Tao* ( $\tau$ ) descrita en la *Medida de eficacia*, es posible representar cómo evolucionan o varían los tiempos de cada página por cada recomendación que los usuarios soliciten, como se muestra en la (Ecuación 28).

$$Et(\tau) = \frac{\sum_{i=1}^{u(\tau)} T_{p_k,u_i}(\tau)}{u(\tau)} \quad (28)$$

En (Ecuación 28),  $u(\tau)$  corresponde al conjunto de usuarios cuyo tamaño se define por la cantidad de usuarios que hayan solicitado la recomendación en el valor de  $\tau$  correspondiente, y  $T_{p_k,u_i}(\tau)$  al *tiempo de observación por recomendación en función de  $\tau$* , del usuario  $u_i$ .

### **Promedio de tiempo empleado para la visualización de las interfaces de recomendación, por página**

Esta medida, representa el *tiempo promedio total de visualización* por cada página, donde se hace un promedio de la sumatoria de tiempos de observación por recomendación, registrados de un usuario (referentes a una página específica).

$$Ptemp_{p_k} = \frac{\sum_{i=1}^U \sum_{\tau=1}^P T_{p_k,\tau,u_i}}{U} \quad (29)$$

En (Ecuación 29),  $P$  representa el número total de recomendaciones solicitadas por el usuario  $u_i$  (el orden máximo de  $\tau$ ), del mismo modo  $U$  corresponde al número total de usuarios tenidos en cuenta para este cálculo (generalmente todos los usuarios del sistema) y  $T_{p_k,\tau,u_i}$  al *tiempo de observación por recomendación* del mismo usuario  $u_i$ .

### **Promedio de tiempo empleado para la visualización de las interfaces de recomendación, total**

Similar al término anterior, se toman las sumatorias de tiempos de observación por recomendación registrados de cada usuario, pero esta vez se tienen en cuenta los registros de todas las páginas. La expresión es similar a la de la ecuación No. con la única diferencia que en el factor  $T_{p_k,\tau,u_i}$ , se incluyen y se suman los tiempos de observación de las otras páginas.  $T_{p_k,u_i}$ .

$$Ptemp_{total} = \frac{\sum_{i=1}^U \sum_{\tau=1}^P T_{total,\tau,u_i}}{U} \quad (30)$$

#### **6.1.4. MAE (Off-line)**

Se define el MAE (Mean Absolute Error) como la medida promedio de cuan acertadas o precisas son las *predicciones* hechas por cualquier algoritmo de Filtrado Colaborativo (en este caso solo se refiere al componente Algoritmo, de cada técnica de recomendación). De esta forma, el MAE se da por la siguiente expresión (Ekstrand, y otros, 2011).

$$MAE = \frac{1}{n} \sum_{u,i} |p_{u,i} - r_{u,i}| \quad (31)$$

En la (Ecuación 31), el *error* (la expresión contenida en el valor absoluto), es la diferencia entre el rating predicho y el real del ítem *i*. Así, *n* es el tamaño del conjunto de ítems que se tienen en cuenta,  $p_{u,i}$  es el rating predicho por el algoritmo de Filtrado Colaborativo y  $r_{u,i}$  es el rating real.

Uno de los métodos más utilizados para el cálculo del MAE, es la realización de Benchmark's con usuarios conocedores del tema, con el fin de obtener las verdaderas *apreciaciones* y *ratings* de los ítems considerados ( $r_{u,i}$ ). Sin embargo, dadas las condiciones de consumo e inferencia automática de calificaciones en esta propuesta, se prescinde de la realización de Benchmark's (ya que en ningún caso se incluye la apreciación del usuario sobre el rating predicho), y se toman estas calificaciones como la referencia  $r_{u,i}$ .

## Calculo del MAE

La metodología adoptada para el cálculo del MAE, consiste en tomar un conjunto de aplicaciones que hayan sido consumidas por el mismo usuario o diferentes usuarios, infiriendo sus calificaciones a partir del consumo que hayan tenido. Posteriormente, asumiendo que las aplicaciones en cuestión van a ser recomendadas, se aplicarán los algoritmos de FC respectivos para predecir sus ratings, obteniendo los dos términos necesarios para determinar el *error*, para cada elemento de conjunto.

$$MAE = \frac{1}{n} \sum_{u,i} |Predicción_i - Calificación a partir del consumo_i| \quad (32)$$

Sin embargo, debido a que la inferencia de calificaciones a partir del consumo, está sujeta a cambios en el tiempo debido al *factor de actualidad* (Capítulo 3). Es recomendable que esta medida sea tomada 30 días después de haber sido instalada cualquier aplicación (Donde se involucran todos los eventos de consumo, por ende la calificación probablemente sea la más alta), para eliminar la presencia de errores.

## 6.2. Plan de ejecución de pruebas y resultados

Todas las pruebas a las cuales se hace referencia a continuación, son parte de un proceso **constante** de detección y análisis de la información recolectada por Vanilla. Este proceso se realizó periódicamente debido a que, por la naturaleza de las técnicas utilizadas, el sistema tiende a mejorar la calidad de sus respuestas, dependiendo de una mayor y prolongada interacción del usuario con la aplicación.

En este orden de ideas, se realizó la programación de un módulo de generación de reportes que procese la información presente en la base de datos, y la muestre de forma coherente y entendible. Esto con el objetivo de llevar un seguimiento del comportamiento del sistema y de los usuarios, casi en tiempo real.

### 6.2.1. Plan de pruebas - eficacia

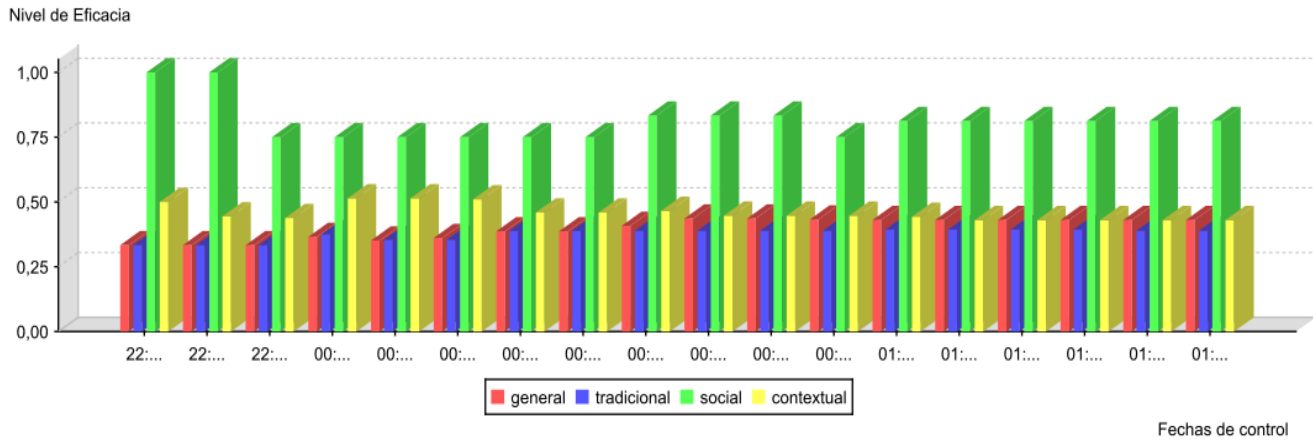
Debido a que el sistema propuesto para el presente trabajo se desarrolló con el fin de especializar el número de aplicaciones relevantes a ser recomendadas al usuario; es de suma importancia detectar el grado de la relevancia o impacto de las recomendaciones que se generan, a través de la *eficacia promedio*, junto con su *evolución* (respecto a la dimensión *Tao*, definida).

**Tabla 22.** Plan de pruebas - eficacia.

<b>Análisis de la medida de eficacia promedio general, y de su evolución respecto al tiempo</b>
<b>Plan de calidad:</b> permite evaluar y comparar la calidad de las técnicas de recomendación propuestas, analizando el nivel de <i>eficacia promedio</i> de cada una. De la misma forma, la evolución de esta variable, es una medida complementaria para poder determinar causas y efectos, que inciden sobre su nivel.
<b>Procedimiento</b>
La inferencia del nivel de la <i>eficacia promedio</i> de cada técnica, consiste en aplicar la expresión correspondiente (ecuación No.) en cada uno de los casos (implementada por el módulo de generación de reportes). Adicionalmente, para observar cual ha sido el cambio de esta variable, dicha acción se ejecuta en una vez al día (debido al periodo en que se reciben los nuevos eventos de consumo) utilizando la información presente en cada "Back-up" de la base de datos generado diariamente. Lo que respecta a al cálculo de la <i>evolución</i> del nivel de <i>eficacia</i> , este reporte ya se encuentra implementado en el módulo de generación de reportes mencionado, por lo que solo basta con ejecutarlo.

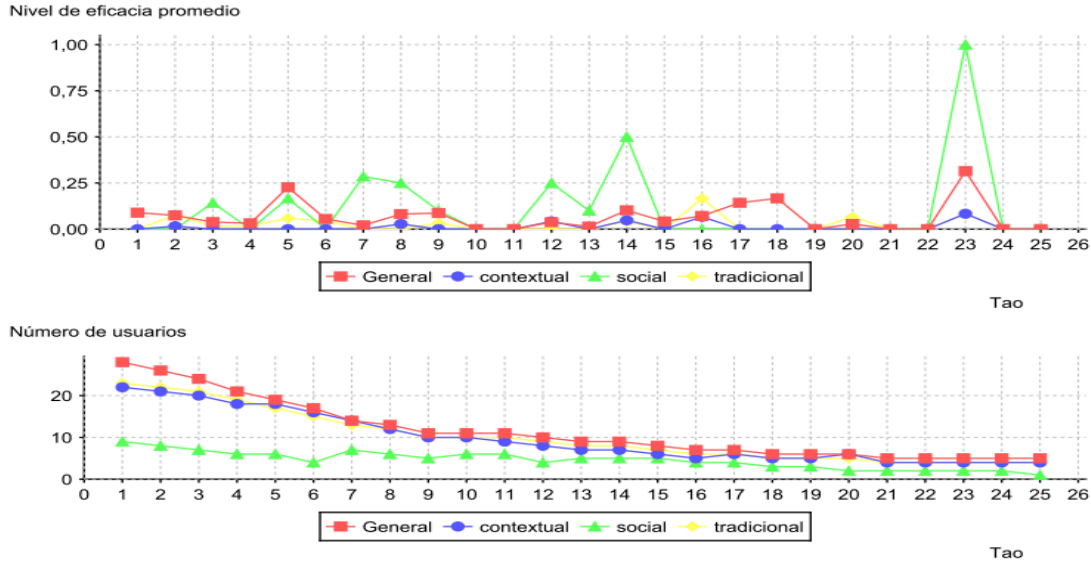
#### 6.2.1.1. Resultados

La cantidad de resultados es dependiente de la fecha de la habilitación de recomendaciones en Vanilla (24 de Mayo de 2013); por tal razón el informe sobre el nivel de *eficacia* es presentado a partir de dicho momento. De esta manera la Figura 17, presenta las muestras del nivel de eficacia capturadas durante el tiempo de observación



**Figura 17.** Eficacia promedio general.

Los resultados muestran una tendencia positiva respecto a la *eficacia promedio* de la técnica social, la cual tiene un nivel muy superior respecto a la *eficacia promedio* de las otras técnicas, tal y como se expone en la Figura 17. Sin embargo, esta diferencia considerable se da por las pocas recomendaciones sociales que se han provisto hasta el momento (pocas aplicaciones dadas por la técnica social por cada recomendación, debido a la dispersión en la red social de contactos). De esta manera, estos resultados muestran una tendencia alentadora respecto a la mejoría que la técnica principal propuesta en este proyecto, sugiere.



**Figura 18.** Evolución de la eficacia en función de Tao.

Por otro lado en la Figura 18. Es posible observar cómo ha sido la evolución de la *eficacia promedio* para diferentes de valores de *Tao* (donde 1, corresponde a la primera recomendación que se solicita y así sucesivamente). Sin embargo, el comportamiento de dicha evolución aún no ofrece información clara y estable, debido al pequeño número de usuarios que han solicitado un elevado número de recomendaciones (tal y como se muestra



en la gráfica de debajo de la misma figura, que corresponde al número de usuarios sobre el que se calcula el *Nivel de eficacia promedio* por cada *Tao*).

### 6.2.2. Plan de pruebas - interés

Con el objetivo de medir el interés relacionado con el tiempo de observación que un usuario gasta en cada una de las páginas donde se expone la recomendación, se ha implementado un módulo lógico para la detección del tiempo en milisegundos. En este orden de ideas, por cada recomendación, se provee un número máximo de 9 aplicaciones (Capítulo 4), en donde se distribuyen en cantidades de tres, en tres páginas disponibles.

#### NOTA:

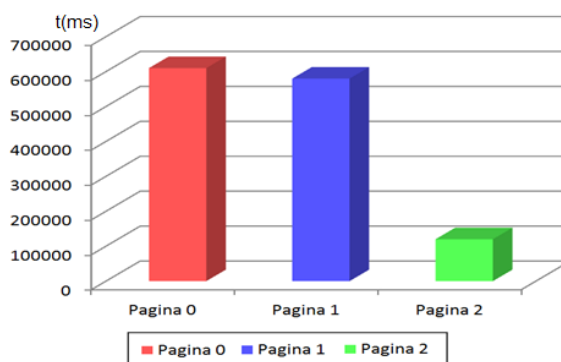
La cantidad de tiempo de observación en cada página, puede verse afectado en el caso de que el sistema recomiende menos de 9 aplicaciones y se habiliten menos páginas.

**Tabla 23.** Plan de pruebas – interés.

<b>Grado de interés del usuario, a través del análisis del tiempo de observación a las interfaces</b>
<b>Plan de calidad:</b> permite evaluar y comparar el interés que el usuario pueda tener respecto a los servicios del sistema, específicamente al de recomendación.
<b>Procedimiento</b>
La detección del tiempo de observación por página, para su posterior análisis, consiste en la ejecución de la expresión correspondiente $n(\text{Tiempo})$ , siguiendo la misma dinámica del cálculo de la <i>eficacia promedio</i> .

#### 6.2.2.1. Resultados

La siguiente figura (Figura 19) presenta el tiempo promedio de observación (general y por páginas), de acuerdo a la fecha de la habilitación de recomendaciones en Vanilla, el cual es dependiente del periodo de observación.



**Figura 19.** Tiempo promedio de observación por página.

Cómo era de esperarse, la página que cuenta con el mayor *Promedio de tiempo empleado para la visualización de las interfaces de recomendación, por página* (medida de interés), es

la página 0, donde se exponen las aplicaciones con el mayor rating, por ende, más relevantes para el usuario. De igual forma, teniendo en cuenta estos resultados, es posible afirmar (parcialmente) que el nivel de interés es moderado (con tendencias a elevado), debido a que los usuarios si se toman el tiempo necesario para analizar la página 1, más no están dispuestos a seguir explorando en la siguiente página (página 2).

### 6.2.3. Plan de pruebas - MAE

Como se ha mencionado anteriormente, para este caso la medida del MAE corresponde al nivel de precisión únicamente del algoritmo de filtrado colaborativo. De esta forma se calculará el MAE para los algoritmos de FC utilizados en la técnica tradicional y en la técnica social-contextual (Se descarta medir el MAE para el FC utilizado en la técnica contextual de forma convencional, debido a las restricciones que implican tanto los pesos que cada usuario define para cada dimensión contextual, como el mismo contexto en los scores calculados de acuerdo a los estados contextuales que se experimentan).

**Tabla 24.** Plan de pruebas – interés.

<b>MAE a través de la comparación entre los scores (consumo) y los ratings (rating predichos por algoritmos de FC)</b>
<b>Plan de calidad:</b> permite evaluar y comparar la precisión de los algoritmos de filtrado colaborativo implementados respecto al consumo real del usuario.
<b>Procedimiento</b>
En el presente proyecto, se seleccionan las aplicaciones que vienen siendo consumidas al menos por 30 días antes a este cálculo (punto donde se espera que el score alcance su valor máximo al involucrar la mayor cantidad de eventos para la inferencia del mismo), aplicándoles tanto el algoritmo de FC tradicional como el FC basado en el factor de similitud social (Capítulo 5). Posteriormente se infiere el MAE por cada técnica propuesta (excepto la técnica contextual) de acuerdo a la (Ecuación 32). Al igual que las otras métricas descritas, este cálculo también se encuentra implementado en el módulo de generación de reportes mencionado.

#### 6.2.3.1. Resultados

**Tabla 25.** MAE algoritmos de FC tradicional y basado en similitud social.

Técnica	Nivel de MAE
Algoritmo de FC tradicional	0,6784
Algoritmo de FC basado en el factor de similitud social	0,46525

De acuerdo a los resultados expuestos en la Tabla 25. El MAE es menor para predicciones (ratings) realizadas con el algoritmo de FC basado en factor de similitud social, lo que se define como un punto a favor para la propuesta principal de este proyecto. Sin embargo, como ya se ha dicho, el score o calificación inferida a partir del consumo, puede variar con el tiempo, por lo que es necesario llevar un constante seguimiento para verificar que esta tendencia se mantenga.

## 6.2.4. Plan de pruebas – Disponibilidad

Con el fin de medir la fiabilidad y disponibilidad de las tecnologías y recursos utilizados (Hardware y software), este plan de pruebas incluye información relacionada con estadísticas brindadas por el servidor. El servidor utilizado para la implementación del prototipo cuenta con las siguientes características.

**Tabla 26.** Especificaciones del servidor utilizado.

<b>Microprocesador</b>	Intel Pentium Xeon 3,20GHz
<b>Caché del procesador</b>	8MB de caché de nivel 3
<b>Memoria</b>	8192MB
<b>Disco duro</b>	320GB (5400 RPM)
<b>Sistema Operativo</b>	Debian Squeeze

**Tabla 27.** Plan de pruebas – Disponibilidad.

<b>Disponibilidad</b>
<b>Plan de calidad:</b> permite evaluar y determinar si las características hardware y software de los recursos utilizados, cumplen con los requerimientos del sistema en un entorno de producción.
<b>Procedimiento</b>
Capturar los datos y reportes expuestos en las interfaces gráficas de la consola del servidor.

### 6.2.4.1. Resultados

Tráfico	Ø por hora		Conexiones	Ø por hora		%
Recibido	7.5 GB	10.6 MB	Número máx. de conexiones concurrentes	42	---	---
Enviado	40.7 GB	57.6 MB	Intentos fallidos	1	1.38 m	0.00%
Total	48.2 GB	68.2 MB	Abortado	8,989	12.43	0.19%
			Total	4,704 k	6,503.84	100.00%

**Figura 20.** Estadísticas generales de la base de datos central.

Uno de los problemas detectados, es la cantidad de intentos abortados en el establecimiento de conexiones con la base de datos. El análisis de este fenómeno, hizo necesaria la consulta a un experto, ya que no se encontró ningún problema en cuanto a la implementación realizada. Este afirmó que dichos abortos se presentan con mucha frecuencia en motores de bases de datos libres, ya que no cuentan con las mismas capacidades ni fiabilidad de motores privativos como el de Oracle, por ejemplo. Sin embargo por las limitaciones en recursos y dinero, no fue posible trabajar con un motor de ese tipo.

Las principales incidencias de este fenómeno, es la pérdida de información así sea en cantidades mínimas, lo cual puede repercutir en la precisión de las recomendaciones y resultados a largo plazo.

### 6.2.5. Plan de pruebas – Datos estadísticos complementarios

Gracias al tipo de data-set con el que se cuenta, es posible determinar otras medidas estadísticas con el fin de evidenciar el comportamiento de los usuarios del sistema en términos de frecuencia de uso promedio diaria de aplicaciones, categorías más consumidas, entre otros.

**Tabla 28.** Plan de pruebas – Datos estadísticos complementarios.

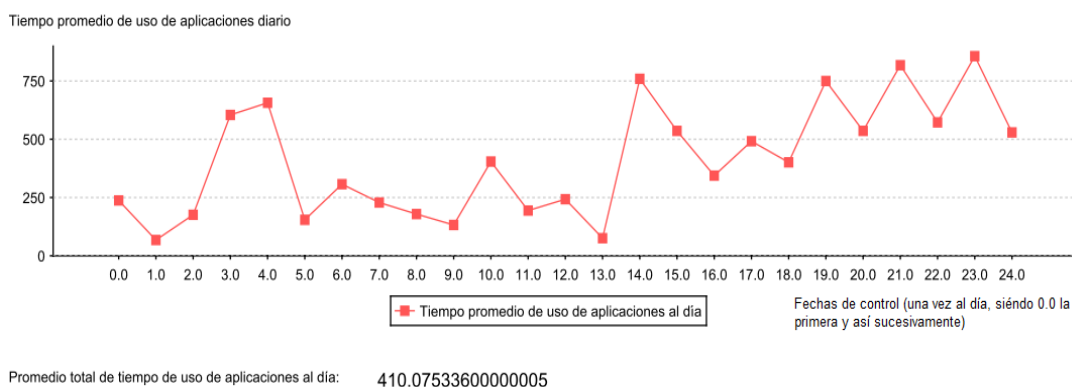
<b>Disponibilidad</b>
<b>Plan de calidad:</b> permite evaluar y determinar las características del grupo de personas que actualmente utilizan el sistema.
<b>Procedimiento</b>
Ejecutar los reportes respectivos, implementados en el <i>módulo de generación de reportes</i> desarrollado.

#### 6.2.5.1. Resultados preliminares

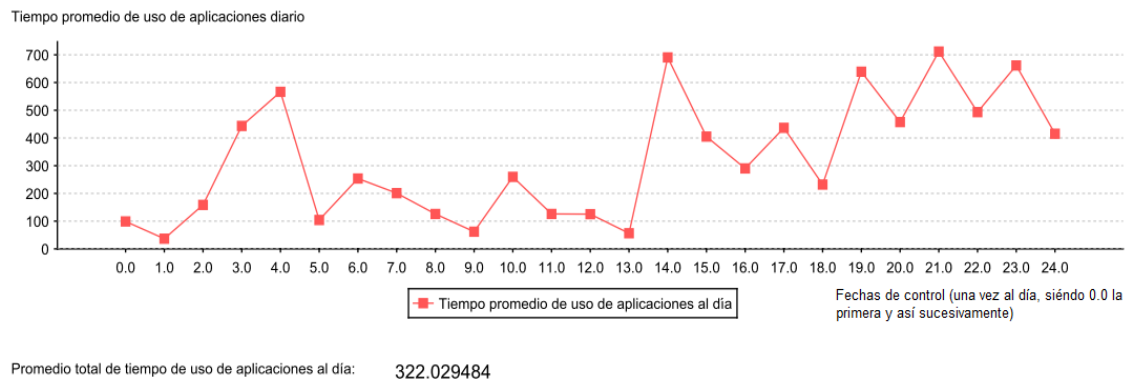
Como se ha especificado en el plan de pruebas correspondiente a esta parte, el objetivo es generar datos estadísticos que evidencien las tendencias y el comportamiento (características) del grupo de personas que utilizan o han utilizado el sistema, con el objetivo de soportar algunas conclusiones y trabajos futuros, planteados posteriormente.

##### Tiempo de uso promedio a día

Esta medida representa el promedio de tiempo que los usuarios registrados consumen aplicaciones al día (incluidas las aplicaciones nativas de los SmartPhones, como la gestión de contactos, agenda, ajustes, etc.).



**Figura 21.** Tiempo promedio de uso de aplicaciones (en segundos) por día (incluyendo uso de aplicaciones móviles nativas como gestión de contactos, agenda, ajustes, etc.).

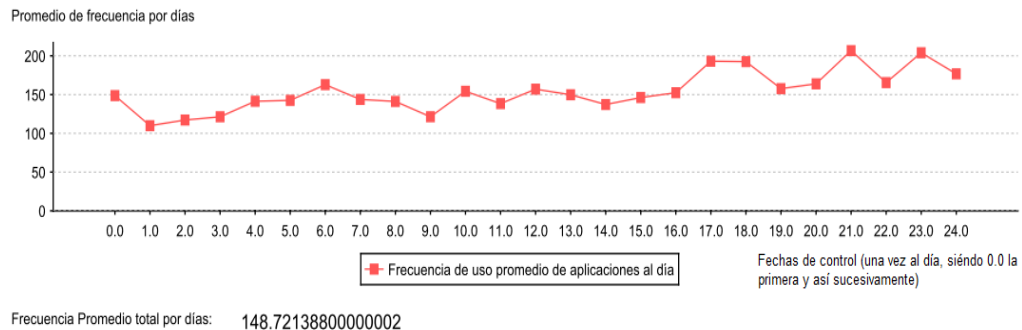


**Figura 22.** Tiempo promedio de uso de aplicaciones (en segundos), diariamente (sin incluir aplicaciones nativas).

Tal y como lo sugieren los resultados presentados en las Figuras 21 y 22. El tiempo promedio de consumo de aplicaciones móviles al día por parte de los usuarios de Vanilla, es considerablemente bajo (entre 6 y 7 minutos al día) respecto a estadísticas de indican un promedio diario de 81 minutos<sup>24</sup>.

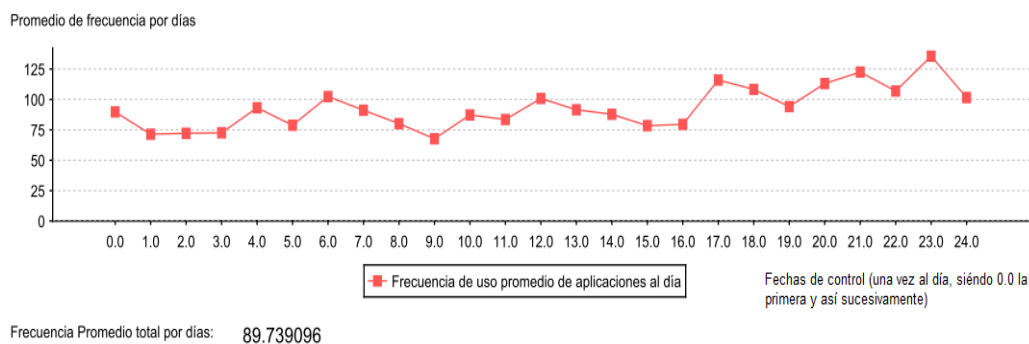
### Frecuencia promedio de uso de aplicaciones al día

Medida que evidencia el número de veces promedio al día, en que los usuarios inician o utilizan una aplicación, (incluidas las aplicaciones nativas de los SmartPhones, como la gestión de contactos, agenda, ajustes, etc.).



**Figura 23.** Frecuencia promedio de uso de aplicaciones, diariamente (incluyendo uso de aplicaciones nativas como gestión de contactos, agenda, ajustes, etc.)

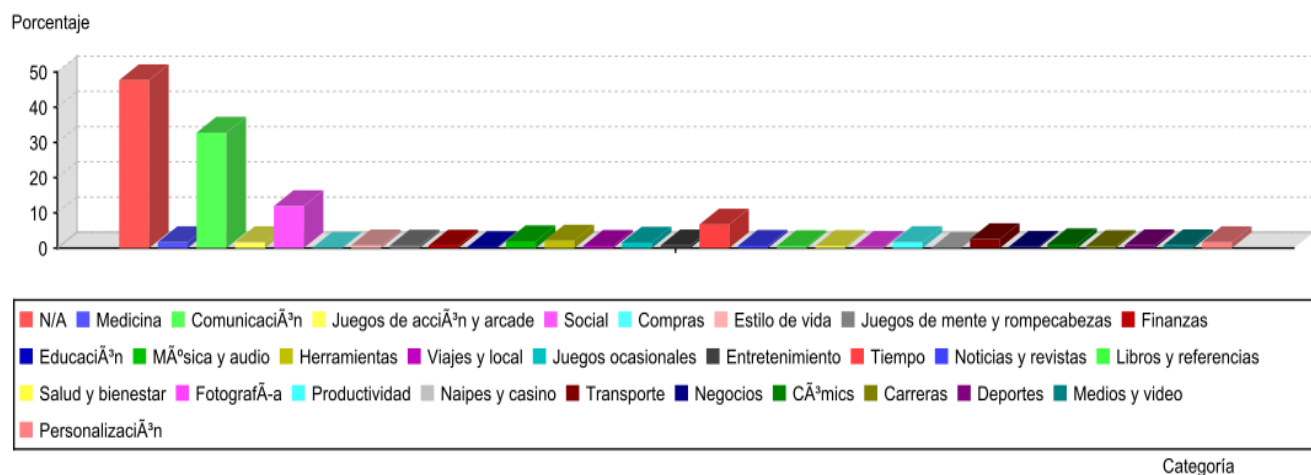
<sup>24</sup> <http://www.gigle.net/el-tiempo-dedicado-a-las-aplicaciones-moviles-supera-al-tiempo-de-consumo-en-la-web/>



**Figura 24.** Frecuencia promedio de uso de aplicaciones, diariamente (sin incluir aplicaciones nativas).

Como puede observarse en las Figuras 23 y 24. El número de veces promedio que los usuarios inician o usan aplicaciones, es considerablemente alto respecto al tiempo de uso total, lo que induce a suponer inicialmente, que la mayoría de aplicaciones consumidas son de *uso frecuente*.

### Categorías más consumidas por los usuarios



**Figura 25.** Porcentaje promedio de categorías consumidas.

Cómo puede apreciarse en la gráfica de arriba, el consumo de aplicaciones de comunicación, es muy superior frente al consumo detectado en otras categorías. Del mismo modo, el consumo de aplicaciones nativas (N/A) es un referente respecto al bajo consumo de los usuarios en cuanto a diversidad se refiere.

### 6.3. Resumen

El establecimiento del experimento, los sujetos experimentales, las métricas, el plan de pruebas, etc. Resume el objetivo de lo expuesto en este capítulo correspondiente a identificar y medir la calidad del sistema de la forma más precisa y coherente posible, el cual también incluye la validación de las hipótesis propuestas que respalden o no, dicha calidad.

**Tabla 29.** Validación hipótesis.

Hipótesis	Validación
<p><b>El uso de calificaciones inferidas teniendo en cuenta el consumo y el contexto físico del usuario generan una mejor calidad en los ARS</b></p>	<p><b>H1a:</b> La validez de esta hipótesis puede justificarse desde varias perspectivas: en primer lugar, el cálculo automático de calificaciones, definitivamente mejora la experiencia que el usuario tiene con la aplicación, al ahorrarle el trabajo de calificar manualmente las aplicaciones que este consume (Yan, y otros, 2011). En segundo lugar, el intento de mitigar la dispersión en las calificaciones de las aplicaciones, lógicamente sugiere recomendaciones más precisas (Ekstrand, y otros, 2011), sin embargo esta hipótesis aún no puede validarse desde esta perspectiva, debido al bajo consumo de aplicaciones móviles en general, por parte de los usuarios de Vanilla.</p>
<p><b>Otorgar una mayor relevancia a la apreciación de usuarios expertos por cada categoría de aplicaciones dentro de la red social del usuario, incrementa la calidad de las recomendaciones de los ARS basados en FC.</b></p>	<p><b>H2a:</b> La eficacia es una medida que indica la relevancia de las recomendaciones que se proveen. En este sentido, dicha relevancia será determinada tanto por el descubrimiento de aplicaciones móviles (selección de aplicaciones móviles de entrada al algoritmo de FC), como por los algoritmos de FC. Consecuentemente, los resultados relacionados a la eficacia medida en las diferentes técnicas de recomendación, son favorables respecto a la técnica social-contextual dándole, validez a la hipótesis en cuestión.</p>

## Capítulo VII, Conclusiones y trabajos Futuros

En la actualidad existe un crecimiento continuo del número de aplicaciones móviles, incentivado por la fuerte competencia en este entorno y las posibilidades de crecimiento, motivadas por el aumento acelerado en el número de usuarios poseedores de Smartphones; lo cual crea un ecosistema propicio para el desarrollo y consumo de aplicaciones. En consecuencia, el acelerado crecimiento del número de aplicaciones móviles disponibles, dificulta cada vez más la identificación de aplicaciones de interés y acordes a las necesidades y preferencias del usuario en situaciones particulares.

Con el fin de presentar una solución a este problema, y a partir de trabajos relacionados que expusieron premisas importantes en esta área, fue elaborado el presente trabajo de grado que consta de un sistema de recomendación, que permite a los usuarios encontrar aplicaciones móviles de una forma simple y pertinente, teniendo en cuenta factores como su contexto actual, sus preferencias (representadas a través de su consumo), su red social, entre otros.

Así, este capítulo describe inicialmente las principales contribuciones del trabajo realizado, posteriormente presenta las conclusiones obtenidas, y finalmente se propone algunos trabajos futuros.

### 7.1. Contribuciones

Entre las principales contribuciones de este proyecto de grado se desatacan las siguientes:

- La definición de un modelo de recolección y clasificación de información en entornos móviles, compuesto por la descripción de las fuentes y tipos de información. Este modelo puede extenderse a diferentes áreas de investigación.
- La definición de un modelo contextual que describe la situación temporal de un usuario mediante el uso de dimensiones y estados contextuales. Este modelo fue aplicado para describir el contexto del usuario y generar la recomendación de aplicaciones móviles propuesta en este trabajo. Sin embargo, puede extenderse a otras áreas de investigación. (Objetivo específico 1)
- La definición de técnicas y mecanismos para calificar aplicaciones móviles en función de su *consumo* y del contexto del actual del usuario. (Objetivo específico 2)
- La definición de *usuarios expertos* para optimizar el descubrimiento de aplicaciones móviles y la vecindad en el filtrado colaborativo, los cuales se definen como el grupo de personas pertenecientes a la red social del usuario cuyo patrón de consumo se inclina a una categoría de aplicaciones. De esta manera pueden existir contactos expertos en aplicaciones de juegos, entretenimiento, estilo de vida, etc. (Objetivo específico 2)
- La definición de las técnicas basadas en el Filtrado Colaborativo, las cuales generan una recomendación integral: Tradicional basada en consumo, Contextual y Social-Contextual. (Objetivo específico 2)



- El desarrollo de un Framework para la recolección de información en la plataforma Android, el cual puede ser utilizado en futuros proyectos que consideren el ámbito contextual. Este Framework se diseñó con el fin de que sea flexible según las necesidades de quien desee utilizarlo. (Objetivos específicos 1 y 3)
- El prototipo funcional para la recomendación de aplicaciones móviles Vanilla, en el cual se implementan todas las técnicas y algoritmos definidos en esta propuesta. (Objetivos específicos 2 y 3)
- Un amplio data-set, que incluye: i) la información de comportamiento de usuarios a nivel de consumo de aplicaciones móviles y de los contextos físicos que experimentan y ii) Una red social basada en contactos telefónicos, llamadas y SMS. (Objetivo específico 3)
- Incursionar en el desarrollo y la investigación sobre sistemas de recomendación en la comunidad académica de la Universidad del Cauca, implementando un prototipo de recomendación de aplicaciones móviles el cual crea un precedente alrededor de éste tema, y despertar así el interés por estas tecnologías en la comunidad investigativa del país, debido al positivo panorama social y económico factible de obtener con su difusión. Además, a partir del proceso seguido en este proyecto de investigación, se logró fundamentar la nueva materia electiva del departamento de telemática “Web Social”.
- Como contribución final se destaca el aporte realizado del presente trabajo, al trabajo de grado de maestría “Recomendación de aplicaciones móviles basada en el análisis del comportamiento contextual del usuario” del Ingeniero Luis Antonio Rojas, desarrollada al interior de la Universidad del Cauca. De esta manera se contribuye en la investigación en torno a la temática de sistemas de recomendación y entornos móviles, incentivando futuros trabajos dentro del Grupo de Ingeniería Telemática (GIT) de la Universidad del Cauca, los cuales integren las líneas de investigación relacionadas con sistemas de recomendación, aplicaciones móviles, redes sociales y contexto.

## 7.2. Conclusiones

Dentro del proyecto, se ha abordado una propuesta de filtrado colaborativo (que incluye los factores como contexto, consumo e información social de los usuarios, como ha sido mencionado anteriormente), para la predicción del rating de las aplicaciones no consumidas por el usuario. Para esto se propuso tres técnica: tradicional, contextual y la social-contextual.

Al finalizar este proyecto se entregó un prototipo denominado Vanilla, el cual se basa en las técnicas y algoritmos mencionadas al interior de este documento, permitiendo así la obtención de una recomendación de aplicaciones móviles basadas en el contexto y en la información social. Este prototipo permite además personalizar la recomendación, con el establecimiento de los pesos y con la configuración de muestreo de cada dimensión contextual, lo cual genera una recomendación más dinámica y robusta.

A continuación se describen las principales conclusiones obtenidas a partir de la ejecución del presente proyecto sobre la muestra planteada en el capítulo 8.

- El bajo promedio de consumo de aplicaciones móviles de los usuarios de Vanilla (6 a 7 minutos al día), frente al promedio mundial (81 minutos al día), sugiere que existen

grupos de personas o mercados que pueden comportarse de mejor forma frente al sistema de recomendación propuesto, al ofrecer mucha más información.

- Debido al estancamiento del número de descargas de la aplicación, se genera lo que se denomina “efecto burbuja”, el cual consiste en alcanzar un estado en el que no es posible recomendar nuevas aplicaciones a los usuarios existentes, al no contar con retroalimentación de nuevos usuarios, y por ende de nuevas aplicaciones.
- Dicho estancamiento, acompañado por el bajo promedio de consumo de los usuarios de Vanilla mencionado en la primera conclusión, evidencia la falta de madurez del dataset recolectado hasta el momento. Aun así, los resultados obtenidos, muestran tendencias favorables en pro de dar validez a las hipótesis formuladas y de afirmar una mejoría del sistema en cuestión.
- La red social generada a partir de contactos telefónicos, llamadas y SMS, sufre un alto grado de dispersión (falta de información), debido a la existencia de aplicaciones móviles que suplen necesidades de comunicación como WhatsApp, Line, Skype, Viper, entre otras. Esto evidencia la necesidad de buscar redes sociales alternativas que permitan abstraer la información necesaria para optimizar el sistema de recomendación, teniendo en cuenta que las aplicaciones mencionadas, restringen el acceso a su información.
- Observando el fenómeno mencionado anteriormente, es posible corroborar que en Colombia el principal propósito del dispositivo móvil, aún se limita a satisfacer necesidades de comunicación, dejando en un segundo plano el consumo de aplicaciones móviles (diferentes a las de comunicación). Esto se respalda aún más con las estadísticas que demuestran que el tiempo promedio de consumo de aplicaciones móviles al día de la población de prueba, está entre 6 y 7 minutos (frente al promedio mundial de 81 minutos al día), y en donde la mayoría de aplicaciones corresponden a Apps de comunicación con un 35% (teniendo en cuenta que el 50% de las apps consumidas, corresponden a las aplicaciones nativas del Smartphone).
- La alta segmentación de cada dimensión del contexto en estados contextuales (situaciones contextuales con un amplio número de componentes o estados), no necesariamente refleja un aumento en la precisión de la especialización en la recomendación de aplicaciones, ya que es mucho más probable que el nivel de dispersión en los datos, aumente (a mayor cantidad de estados, mayor número de posibles contextos que se experimentan, lo que significa una menor probabilidad de que dos usuarios experimenten el mismo o contextos muy parecidos).
- Los tiempos de respuesta del sistema de recomendación, se ven mejorados considerablemente con la utilización de procesos en background (Demonios), encargados de procesar la mayor cantidad de información posible, para exponerla de forma sencilla a los algoritmos de recomendación.
- En las pruebas de tiempo promedio de observación por página, se puede apreciar que en las páginas cero y uno, los usuarios permanecen más tiempo en comparación a la página dos. Esto obedece al orden establecido donde las aplicaciones móviles recomendadas con mayor rating, se muestran en las primeras páginas. Evidencia que da otro soporte, a al buen nivel de la relevancia de las recomendaciones que se dan.

- Sin embargo, teniendo en cuenta lo anterior, es posible afirmar que el nivel de interés es moderado teniendo en cuenta el poco tiempo empleado en la observación de la página 2. Esto puede deberse a que los usuarios, posiblemente, no están dispuestos a seguir explorando las páginas, o simplemente, a que no se cuenta aún, con la información de consumo suficiente para recomendar las 9 aplicaciones completas. De esta forma se propone inferir esta medida en el futuro donde la falta de información no sea un problema.
- Gracias a las pruebas de eficacia promedio realizadas hasta la fecha, se puede ver una tendencia en el aumento de selección de aplicaciones móviles recomendadas por la técnica social-contextual. Por lo tanto, según los resultados preliminares se puede determinar que es la mejor técnica para la recomendación de aplicaciones móviles en el sistema propuesto.
- De la misma forma, se da respaldo a los resultados que afirman que el consumo de aplicaciones móviles depende del contexto en situaciones particulares (Capítulo 1), gracias a que la eficacia de la técnica contextual se muestra levemente superior a la técnica tradicional.
- La aplicación Vanilla no es de uso intensivo, ni de uso prolongado, esto dado por que los usuarios solo la utilizan cuando la necesitan o cuando tengan curiosidad de que aplicación puedan usar en un momento determinado. Esto se puede apreciar en las pruebas de número de usuarios versus número de recomendación, en donde la tasa de usuarios disminuye considerablemente a medida que avanza el tiempo, pasando de una aplicación de novedad a una aplicación de uso esporádico.

### 7.3. Trabajos futuros

Este proyecto puede considerarse como un estudio inicial para la recomendación de aplicaciones móviles basada en la información social y contextual del usuario, que contribuye en la solución del problema de la optimización y especialización de la recomendación. De esta manera, en el campo de investigación del presente trabajo de grado, se proponen los siguientes trabajos futuros:

El modelo contextual y el análisis de la información social, planteados dentro del presente proyecto pueden ser evaluados y adaptados a otros dominios de aplicación, con el fin de proponer soluciones en distintos campos del conocimiento aparte de los sistemas de recomendación.

**Extender el soporte del sistema de recomendación a Tablets**, en la búsqueda de generar un escenario más adecuado respecto a la diversificación en el consumo de aplicaciones móviles, para la validación de la presente propuesta y para generación de resultados más coherentes y precisos. El alcance del mercado a tipo de dispositivos como Tablets, que pueden representar más y mejores fuentes de información de consumo.

**Extender el soporte a otros idiomas y regiones.** Las limitaciones presentadas por Google Play para el acceso de la información en otros idiomas y regiones, no han permitido que el sistema sea multilinguaje. Sin embargo se planea traducir la descripción de cada aplicación

móvil consumida en el sistema para múltiples lenguajes, lo que permitiría extender potencialmente el mercado y la oferta de aplicaciones.

**Extender el soporte del sistema de recomendación a otras plataformas móviles.** Se plantea la necesidad de migrar la aplicación a otras plataformas móviles, como pueden ser iOS, Windows Phone o B10; dado que cada plataforma refleja su propia filosofía, lo que se traduce en una experiencia diferente para el usuario donde el consumo puede ser más o menos intensivo. Esto con el fin de observar los efectos que puede generar el tipo del mercado o plataforma, sobre el sistema propuesto.

**Inferencia de la prioridad de las dimensiones contextuales por medio del análisis del consumo,** actualmente, el sistema requiere el ingreso de la prioridad de cada dimensión contextual de forma manual (factor que permite una recomendación personalizada a las necesidades y preferencias del usuario). in embargo, en pro de mejorar la experiencia que el usuario tiene con la aplicación, se desea determinar estos factores de forma automática mediante el análisis del consumo del usuario.

**Emplear redes sociales alternativas para el sistema de recomendación propuesto,** debido a que los cambios constantes en la manera como la gente interactúa en el tiempo. Se propone establecer una plataforma en donde los usuarios puedan desarrollar interacciones influyentes en el consumo de las aplicaciones móviles. En otras palabras, se propone el desarrollo de una nueva red social interacciones que signifiquen un alto grado de cercanía.

**Adaptación de las técnicas y algoritmos propuestos a sistemas de recomendación híbridos,** ya que el presente trabajo adapta técnicas y algoritmos de filtrado colaborativo basado en memoria debido a su popularidad y facilidad de implementación. Sin embargo, en pro de extender las capacidades del sistema, se propone el uso de algoritmos híbridos entre filtrado colaborativo basado en memoria y basado en modelo.

**Experimentación del Sistema de recomendación propuesto en un entorno de real exigente,** adoptando las estrategias debidas, se propone evaluar al sistema en un ambiente real y exigente para validar su desempeño y realizar una evaluación más precisa de la eficacia, predicción y recomendación.

**Generar estrategias comerciales y de difusión,** debido necesario aumentar la cantidad de información, se propone idear estrategias de difusión comercial, para incentivar la descarga de la aplicación y así aumentar el repositorio de aplicaciones, la información de consumo y social (potencialmente), en pro de reducir el “efecto burbuja”

**Mejorar las prestaciones de la capa de datos del sistema,** debido a la elevada tasa de conexiones abortadas a la base de datos central (0,33%, tasa supremamente alta, si se tiene en cuenta la cantidad de información registrada). Se propone trabajar con motores de bases de datos más robustos y potentes, siempre y cuándo se cuenten con los recursos necesarios, y la evolución de la aplicación sea positiva y así lo sugiera.

## Referencias

**Adamic L.** Social Network Analysis [Conference]. - 2012.

**Adomavicius Gediminas and Sankaranarayanan, Ramesh and Sen, Shahana and Tuzhilin, Alexander** Incorporating contextual information in recommender systems using a multidimensional approach [Article] // ACM Trans. Inf. Syst.. - 2005. - Vol. 23.

**Aharony Nadav [et al.]** The social fMRI: measuring, understanding, and designing social mechanisms in the real world [Conference]. - [s.l.] : ACM, 2011. - pp. 445-454.

**Aharony Nadav** Social fMRI : measuring and designing social mechanisms using mobile phones [Report] : Doctoral / Massachusetts Institute of Technology. - 2012. - p. 168.

**Android** index:Android [Online] // Android. - Google. - February 11, 2012. - <http://developer.android.com/index.html>.

**Appelquist D., Brickley D. and Carvahlo M.** Standards-based, Open and Privacy-aware Social Web [Conference]. - 2010.

**Baltrunas Linas** Context-Aware Collaborative Filtering Recommender Systems [Report] / University of Bolzano. - 2011.

**Barkley Brian** Using an Accelerometer Sensor to Measure Human Hand Motion [Report] / Massachusetts Institute of Technology. - 2000.

**Belkin N. J. and Croft** Information Filtering and Information Retrieval: Two Sides of the Same Coin? [Journal] // Communications of the ACM 35(12). - 1992. - pp. 29-38.

**Bellotti Victoria [et al.]** Activity-based serendipitous recommendations with the Magitti mobile leisure guide [Conference]. - [s.l.] : ACM, 2008. - pp. 1157-1166.

**Böhmer Matthias [et al.]** Falling asleep with Angry Birds, Facebook and Kindle: a large scale study on mobile application usage [Conference]. - [s.l.] : ACM, 2011. - pp. 47-56.

**Böhmer Matthias and Ganev Lyubomir** AppFunnel: A Framework for Usage-centric Evaluation of Recommender Systems that Suggest Mobile Applications [Conference] // Proceedings of IUI'13. Santa Clara 2013. - 2013.

**Böhmer Matthias, Bauer Gernot and Prinz Moritz** Contextualizing Mobile Applications for Context-aware Recommendation. [Conference]. - 2010.

**Burke R. D.** Hybrid Recommender Systems: Survey and Experiments [Journal]. - 2002. - p. 29.

**Burke Robin D., Felfernig Alexander and Gökler Mehmet H.** Recommender Systems: An Overview [Journal] // AI Magazine. - 2011. - 3 : Vol. 32. - pp. 13-18.

**Chawla Surbhi** Google Play Store reaches 700,000 apps landmark // Google Play Store reaches 700,000 apps landmark. - 2012.

**Chen A.** Context-Aware Collaborative Filtering System: Predicting the User's Preference in the Ubiquitous Computing Environment [Journal]. - 2005. - pp. 75-81.

**Cheng-Yu Shih-Fu Chang** Color-Mood Analysis of Films Based on Syntactic and Psychological Models [Journal] // Multimedia and Expo, 2004. ICME '04.. - 2004. - Vol. 2. - pp. 831-834.

**Church Karen and Smyth Barry** Understanding mobile information needs [Conference]. - [s.l.] : ACM, 2008. - pp. 493-494.

**Condliff M. K. [et al.]** Bayesian Mixed-Effects Models for Recommender Systems [Conference]. - 1999.

**Demumieux Rachel and Losquin Patrick** Gather customer's real usage on mobile phones [Conference]. - [s.l.] : ACM, 2005. - pp. 267-270.

**Dey Anind K.** Understanding and Using Context [Article] // Personal Ubiquitous Comput.. - 2001. - Vol. 5.

**Ekstrand M. D., Riedl J. T. and Konstan J. A.** Collaborative Filtering Recommender Systems [Conference]. - 2011.

**Falaki Hossein [et al.]** Diversity in smartphone usage [Conference]. - [s.l.] : ACM, 2010. - pp. 179-194.

**Falaki Hossein, Mahajan Ratul and Estrin Deborah** SystemSens: a tool for monitoring usage in smartphone research deployments [Conference]. - [s.l.] : ACM, 2011. - pp. 25-30.

**Foltz P. W.** Using Latent Semantic Indexing for Information Filtering [Journal] // R. B. Allen (ed.): Proceedings of the Conference on Office Information Systems, Cambridge, MA. - 1990. - pp. 40-47.

**Froehlich Jon [et al.]** MyExperience: a system for in situ tracing and capturing of user feedback on mobile phones [Conference]. - [s.l.] : ACM, 2007. - pp. 57-70.

**G. Adomavicius R. Sankaranarayanan, S. Sen, and A. Tuzhilin** Incorporating contextual information in recommender systems using a multidimensional approach [Journal] // ACM Transactions on Information Systems. - 2005. - 23. - pp. 103-145.

**Galán S. M.** Filtrado Colaborativo y Sistemas de Recomendación [Conference]. - 2007.

**Gartner** Market Share Analysis: Mobile Devices, Worldwide, 2Q12 // Market Share Analysis: Mobile Devices, Worldwide, 2Q12. - 2012.

**Girardello Andrea and Michahelles Florian** AppAware: which mobile applications are hot? [Conference]. - [s.l.] : ACM, 2010. - pp. 431-434.

**Golbeck J. and Hendler J.** FilmTrust: Movie Recommendations using Trust in Web-based Social Networks [Conference]. - 2006.

**Google Play Store** store:Google Play Store [Online] // Google Play Store. - Google. - March 18, 2012. - <https://play.google.com/store>.

**Hayes C. [et al.]** An online evaluation framework for recommender systems [Conference]. - 2002.

**Herlocker J.L., Konstan J.A. and Riedl J.** Explaining Collaborative Filtering Recommendations [Conference]. - 2000.

**Hogg T., Wikinson D. M. and Szabo, M. J.** Multiple Relationship Types in Online Communities and Social Networks [Conference]. - 2008.

**Horozov Tzvetan, Narasimhan Nitya and Vasudevan Venu** Using Location for Personalized POI Recommendations in Mobile Environments [Conference]. - [s.l.] : IEEE Computer Society, 2006. - pp. 124-129.

**Hua Tomonori Aoyama** A Context-aware Collaborative Filtering Algorithm for Real World Oriented Content Delivery Service [Journal] // Ubicomp Metapolis and Urban Life. - 2005.

**Ipsos MediaCT - The Media Content and Technology Research Specialists** Our Mobile Planet: Global Smartphone Users. - [s.l.] : Google, 2012.

**Jennings A. and Higuchi H.** ‘A User Model Neural Network for a Personal News Service. [Conference]. - 1993.

**Jung J. J.** Contextualized mobile recommendation service based on interactive social network discovered from mobile users [Journal]. - [s.l.] : Expert System with Applications, 2009. - 36. - pp. 11950-11956.

**Karatzoglou A. [et al.]** Climbing the App Wall: Enabling Mobile App Discovery through Context-Aware Recommendations [Conference]. - 2012.

**Kramer Ronny, Modsching Marko and Hagen Klaus ten** Field study on methods for elicitation of preferences using a mobile digital assistant for a dynamic tour guide [Conference]. - [s.l.] : ACM, 2006. - pp. 997-1001.

**Krulwich B.** Lifestyle Finder: Intelligent User Profiling Using Large-Scale Demographic Data [Journal] // Artificial Intelligence Magazine. - 1997. - Vol. 18. - pp. 37-45.

**Liu Fengkun and Lee Hong Joo** Use of social network information to enhance collaborative filtering performance [Journal] // Expert Systems with Applications . - 2010. - 7 : Vol. 37. - pp. 4772-4778.

**López Diego M and Blobel, Bernd and Gonzalez, Carolina** Quality evaluation of health information system's architectures developed using the HIS-DF methodology [Book]. - [s.l.] : Studies in health technology and informatics, 2010. - Vol. 489 : p. 21.

**Luis Rojas-Potosi Luis Suarez-Meza, Leandro Ordoñez-Ante, Juan Corrales** Composite social network for predicting mobile apps Installation [Conference] // AAAI Spring Symposium Series. - AAAI : [s.n.], 2012.

**Ma Haiping [et al.]** A habit mining approach for discovering similar mobile users [Conference]. - [s.l.] : ACM, 2012. - pp. 231-240.

**Maes Pattie** Agents that reduce work and information overload [Journal] // Commun. ACM. - [s.l.] : ACM, 1994. - 7 : Vol. 37. - pp. 30-40.

**Mayer JD Salovey P, Gomberg-Kaufman S, Blainey K** A broader conception of mood experience [Journal] // Journal of Personality and Social Psychology. - 1991. - Vol. 60. - pp. 100-111.

**McKechinie** Webster's New Twentieth Century Dictionary of the English Language [Book] / ed. Schuster Simon and. - [s.l.] : J. L., 1989. - 2nd edition. - Number 978-0671418199.

**MediaCT Content Ipsos and Specialists Technology Research** Our Mobile Planet: Global Smartphone Users [Report] / Google. - 2012.

**Melville P. and Sindhvani V.** Recommender Systems [Conference]. - 2010.

**Nguyen Quang Nhat and Ricci Francesco** Replaying live-user interactions in the off-line evaluation of critique-based mobile recommendations [Conference]. - [s.l.] : ACM, 2007. - pp. 81-88.

**Pabjan B.** Measuring the Social Relations: Social Distance in Social Structure [Conference]. - 2005.

**Pan Wei, Aharony Nadav and Pentland Alex** Composite Social Network for Predicting Mobile Apps Installation. [Conference] / ed. Burgard Wolfram and Roth Dan. - [s.l.] : AAAI Press, 2011.

**Park Moon-Hee, Hong Jin-Hyuk and Cho Sung-Bae** Location-Based Recommendation System Using Bayesian User's Preference Model in Mobile Devices [Book Section] // Ubiquitous Intelligence and Computing / ed. Indulska Jadwiga [et al.]. - [s.l.] : Springer Berlin Heidelberg, 2007. - Vol. 4611.

**Pennok D. M., Horvits E. and Giles C. L.** Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering [Conference]. - 2000.

**Resnick P. and H R. Varian,** Recommender Systems [Conference]. - 1997.

**Ricci F. and Baltrunas L.** Context-based splitting of item ratings in collaborative filtering [Journal]. - 2009. - pp. 245-248.

**Ricci Francesco, Rokach Lior and Shapira Bracha** Introduction to Recommender Systems Handbook [Book Section] // Recommender Systems Handbook / ed. Ricci Francesco [et al.]. - [s.l.] : Springer US, 2011.

**Ricci René T. A. Wietsma and Francesco** Product Reviews in Mobile Decision Aid Systems // PERMID. - 2005.



**Ruth E. Mayagoitia Anand V. Nene, Peter H. Veltink** Accelerometer and rate gyroscope measurement of kinematics: an inexpensive alternative to optical motion analysis systems [Journal] // Journal of Biomechanics. - 2002. - Vol. 35. - pp. 537-542.

**Sami Rahul and Harmala Mike** SI 583 - Recommender Systems Course // SI 583 - Recommender Systems Course. - 2009. - Graduate Course of Recommender systems.

**Schafer J. B., Konstan J. and Riedl J.** 'Recommender Systems in E-Commerce [Conference]. - 1999.

**Schilit B. and Theimer, M.** Disseminating active map information to mobile hosts [Journal]. - [s.l.] : IEEE Network, 1994. - 8. - pp. 22-32.

**Schilit B., Adams, N. and Want, R.** Context-aware computing applications [Journal]. - [s.l.] : Mobile Computing Systems and Applications, 1994. - pp. 85-90.

**Sensors Overview** Sensors Overview [Online]. - Google. - February 11, 2012. - [http://developer.android.com/guide/topics/sensors/sensors\\_overview.html](http://developer.android.com/guide/topics/sensors/sensors_overview.html).

**Serrano Carlos Enrique** Modelo integral para el profesional en ingeniería [Book]. - Popayán : [s.n.], 2005.

**Setten Mark, Pokraev Stanislav and Koolwaaij Johan** Context-Aware Recommendations in the Mobile Tourist Application COMPASS [Book Section] // Adaptive Hypermedia and Adaptive Web-Based Systems / ed. Bra PaulM.E. and Nejd Wolfgang. - [s.l.] : Springer Berlin Heidelberg, 2004. - Vol. 3137.

**Skraba R. [et al.]** Developing Compelling Social-Enabled Applications with Context-based Social Interaction Analysis [Journal]. - 2009. - pp. 206-211.

**Su Xiaotuan and Khoshgoftaar Taghi M.** A Survey of Collaborative Filtering Techniques [Conference]. - 2009.

**Terveen L. and Hill W.** Human-Computer Collaboration in Recommender Systems [Conference]. - 2001.

**Veltink Peter H.** Accelerometer and rate gyroscope measurement of kinematics: an inexpensive alternative to optical motion analysis systems [Journal] // Journal of Biomechanics. - 2002. - Vol. 35. - pp. 537-542.

**Verkasalo Hannu** Contextual patterns in mobile service usage [Journal] // Personal and Ubiquitous Computing. - [s.l.] : Springer, 2009. - 5 : Vol. 13. - pp. 331-342.

**Viswanath P. [et al.]** You are who you know: inferring user profiles in online social networks [Journal]. - 2010. - pp. 251-260.

**Weather Yahoo** Yahoo! Weather RSS Feed [Online]. - Yahoo. - April 8, 2012.

**Whitney Lance** Google: 200 million Android devices now active worldwide // Google: 200 million Android devices now active worldwide. - 2011.

**Woemdl Wolfgang and Schulze Florian** Capturing, Analyzing and Utilizing Context-Based Information about User Activities on Smartphones [Conference]. - 2011.

**Woerndl Wolfgang and Schueller, Christian and Wojtech, Rolf** A Hybrid Recommender System for Context-aware Recommendations of Mobile Applications // Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop. - 2007.

**Xu Qiang [et al.]** Identifying diverse usage behaviors of smartphone apps [Conference]. - [s.l.] : ACM, 2011. - pp. 329-344.

**Yan Bo and Chen Guanling** AppJoy: personalized mobile application discovery [Conference]. - [s.l.] : ACM, 2011. - pp. 113-126.

**Yan Tingxin [et al.]** Fast app launching for mobile devices using predictive user context [Conference]. - [s.l.] : ACM, 2012. - pp. 113-126.

**Yuan Soe-Tsyr and Tsao Y. W.** A recommendation mechanism for contextualized mobile advertising [Journal] // Expert Syst. Appl.. - 2003. - 4 : Vol. 24. - pp. 399-414.