

Sistemas de Recomendaciones en un entorno turístico móvil soportado en interacción contextual con NFC



Trabajo de Grado de Maestría

Ing. Eliana Andrea Concha Agredo

Director: PhD. Ing. Gustavo Adolfo Ramírez González

Asesor: Msc. Ing. Francisco Martínez

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Departamento de Telemática

Línea de Investigación en Aplicaciones sobre Internet

Popayán, Febrero de 2016

CONTENIDO

| | Pág. |
|--|-------------|
| 1 Introducción | 1 |
| 1.1 Contexto de Investigación | 2 |
| 1.2 Planteamiento del Problema | 3 |
| 1.3 Objetivos | 4 |
| 1.3.1 Objetivo General | 4 |
| 1.3.2 Objetivos Específicos | 4 |
| 1.4 Alcance | 4 |
| 1.5 Contribuciones | 5 |
| 1.6 Contenido de la Monografía | 5 |
| 1.6.1 Parte I. Introducción | 5 |
| 1.6.2 Parte II. Propuesta | 6 |
| 1.6.3 Parte III. Evaluación y Resultados | 6 |
| 2 Estado del Actual de Conocimiento | 7 |
| 2.1 Contexto general | 7 |
| 2.1.1 Contexto del usuario | 7 |
| 2.1.2 Sistemas de Recomendación | 10 |
| 2.1.3 NFC | 14 |
| 2.1.4 Aprendizaje Automático | 17 |

| | | |
|----------|---|-----------|
| 2.1.5 | SR Basados en conocimiento u ontologías | 22 |
| 2.2 | Trabajos Relacionados | 27 |
| 2.2.1 | Guías de turismo móviles sensibles al contexto | 27 |
| 2.2.2 | Sistemas de Recomendaciones para el entorno turístico móvil | 28 |
| 2.2.3 | Alternativas de obtención de información contextual basada en ubicación de los usuarios móviles | 29 |
| 2.2.4 | Aproximaciones de recomendación basadas en conocimiento y sensibles al contexto | 30 |
| 3 | Diseño del Modelo Contextual | 33 |
| 3.1 | Diseño | 33 |
| 3.1.1 | Modelado de datos en el contexto | 34 |
| 3.1.2 | Modelo Contextual | 34 |
| 3.2 | Resumen | 46 |
| 4 | Enfoque del Sistema de Recomendaciones Propuesto | 47 |
| 4.1 | Diseño del Modelo de Aplicación | 48 |
| 4.1.1 | Propósito del motor de recomendaciones: | 48 |
| 4.1.2 | Tipos de recomendaciones: | 48 |
| 4.1.3 | Integración con el contenido de navegación: | 49 |
| 4.1.4 | Métricas de rendimiento: | 49 |
| 4.1.5 | Únicos o múltiples dispositivos: | 50 |
| 4.1.6 | Únicos o múltiples usuarios: | 50 |
| 4.1.7 | Infraestructura de la aplicación: | 51 |
| 4.2 | Inclusión de contexto en Sistemas de Recomendaciones | 51 |

CONTENIDO

| | | |
|-------|---|----|
| 4.2.1 | Paradigmas para incluir contexto en sistemas de recomendaciones | 52 |
| 4.3 | Métodos de Hibridación | 54 |
| 4.3.1 | Weighted | 54 |
| 4.3.2 | Switching | 55 |
| 4.3.3 | Mixto | 55 |
| 4.3.4 | Combinación de características | 55 |
| 4.3.5 | Cascada | 55 |
| 4.3.6 | Característica de Aumento | 55 |
| 4.3.7 | Meta nivel | 56 |
| 4.4 | Modelo de Aplicación | 56 |
| 4.4.1 | Propósito del motor de recomendación | 57 |
| 4.4.2 | Tipo de recomendaciones | 57 |
| 4.4.3 | Uno o varios dispositivos | 57 |
| 4.4.4 | Usuarios individuales o múltiples | 57 |
| 4.4.5 | Métrica de rendimiento | 57 |
| 4.4.6 | Infraestructura de aplicación | 57 |
| 4.5 | Fases del enfoque propuesto | 57 |
| 4.5.1 | Inferencia de Perfiles Contextuales | 58 |
| 4.5.2 | Recomendación basada en Conocimiento | 67 |
| 4.5.3 | Recomendación por Filtrado Colaborativo | 75 |
| 4.5.4 | Algoritmo de Maximización de consumo de recursos turísticos en un tiempo limitado . | 89 |
| 4.5.5 | Priorización por patrones de consumo | 91 |
| 4.5.6 | Optimización de Trayectoria | 91 |

| | | |
|----------|--|------------|
| 4.6 | Resumen | 94 |
| 5 | Implementación del Sistema Propuesto | 95 |
| 5.1 | Implementación de la solución | 98 |
| 5.1.1 | Smart Tags | 98 |
| 5.1.2 | Cliente Móvil | 99 |
| 5.1.3 | Servidor | 100 |
| 5.2 | Resumen | 106 |
| 6 | Experimentación y Resultados | 107 |
| 6.1 | Metodología | 107 |
| 6.1.1 | Planeación del experimento | 107 |
| 6.1.2 | Métricas de Evaluación | 109 |
| 6.2 | Plan de ejecución de pruebas | 111 |
| 6.2.1 | Plan de Ejecución de Pruebas: Métricas | 112 |
| 6.3 | Resultados de Evaluación | 113 |
| 6.3.1 | Resultados de Evaluación de Métricas | 114 |
| 6.3.2 | Resultados de Evaluación de aceptación del sistema | 119 |
| 6.4 | Resumen | 126 |
| 7 | Conclusiones y Trabajo Futuro | 129 |
| 7.1 | Conclusiones | 129 |
| 7.2 | Trabajos Futuros | 131 |

CONTENIDO

Bibliografía

133

LISTA DE TABLAS

| | Pág. |
|---|-------------|
| 2.1 Categorías de métodos híbridos | 14 |
| 3.1 Valores que pueden tomar las Variables contextuales. | 38 |
| 3.2 Tipos de Actividades | 39 |
| 3.3 Vector de Contexto | 40 |
| 4.1 Métodos de Hibridación | 56 |
| 4.2 Vector de perfil de contexto | 58 |
| 4.3 Ejemplo de posibles escenarios para un Turista | 59 |
| 4.4 Definición de Perfiles Contextuales | 60 |
| 4.5 Vector de Perfil contextual | 60 |
| 4.6 Dimensiones de Usuario tomadas de la ontología [199] | 70 |
| 4.7 Características de las dimensiones del modelo de Usuario tomadas de la ontología [199] | 71 |
| 4.8 Tipos de Perfil de Usuario tomados de la ontología la ontología de Usuario del proyecto Travel Guide Ontology [171] | 72 |
| 4.9 Tipos de actividades turísticas adicionadas al diseño de la ontología del proyecto | 72 |
| 6.1 Cálculo de Utilidad R_{a_i} | 114 |
| 6.2 Cálculo de Utilidad $R_{a_i}^{max}$ | 114 |
| 6.3 Cálculo de Utilidad para todos los usuarios. | 115 |
| 6.4 Lista 1 (Hora: 9:43:14) de Recomendaciones para el usuario con ID 323 | 116 |

| | | |
|------|--|-----|
| 6.5 | Lista 2 (Hora: 9:48:42) de Recomendaciones para el usuario con ID 323 | 116 |
| 6.6 | Cálculo de Precision y Recall Promedio para todos los usuarios. | 118 |
| 6.7 | Escala de Calificación transformada. | 120 |
| 6.8 | Tabla de Frecuencia Pregunta 1. | 120 |
| 6.9 | Tabla de Frecuencia Pregunta 2. | 120 |
| 6.10 | Tabla de Frecuencia Pregunta 3. | 121 |
| 6.11 | Tabla de Frecuencia Pregunta 4. | 121 |
| 6.12 | Tabla de Frecuencia Pregunta 5. | 121 |
| 6.13 | Tabla de Frecuencia Pregunta 6. | 121 |
| 6.14 | Tabla de Frecuencia Pregunta 7. | 122 |
| 6.15 | Tabla de Frecuencia Pregunta 8. | 122 |
| 6.16 | Tabla de Frecuencia Pregunta 9. | 122 |
| 6.17 | Tabla de Frecuencia Pregunta 10. | 122 |
| 6.18 | Tabla de Frecuencia Pregunta 11. | 123 |
| 6.19 | Tabla de Frecuencia Pregunta 12. | 123 |
| 6.20 | Tabla de Frecuencia Pregunta 13. | 123 |
| 6.21 | Tabla de Frecuencia Pregunta 14. | 123 |
| 6.22 | Tabla de Frecuencia Pregunta 15. | 124 |
| 6.23 | Tabla de Frecuencias. | 124 |

LISTA DE FIGURAS

| | Pág. |
|--|-------------|
| 2.1 Modos de operación del NFC [65] | 16 |
| 2.2 Cuadro comparativo. Tipos de etiquetas NFC [66] | 16 |
| 2.3 Estructura de un método de aprendizaje y su notación. Adaptado de [71] | 18 |
| 3.1 Partición del día en zonas horarias. | 36 |
| 3.2 Partición de la semana. | 36 |
| 3.3 Etiqueta NFC representativa de la ubicación de un Lugar Turístico. | 37 |
| 3.4 Modelo de variables o dimensiones contextuales. | 37 |
| 3.5 Modelo de Usuario propuesto. | 44 |
| 4.1 Enfoque del sistema propuesto. | 48 |
| 4.2 Paradigmas para incluir contexto en Sistemas de Recomendaciones. | 54 |
| 4.3 Identificación de Perfil Contextual. | 59 |
| 4.4 Evaluación de Precision para Bayes Classifiers. | 63 |
| 4.5 Evaluación de Precision para Functions Classifiers. | 63 |
| 4.6 Evaluación de Precision para Lazy Classifiers. | 63 |
| 4.7 Evaluación de Precision para Rule Classifiers. | 64 |
| 4.8 Evaluación de Precision para Tree Classifiers. | 64 |
| 4.9 Evaluación de Recall para Bayes Classifiers. | 65 |
| 4.10 Evaluación de Recall para Functions Classifiers. | 65 |

| | |
|---|----|
| 4.11 Evaluación de Recall para Lazy Classifiers. | 65 |
| 4.12 Evaluación de Recall para Rule Classifiers. | 66 |
| 4.13 Evaluación de Recall para Tree Classifiers. | 66 |
| 4.14 Estructura de la Ontología empleada en el sistema. | 74 |
| 4.15 Coeficiente de Tanimoto. | 79 |
| 4.16 Evaluación del número de usuarios para los que el recomendador es capaz de recomendar - Similitud Log Likelihood basada en Nearest Neighborhood. | 81 |
| 4.17 Evaluación del número máximo de recomendaciones por usuario y número de usuarios para cada número máximo de recomendaciones - Similitud Log Likelihood basada en Nearest Neighborhood. | 81 |
| 4.18 Evaluación de Precision - Similitud Log Likelihood basada en Nearest Neighborhood. | 82 |
| 4.19 Evaluación de Recall - Similitud Log Likelihood basada en Nearest Neighborhood. | 82 |
| 4.20 Evaluación del número de usuarios para los que el recomendador es capaz de recomendar - Similitud Log Likelihood basada en Neighborhood Threshold. | 83 |
| 4.21 Evaluación del número máximo de recomendaciones por usuario y número de usuarios para cada número máximo de recomendaciones - Similitud Log Likelihood basada en Neighborhood Threshold. | 83 |
| 4.22 Evaluación de Precision - Similitud Log Likelihood basada en Neighborhood Threshold. | 84 |
| 4.23 Evaluación de Recall - Similitud Log Likelihood basada en Neighborhood Threshold. | 84 |
| 4.24 Evaluación del número de usuarios para los que el recomendador es capaz de recomendar - Similitud Log Likelihood basada en Neighborhood Threshold. | 85 |
| 4.25 Evaluación del número máximo de recomendaciones por usuario y número de usuarios para cada número máximo de recomendaciones - Coeficiente Tanimoto basado en Nearest Neighborhood. | 85 |
| 4.26 Evaluación de Precision - Coeficiente Tanimoto basado en Nearest Neighborhood. | 86 |
| 4.27 Evaluación de Recall - Coeficiente Tanimoto basado en Nearest Neighborhood. | 86 |
| 4.28 Evaluación del número de usuarios para los que el recomendador es capaz de recomendar - Coeficiente Tanimoto basado en Neighborhood Threshold. | 87 |

LISTA DE FIGURAS

| | | |
|------|--|-----|
| 4.29 | Evaluación del número máximo de recomendaciones por usuario y número de usuarios para cada número máximo de recomendaciones - Coeficiente Tanimoto basado en Neighborhood Threshold. | 87 |
| 4.30 | Evaluación de Precision - Coeficiente Tanimoto basado en Neighborhood Threshold. | 88 |
| 4.31 | Evaluación de Recall - Coeficiente Tanimoto basado en Neighborhood Threshold. . . | 88 |
| 4.32 | Grafo representativo del Problema del Agente Viajero. | 92 |
| 5.1 | Arquitectura física del sistema. | 96 |
| 5.2 | Arquitectura lógica del sistema. | 97 |
| 5.3 | Usos de los dispositivos NFC. Adaptado de [214]. | 99 |
| 5.4 | NDEF Message. Adaptado de [214]. | 100 |
| 6.1 | Ejemplo de pregunta de la Encuesta. | 112 |
| 6.2 | Ejemplo de pregunta de la Encuesta. | 119 |
| 6.3 | Prueba T - Evaluación del sistema. | 125 |
| 6.4 | Distribución de las calificaciones. | 126 |
| 6.5 | Porcentaje de frecuencia. | 127 |

Capítulo 1

Introducción

El turismo es una de las industrias de más rápido desarrollo en el mundo [1]. Las llegadas de turistas internacionales crecieron un 5 % en 2013, alcanzando un récord de 1.087 millones de llegadas, según el último Barómetro OMT del Turismo Mundial. A pesar de los desafíos económicos globales, los resultados del turismo internacional estuvieron muy por encima de las expectativas, con un adicional de 52 millones de turistas internacionales viajando por el mundo en 2013. Para 2014, la OMT previó un 4 % a un 4,5 % de crecimiento - una vez más, por encima de las proyecciones a largo plazo [2]. El turismo es un dominio desafiante para el desarrollo de servicios basados en la localización que proporcionan servicios de asistencia e información personalizados y móviles a los turistas. Las guías de turismo móviles suelen incluir una amplia gama de servicios de navegación, planificación de rutas y re-planificación, entrega de contenido basado en la localización, alertas sobre eventos relevantes, servicios de pago y reserva de instalaciones (por ejemplo, aparcamiento y reserva de asiento).

Por otro lado, dentro de los entornos turísticos se hace necesario personalizar la interacción del usuario con su ambiente por medio de la tecnología, más específicamente con las (TICs), rescatando dentro de este campo, el rápido desarrollo de las tecnologías de computación móvil [3]. Para el campo del turismo, el uso efectivo de dichas tecnologías permite personalizar la interacción del turista con su ambiente, y para proporcionar servicios personalizados y relevantes a los usuarios son empleados los sistemas de recomendación, y en este contexto, los dispositivos móviles son particularmente útiles debido a su naturaleza ubicua que los convierte en una plataforma atractiva para asistir a los turistas a elegir los puntos de interés para visitar en función de su ubicación física (Noguera et al. 2012), y una tecnología que está emergiendo y es incipiente, es NFC (Near Field Communications) [4] [5] la cual proporciona una forma natural de interacción entre el usuario y su entorno. La tecnología NFC ofrece una solución simple basada en el “paradigma de tocar” que hace posible el intercambio de información y el acceso a contenidos y servicios de forma intuitiva.

Uno de los desafíos de los sistemas de recomendaciones para turismo es el uso de información contextual, y debido a las ventajas que ofrece la computación móvil [6], permite tener conocimiento contextual (ubicación física, la velocidad de movimiento, tiempo, entre otros) de los usuarios en consideración [7] [8], específicamente usando la tecnología NFC para dicho propósito en el presente trabajo.

1.1. Contexto de Investigación

En los últimos tiempos el turismo ha tomado gran importancia en la economía mundial, no solo por ser una actividad generadora de riqueza y crecimiento económico, sino también por el efecto positivo generado en otros sectores involucrados en su operación. En la actualidad, existe un gran número de destinos alrededor del mundo con diferentes atractivos, lo cual ha generado un aumento en la competitividad entre los destinos, y por ende todo un juego en precios, calidad, oferta, servicio, entre otros, con el fin de atraer más turistas y de brindarles una experiencia atractiva en el lugar. El turismo es una actividad que consiste en una gran variedad de comercios al por menor y de servicios, y los sectores más afectados son los hoteles y otros alojamientos, establecimientos de comida y bebida, y las instalaciones recreativas y de entretenimiento, como parques temáticos y centros comerciales [9]. Algunas personas están dispuestas a gastar más dinero y tiempo de ocio en el turismo que otras personas, y para ellos, éste se ha convertido en parte de su estilo de vida, así mismo los hábitos de consumo que representan estilos de vida y las variables de antecedentes demográficos influyen en el consumo y la disposición a consumir más [10].

En consecuencia, un destino turístico debe contar con herramientas tecnológicas y metodológicas que permitan motivar o incentivar el consumo turístico, priorizando posibles patrones de consumo o visita, que permitan influir en el tiempo de permanencia y su movimiento dentro de un destino; razón por la cual se hace necesario personalizar la interacción del usuario con su ambiente por medio de la tecnología, más específicamente con las Tecnologías de la Información y las Comunicaciones (TICs), rescatando dentro de este campo, el rápido desarrollo de las tecnologías de computación móvil que permite el acceso a la información en cualquier momento y en cualquier lugar, incluso en entornos con escasas conexiones de red físicas.

En esta línea, los Sistemas de Recomendaciones (S.R) representa un hilo de investigación con numerosos campos de aplicación posibles (por ejemplo, las compras móviles, publicidad/marketing y suministro de contenido), los cuales son capaces de obtener información relacionada con los gustos, preferencias y demás características del turista, para estimular el deseo por visitar y conocer la oferta propia del destino [11]. Definiendo S.R. como cualquier sistema que produce recomendaciones individualizadas o que tiene el efecto de guiar al usuario en un camino personalizado de acuerdo a sus intereses para tomar una decisión en un gran espacio de posibles opciones [12]. Para el caso del turismo, las recomendaciones son un medio común de la planificación en viajes y hospedaje, siendo una extensión intuitiva y valiosa a los sistemas de información turística [13].

Luego, el apoyo de tecnologías ubicuas se hace indispensable para mejorar la captura de información relacionada con el contexto del usuario, las cuales permiten la interacción con el ambiente que lo rodea (Inteligencia Ambiental AmI) [14] [15]. En este sentido, dentro de los entornos en donde se pueden ofrecer información y servicios personalizados de acuerdo al contexto del usuario es en una ciudad, y es aquí donde se introduce el concepto de SmartCity (en castellano Ciudad Inteligente) la cual es una ciudad que usa las tecnologías de la información y las comunicaciones para hacer que tanto su infraestructura crítica, como sus componentes y servicios públicos ofrecidos sean más interactivos, eficientes y los ciudadanos puedan ser más conscientes de ellos [16].

En estos entornos, el usuario está rodeado de recursos ubicuos incrustados en objetos aumentados (ambientes etiquetados), en los cuales, la interacción con el usuario, la visualización de información, acceso a la personalización y la sensibilidad del contexto, juegan un papel muy importante [17], [18] y la tecnología NFC [4], [5] proporciona una forma natural de interacción entre el usuario y su entorno. La tecnología NFC ofrece una solución simple basada en el “paradigma de tocar” que hace posible el intercambio de información y el acceso a contenidos y servicios de forma intuitiva. Además, se

1.2. Planteamiento del Problema

simplifica la interacción de las personas con el medio ambiente, lo que resulta en el paradigma “Touch Computing”, donde los usuarios obtienen en su dispositivo móvil objetos cercanos de la vida cotidiana, dotados con marcas visuales y etiquetas RFID u otros dispositivos NFC, con el fin de desencadenar los servicios inteligentes ofrecidos por esos objetos. Esta característica hace que sea el candidato preferido para el desarrollo de ambientes inteligentes y permite la interacción del usuario con los sitios de la oferta turística, así como también ofrece una alternativa de provisión de información contextual basada en localización con NFC.

No obstante, en entornos turísticos la motivación del consumo turístico por medio de la tecnología, específicamente a través de Sistemas de Recomendaciones, enfrenta nuevos desafíos, si se tiene en cuenta que no basta con recomendar al usuario basado sólo en sus gustos y preferencias sino que es posible priorizar o influir en posibles patrones de consumo turístico en cuanto a su permanencia y su movimiento dentro de un destino. En este escenario, los algoritmos de recomendación deberán operar para ofrecer recomendaciones que habiliten movimientos dentro del destino de acuerdo a varios patrones o características de consumo en los que se esté interesado en incentivar.

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

1.2. Planteamiento del Problema

El turismo es una de las industrias de más rápido desarrollo en el mundo [1], y por esto es un desafío el desarrollo de servicios basados en localización que proporcionan servicios de asistencia e información personalizadas y móviles a los turistas. Por lo general, el turismo no es una industria en sí, sino un conjunto de industrias interrelacionadas, que venden productos a los turistas, así como a una serie de otros clientes: hoteles, operadores turísticos, y agencias de viajes, compañías aéreas, etc [9]. Entender el consumo turístico se ha convertido en una característica importante de la geografía del turismo en los últimos años, como el reconocimiento de que ha crecido la incorporación de los lugares y la gente en el turismo como una formación socio-económica es en gran parte estructurada por los deseos y las decisiones de los turistas ordinarios [19]. Hay dos dimensiones que deben tenerse en cuenta cuando se refiere al consumo del turismo. Por un lado, el consumo requiere capacidades financieras, por otra parte, debe haber un deseo de consumir. De acuerdo con este punto de vista los factores estructurales tales como las restricciones monetarias y la demografía son todavía importantes predictores sobre el consumo en el turismo (consumo turístico) y la disposición a consumir más en el turismo (el deseo de viajar más) [10]. Para el campo del turismo, el uso efectivo de las (TICs) permite personalizar la interacción del turista con su ambiente, generando ambientes etiquetados en los que se encuentre rodeado de objetos con los cuales interactuar de una manera natural y directa. En este sentido, para proporcionar servicios personalizados y relevantes a los usuarios son empleados los sistemas de recomendación, y uno de los desafíos de los sistemas de recomendaciones para turismo es el uso de información contextual, y debido a las ventajas que ofrece la computación móvil [6], permite tener conocimiento contextual (ubicación física, la velocidad de movimiento, tiempo, entre otros) de los usuarios en consideración [20] [8].

En consecuencia, la mayoría de investigaciones que hacen uso de información contextual en sistemas turísticos no emplean Sistemas de Recomendaciones como enfoque de personalización, y las que emplean algoritmos de recomendación, no los han utilizado para priorizar las recomendaciones de acuerdo a patrones de consumo turístico y optimizar el tiempo de permanencia del turista en un

destino. Por otro lado, en cuanto a trabajos relacionados con la obtención de información contextual en ambientes turísticos, la mayoría de prototipos ofrecen servicios de enrutamiento basado en la ubicación geográfica de los usuarios móviles, obtenida típicamente de receptores GPS, o técnicas de seguimiento de ubicación alternativas (Wi-Fi, celular-id, RFID, etc.) [18] [21] [22].

Para terminar, es claro que se puede influenciar sobre el comportamiento del turista mediante herramientas tecnológicas y metodológicas que permitan obtener una motivación o incentivo al consumo turístico empleando Sistemas de Recomendaciones para priorizarlas de acuerdo a posibles patrones de consumo que permitan incrementar el deseo de consumir más en el mismo. En este contexto, la mayoría de aproximaciones en el área turística se ha enfocado hacia la prestación de servicios de viaje y actividades de asesoramiento, listas de puntos de interés (POI) que responden a las preferencias del usuario y recomendaciones de paquetes turísticos, entre otros.

De acuerdo a lo expuesto anteriormente, se plantea la siguiente pregunta para orientar el proceso de investigación del presente proyecto:

Cómo motivar o incentivar el posible consumo de oferta turística a través de Sistemas de Recomendaciones en un entorno de computación móvil?

1.3. Objetivos

Una vez expuesto el enfoque de este proyecto, se presenta los objetivos de investigación que materializaron la exploración de las hipótesis previamente presentadas.

1.3.1. Objetivo General

Proponer un Sistema de Recomendaciones turístico móvil basado en contexto para motivar o incentivar el posible consumo de oferta turística.

1.3.2. Objetivos Específicos

- Definir los parámetros contextuales que alimentarán el sistema de recomendaciones turístico móvil.
- Diseñar un filtro de recomendación a partir de la combinación de los algoritmos de filtrado de acuerdo a los información contextual previamente caracterizada.
- Evaluar el sistema de recomendaciones propuesto por medio de un conjunto de escenarios simulados con patrones de consumo a motivar o incentivar.

1.4. Alcance

El presente trabajo de investigación está enmarcado en el área de Sistemas de Recomendación en un entorno Turístico Móvil y para limitar su alcance se tiene en cuenta las siguientes consideraciones:

- El presente trabajo es evaluado de manera online en un entorno cerrado controlado por medio de escenarios turísticos simulados que intentan ubicar al usuario en una determinada situación para hacer uso del sistema. Sin embargo, no es evaluado en un entorno real con turistas y sitios turísticos reales, debido a la complejidad que conlleva su despliegue en la vida real.

1.5. Contribuciones

- El conjunto de datos empleado para las recomendaciones fue obtenido en la experiencia del caso de estudio realizada bajo el Proyecto de Trazabilidad Turística de la Universidad del Cauca y adicionalmente se agregaron nuevos items a recomendar de los sitios web Foursquare y TripAdvisor.
- El sistema implementado es un prototipo experimental, enfocado en la utilidad que genera la utilización del sistema propuesto ya que incentiva el consumo de oferta turística por medio de la habilitación y maximización de posibles movimientos a realizar por parte de un turista optimizando su tiempo en un destino.

1.5. Contribuciones

El aporte investigativo del presente proyecto se centra en:

- Definición de un modelo de información contextual y comportamiento contextual (Capítulo 3). 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) un análisis de datos de contexto para realizar el modelado de variables o dimensiones que comparten el mismo tipo de información iii) el diseño del modelo de datos y de usuario en términos de características de datos en que se basan las recomendaciones y las propiedades del lado del usuario que tienen impacto en el diseño del sistema de recomendaciones.
- Un enfoque (Capítulo 4) o planteamiento conceptual para motivar o incentivar el posible consumo de oferta turística por medio de la habilitación y maximización de posibles movimientos a realizar por parte de un turista optimizando su tiempo en un destino por medio de un sistema de recomendaciones turístico móvil.
- Un prototipo experimental que implementa la arquitectura propuesta (Capítulo 5) del Sistemas de Recomendaciones en un entorno turístico móvil soportado en interacción contextual con NFC.
- Un reporte de resultados de pruebas y experimentación del sistema propuesto a través de la evaluación de la calidad y utilidad de las recomendaciones y el grado de aceptación del sistema por parte del turista (Capítulo 6).

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.

1.6.1. Parte I. Introducción

Esta primera parte está dedicada a la descripción del entorno de investigación del presente proyecto de maestría, los principales conceptos relacionados y los trabajos más significativos del estado actual del conocimiento relacionados con el trabajo. Esta parte está conformada por los siguientes capítulos.

Capítulo 1. Introducción Este capítulo está dedicado a presentar el entorno de investigación considerado en el presente proyecto de grado.

Capítulo 2. Estado Actual de Conocimiento Presenta una visión general sobre los trabajos relacionados y los conceptos que giran en torno a las áreas de sistemas de recomendaciones sensibles al contexto en entornos turísticos móviles, alternativas de interacción contextual basada en ubicación de los usuarios móviles y aproximaciones de recomendación basadas en conocimiento.

1.6.2. Parte II. Propuesta

La segunda parte de esta monografía expone los avances propios realizados al interior del presente proyecto de investigación.

Capítulo 3. Diseño del Modelo Contextual En este capítulo se presenta un estudio sobre las características de la información contextual de Usuario y de Datos. En este sentido, se define y presenta un modelo contextual del sistema comprendido por un Modelo de los Datos en los que se basan las recomendaciones y un Modelo de Usuario; los cuales son insumos de referencia para el planteamiento del enfoque de recomendaciones propuesto al interior de este proyecto de investigación, dando cumplimiento al objetivo específico 1.

Capítulo 4. Enfoque del Sistema de Recomendaciones Propuesto Este capítulo propone un enfoque de recomendación diseñado para motivar o incentivar el posible consumo de oferta turística por medio de la habilitación y maximización de posibles movimientos a realizar por parte de un turista optimizando su tiempo en un destino, y generando el modelo conceptual del sistema de recomendaciones a construir para dar planteado en el objetivo específico 2.

Capítulo 5. Implementación del Sistema Propuesto En este capítulo se enfatizará en la arquitectura física y lógica del enfoque propuesto para el sistema de recomendaciones junto con los detalles de implementación de cada uno de los módulos que la integran, con lo cual se termina de dar cumplimiento al objetivo 2.

1.6.3. Parte III. Evaluación y Resultados

Corresponde a la última parte de la monografía, en la cual se presentan los elementos utilizados para la evaluación del sistema propuesto en la parte II.

Capítulo 6. Experimentación y Evaluación El capítulo de experimentación y evaluación, contiene los fundamentos teóricos de la evaluación de técnicas de predicción en sistemas de recomendación, así como el plan de pruebas y resultados, fruto de este proceso, dando cumplimiento al objetivo específico 3.

Capítulo 7. Conclusiones y trabajos futuros Presenta las principales contribuciones del trabajo, conclusiones, restricciones y perspectivas de la investigación, analizando los resultados del trabajo realizado y generando un conjunto de recomendaciones importantes para el desarrollo de trabajos futuros.

Capítulo 2

Estado del Actual de Conocimiento

Este capítulo presenta un análisis detallado de los referentes teóricos y trabajos de investigación, que justifican la realización del presente proyecto y guían su ejecución.

2.1. Contexto general

En esta sección se describen los conceptos más relevantes en el núcleo temático de la presente propuesta.

2.1.1. Contexto del usuario

La importancia de la información contextual ha sido reconocida por los investigadores y profesionales en muchas disciplinas, incluyendo la personalización de comercio electrónico, búsqueda de información, la computación ubicua y móvil, la minería de datos, marketing y gestión. Mientras que una cantidad sustancial de investigaciones se han realizado en el área de sistemas de recomendación, la mayoría de los enfoques existentes se centran en la recomendación de los ítems más relevantes para los usuarios sin tener en cuenta cualquier información contextual adicional, como el tiempo, la ubicación, o la compañía de otras personas (por ejemplo, para ver películas o salir a cenar) [20].

Dado que el ambiente turístico es un ambiente muy diverso, la incorporación de la información contextual sobre el usuario en el proceso de recomendación ha atraído gran interés. Esta contextualización es investigada como un paradigma para la construcción de sistemas inteligentes que pueden predecir y anticipar mejor las necesidades de los usuarios, y actuar de manera más eficiente en respuesta a su comportamiento.

Una de las definiciones más citadas de contexto es la definición de Dey et al. [23] que define el contexto como “cualquier información que se puede utilizar 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, incluyendo el usuario y las propias aplicaciones”. Para operacionalizar contexto, los investigadores han tratado de definir el contexto enumerando categorías. Los primeros trabajos de Schilit et al. [24], [23] dividieron el contexto en tres categorías.

- Informática contexto, como la conectividad de red, los costos de comunicación, y la comunicación ancho de banda y recursos cercanos, como impresoras, pantallas, y estaciones de trabajo.

- Contexto del usuario, tales como el perfil del usuario, la ubicación, las personas cercanas, y la situación social.
- Contexto físico, tales como la iluminación, los niveles de ruido, las condiciones del tráfico, y la temperatura.

Chen y Kotz [25] añaden tiempo como una cuarta categoría de contexto. Schmidt et al. [26] añadió una categoría de tareas y define las siguientes dimensiones: el usuario, el entorno social del usuario, tareas, ubicación, infraestructura, condiciones físicas, y el tiempo. Beale y Lonsdale [27] sostienen que es esencial para capturar las interacciones entre el medio ambiente, el usuario, sus tareas, y otros usuarios. El Medio Ambiente lo constituyen la computación, el tiempo y las características del contexto físico. Li et al. [28] definen cinco dimensiones de contexto: quién (usuario), qué (objeto), cómo (actividades), donde (ubicación), y cuándo (el tiempo). Otros ejemplos incluyen las definiciones de Schmidt [28], Tankelevic IENE y Damas EVIC ius [29], Kurti [30], y Hu y Moore [31].

La mayoría de estas definiciones se refieren a categorías de contexto similar como se define en las definiciones genéricas: la ubicación, el tiempo, la informática, el usuario, la actividad y las relaciones sociales. Aunque el alcance de las categorías de contexto difiere (es decir, algunas definiciones especifican sólo unas pocas categorías que cubren varios elementos mientras que otras definiciones intentan definir las categorías de contexto más precisamente), hay muchas similitudes entre las definiciones de contexto.

La necesidad de definir y modelar el contexto más precisamente y de una manera consistente ha sido identificado por varios investigadores [27], [32]. Una definición y el modelo de contexto preciso puede facilitar la identificación de lo que constituye y no constituye contexto y pueden permitir la reutilización y el intercambio de datos contextuales a través de las aplicaciones. En este sentido, en [7] se introduce una simple clasificación de información de contexto que es relevante para aplicaciones sensibles al contexto. Esta clasificación fue construida por el análisis y la integración de las definiciones de contexto existentes.

Computación

La Computación del Contexto ha sido investigado ampliamente por la comunidad de investigadores [33], incluso por los investigadores en el área de aprendizaje móvil [34], [35]. Las características de computación se pueden clasificar en tres áreas.

- Red: incluye propiedades estáticas y dinámicas de la red, tales como ancho de banda máximo y disponible.
- Hardware: comprende capacidades de entrada y salida de los dispositivos, de almacenamiento o capacidades de CPU, etc.
- Software: describe si el contexto de entrega es compatible con ciertas APIs, formatos de documentos, sistemas operativos, protocolos de la capa de aplicaciones, etc.

Ubicación

Además de la computación de contexto, el contexto de ubicación ha dominado la investigación sobre la informática móvil sensible al contexto en gran medida [26]. Modelos de Ubicación han sido propuestos para la captura de información geométrica de los objetos legible por humanos, incluidas

2.1. Contexto general

las personas y los dispositivos, y las relaciones entre los objetos. Estos incluyen: 1) la proximidad de los objetos dentro de un espacio, 2) la capacidad comunicativa, y 3) la orientación.

Tiempo

El Contexto de Tiempo incluye la fecha y hora, o menos específico, información acerca de la semana, mes o semestre del año. El tiempo se utiliza a menudo en conjunción con otras piezas de contexto, ya sea como una marca de tiempo o como lapso de tiempo, lo que indica un instante o período durante el cual se conoce otra información contextual pertinente [23].

Condiciones Físicas

El Contexto físico describe las condiciones ambientales donde el sistema o el usuario están situados, y por lo general incluye medidas para el calor, la luz y el sonido. El contexto físico ha sido investigado ampliamente en la investigación de automatización del hogar.

Actividad

El Contexto de Actividad refleja las tareas, objetivos o acciones del usuario. Los modelos permiten la captura de las acciones del usuario, que se componen de eventos dentro de una aplicación, la sesión y los datos relacionados con el tiempo. Estos datos se analizan a continuación, para inferir información sobre la actual tarea, objetivo o tema de interés del usuario, por ejemplo mediante el uso de ontologías de dominio que describen el dominio [36]. Los modelos proporcionan elementos para capturar datos contextuales adicionales, pero no definen tipos específicos de dichos datos.

Usuario

Los sistemas sensibles al contexto deben ser capaces de observar el comportamiento del usuario y los cambios en el contexto y hacer generalizaciones y predicciones sobre el usuario y el contexto en base a sus observaciones. La información sobre el usuario normalmente es recolectada en un modelo denominado de usuario y administrado por un sistema de modelado de usuario [37], en el que se definen (en el contexto de un sistema de diálogo) los dos siguientes conceptos:

- **Modelo de Usuario:** La determinación del modelo de usuario es una fuente de conocimiento en un sistema que contiene supuestos explícitos sobre todos los aspectos del usuario que pueden ser relevantes para el comportamiento del sistema. Estos supuestos deben ser separables por el sistema del resto de los conocimientos del sistema.
- **Componente de modelado de usuario:** un componente de modelado de usuario es la parte de un sistema cuya función es construir incrementalmente un modelo de usuario; almacenar, actualizar y borrar entradas; para mantener la consistencia del modelo; y suministrar otros componentes del sistema con suposiciones sobre el usuario.

A continuación se detallan los elementos de datos disponibles en el modelo de usuario:

- **Información Básica Personal:** suele incluir información de identificación, nombre, información de contacto, la información de autenticación, información sobre la accesibilidad, incluyendo capacidades de lenguaje y discapacidades, y otras características personales como el género, la edad, profesión y nivel educativo.

- Intereses: Intereses o preferencias de los usuarios las cuales son características clave para apoyar la personalización.

En la presente propuesta, basado en la definición de las dimensiones de Li et al. [28], se analizan y agrupan los datos de contexto que tienen el mismo tipo de información contextual, combinando las variables contextuales en tres grupos que corresponden con los que se producen los eventos relacionados con el dispositivo durante la recomendación de las actividades turísticas, la ubicación física durante y el tiempo en que se produce este evento.

- Cuándo: El día (o semana o fin de semana) y el marco de tiempo (mañana, tarde, atardecer o noche) se etiquetan como la fecha y tiempo [38].
- Dónde: ubicación física del turista obtenido a partir de la interacción con la etiqueta NFC del sitio que se encuentra visitando.
- Qué: Todas las demás variables o dimensiones que tienen que ver con las tareas que realiza el sistema de recomendaciones, específicamente lo que debe tener en cuenta para proporcionar las actividades turísticas a realizar en dicho momento.

El modelo de contexto-agrupación que se presenta aquí proporciona una base para la representación de las variables de contexto para identificar posteriormente perfiles turísticos contextuales.

2.1.2. Sistemas de Recomendación

Un Recomendador o Sistema de Recomendaciones (en inglés Recommender Systems - RS) tiene como objetivo ayudar a un usuario o un grupo de usuarios en un sistema para seleccionar ítems de un espacio apiñado de información [39]. Con el fin de generar recomendaciones, un RS intenta identificar las coincidencias de las características de los usuarios con las características de los ítems mediante la realización de filtrado de contenido, o podría tomar previas interacciones del usuario en el sistema para que coincida con patrones de usuarios similares [40]. Un dominio típico en de los Sistemas de Recomendaciones es la World Wide Web (WWW), ya que con el abrumador crecimiento de la información disponible y el continuo aumento de dispositivos diferentes que se pueden utilizar para acceder a ella, éstos sistemas han asumido un papel importante en la vida diaria de las personas para ayudarles a encontrar los recursos pertinentes, como películas [41], libros [42], música [43] o destinos turísticos [44].

Clasificación de los Sistemas de Recomendaciones

Filtrado basado en Demografía En [45] Pazzani investigó el enfoque basado en recomendación demográfica. Los datos demográficos se pueden utilizar para identificar el tipo de usuarios a los que les gusta un determinado objeto y poder crear “estereotipos”. Los sistemas de recomendación demográficos apuntan a categorizar el usuario en función de los atributos personales y hacer recomendaciones basadas en las clases demográficas.

- Ventajas: La ventaja de un enfoque demográfico es que las calificaciones del usuario para los ítems no se utilizan, por lo que los nuevos usuarios pueden obtener recomendaciones antes de que hayan clasificado algún ítem. Y no es necesario el conocimiento de los ítems y sus características; Por lo tanto, la técnica es independiente del dominio.

2.1. Contexto general

- Desventajas: La desventaja del enfoque demográfico es que la recopilación de los datos demográficos requeridos conduce a problemas de privacidad [46]. La clasificación demográfica también es demasiado básica para recomendaciones altamente personalizadas. Por ejemplo, no todas las mujeres de 20 años de edad que trabajan disfrutan de las mismas películas. Y los clientes con diferentes opiniones o un inusual gusto resultan en un bajo coeficiente de correlación con otros clientes, por lo tanto las recomendaciones para ellos son muy difíciles de encontrar y también causan recomendaciones impares para sus usuarios correlacionados, este problema se denomina “grey sheep” se discute en [47]. Otro desafío es la dificultad para cambiar un perfil creado de un cliente una vez que el gusto del cliente cambia, lo que se conoce como el problema de estabilidad vs plasticidad [48].

Basado en Contenido El filtrado colaborativo utiliza el supuesto de que las personas con gustos similares calificarían cosas de manera similar, mientras que el filtrado basado en contenido utiliza el supuesto de que los ítems con características similares serán valorados de manera similar. Si bien los métodos de filtrado de información de colaboración y demográficos no requieren ninguna información adicional sobre el ítem, el enfoque de filtrado basado en contenido depende de la disponibilidad de (creado manualmente o extraídos automáticamente) descripciones de los mismos y de un perfil de usuario que asigna relevancia a estas características. Esos ítems que son más similares a una consulta o para el perfil del usuario son recomendados [49].

Por ejemplo, en una aplicación de recomendación de películas, con el fin de recomendar películas a usuarios u , el sistema de recomendación basado en contenido busca las similitudes entre los ítems que el usuario u ha calificado altamente (clasificación de forma explícita, comprado o visto) en el pasado (actores específicos, géneros). Sólo las películas que tienen un alto grado de similitud con las preferencias del usuario pueden ser recomendados.

El perfil de usuario de las preferencias se almacena como un vector de palabras clave. Estos perfiles se obtienen mediante el análisis del contenido de los ítems anteriormente vistos y valorados por el usuario y se construyen generalmente mediante técnicas de análisis de palabras clave del área de la recuperación de información. La recuperación de la información implica la asignación de varios pesos de palabras clave mediante el uso de algoritmos como los algoritmos Winnow [50] y Rocchio.

- Ventajas: Debido a que el enfoque basado en el contenido utiliza características de ítems y lo compara con otros ítems sin importar cual es el ítem, lo importante es recomendar, el enfoque no tiene conocimiento sobre el dominio, y funciona bien si los ítems se pueden representar adecuadamente como un conjunto de características.
- Desventajas: Hay varias desventajas. Cuando no hay características disponibles de los ítems, se debe suministrar manualmente las características de los ítems, que puede ser una tarea desagradable. Cuando se tienen todos los ítems únicos con todas las diferentes características, el enfoque no será capaz de hacer una recomendación adecuada. El cálculo de similitud se limita a las características del ítem, y también adolece del problema de arranque en frío y estabilidad frente a problema plasticidad.

Filtrado Colaborativo Es el enfoque más popular e implementado. En su aplicación más simple y original este enfoque recomienda al usuario activo, los ítems que a otros usuarios con gustos similares les ha gustado en el pasado. La similitud en el gusto de los dos usuarios se calcula en base a la similitud en el historial de calificaciones de los usuarios. De hecho los recomendadores colaborativos

aprovechan los métodos de vecinos más cercanos (método de vecindad basada en el usuario y el método de vecindad basado en ítems). Así, los métodos basados en usuario se basan en la opinión de los usuarios afines para predecir una calificación, y generar recomendaciones, mientras que los enfoques basados en ítems toman las calificaciones otorgadas a los ítems similares y generan recomendaciones.

Los sistemas de recomendación basados en filtrado colaborativo [41] apuntan a resolver algunos de los problemas de los sistemas de recomendación basados en contenido y basado en reglas, por lo que han logrado un éxito aceptable en sitios de comercio electrónico [51]. Estos modelos generalmente incluyen coincidencia de ítems del usuario actual (como calificaciones de libros o películas) con usuarios similares (vecinos cercanos) para recomendar ítems que aún no han sido valorados por este usuario. Los sistemas de filtrado de colaboración tradicionales utilizan un enfoque estándar de clasificación basado en memoria basada en el método k -vecino más cercano (kNN). En este algoritmo, el perfil de usuario de destino se compara con otros perfiles de usuario para identificar los primeros k usuarios que tienen intereses similares a este usuario. En el filtrado tradicional de colaboración, el rating predicho del usuario activo a en cada ítem j se calcula como la suma ponderada de los rankings de los usuarios similares en el mismo ítem como se muestra en la siguiente ecuación, donde n es el número de usuarios similares para tener en cuenta, α es un normalizador, $v_{i,j}$ es el voto del usuario i sobre el ítem j , \bar{v}_i es el rating promedio de los usuarios i y $w(a, i)$ es el peso de estos n usuarios similares.

$$p_{a,j} = \bar{v}_a + \alpha \sum_{i=1}^n w(a, i)(v_{i,j} - \bar{v}_i)(1)$$

El valor de $w(a, i)$ se puede calcular de muchas maneras. Los métodos más comunes son la similitud del coseno, similitud euclidiana o Correlación de Pearson en los perfiles de usuario.

Aunque estos sistemas tienen como objetivo proporcionar una solución a los problemas en los modelos anteriores de los sistemas de recomendación, adolecen de sus propios problemas. El problema más importante de los sistemas de filtrado colaborativo basados en memoria tradicionales es que no son escalables. En el algoritmo kNN, la formación de los vecinos debe hacerse en un método en línea. En otras palabras, contrariamente a los métodos basados en modelos en los que la fase de aprendizaje del modelo se realiza fuera de línea con los datos de entrenamiento, la fase de modelado en estos sistemas se lleva a cabo como una tarea en línea. Con el aumento de los usuarios y los ítems, este método puede ser inaceptablemente lento para producir recomendaciones dinámicas durante la interacción con los usuarios.

Otro problema es debido a la naturaleza dispersa de la mayoría de los conjuntos de datos. Más ítems en el conjunto de datos resultan en una disminución de la densidad del perfil de usuario. Como consecuencia, la probabilidad de similitud de ítems observados entre los usuarios disminuye, lo que resulta en menos confianza en los cálculos de correlación.

Adicionalmente, este método también sufre el problema “nuevo ítem”. Cuando se añade un ítem o producto nuevo al conjunto de ítems, que nunca se ha visto ni calificado por cualquier usuario. Como resultado, no existe en ningún perfil de usuario y el sistema de recomendación no se lo recomienda a cualquier usuario. La falta de capacidad para explicar recomendaciones para los usuarios es otro problema de estos sistemas. Como los recomendadores de filtrado colaborativo no utilizan otros recursos de información, al igual que el contenido o los datos semánticos, no pueden explicar la razón para recomendar un ítem específico al usuario. Para resolver los problemas sparsity y escalabilidad, algunas

2.1. Contexto general

técnicas de utilización de optimización [52] [53] [54]. Estos métodos incluyen técnicas de reducción de dimensionalidad, la indexación de similitud y agrupación offline de perfil de usuario en el pasado para buscar en el grupo combinado, mientras genera las recomendaciones.

Otro método que se basa en filtrado colaborativo es filtrado colaborativo basado en ítem [55]. En este método, una matriz de similitud de ítems se produce sobre la base de los datos de calificación de perfiles de usuario en una manera fuera de línea. Esta matriz se utiliza para generar recomendaciones en la fase en línea. En otras palabras, en lugar de depender de la similitud entre los ítems en sus descripciones de contenido, se calcula sobre la base de valoraciones de los usuarios de los mismos. Cada ítem se muestra como un vector y las similitudes se calculan en base a medidas como la similitud del coseno o similitudes basadas en correlación como Pearson o Spearman. El proceso de la generación de recomendaciones predice la calificación del usuario de destino a un ítem de destino sin ser visto, por una suma ponderada de las calificaciones dadas a ítems similares al ítem de destino. La evaluación de este método demuestra que puede producir recomendaciones con cualidades similares a las recomendaciones de filtrado de colaboración basadas en modelos [56].

- **Ventajas:** La ventaja del enfoque de colaboración es igual que el enfoque demográfico que no se necesita ningún conocimiento acerca de los productos, por lo que también es de dominio independiente. Las técnicas de filtrado de colaboración son capaces de hacer recomendaciones “fuera de la caja” porque se ven fuera de las preferencias del usuario individual [57]. La principal ventaja del enfoque basado en el modelo es poca memoria y requisitos de tiempo de CPU.
- **Desventajas:** Todavía hay varias desventajas, como que el tamaño del conjunto de datos influye en la calidad de las recomendaciones. Y cuando hay nuevos usuarios y nuevos ítems sin calificación, el enfoque no es capaz de dar una recomendación, ya que no es capaz de identificar a los usuarios con la misma preferencia, ya que no hay preferencias del usuario disponibles. Esto se conoce como el problema de arranque en frío [47]. También la oveja gris y la estabilidad frente a la plasticidad son problemas aquí [47] [48]. El enfoque basado en memoria requiere mucha memoria y tiempo de CPU, por lo que sólo es aplicable a un número relativamente pequeño de usuarios e ítems.

Basados en conocimiento La recomendación basada en el conocimiento ha sido discutida en [58]. Utiliza el conocimiento acerca de los usuarios y de los ítems y las razones acerca de qué ítems cumplen con los requisitos del usuario. Los sistemas basados en restricciones y recomendadores basados en casos son ejemplos de tales sistemas. Por ejemplo, en el caso del sistema de recomendaciones de películas, si un cliente ha indicado que él o ella le gustan las películas con el género de la comedia y la música, el sistema buscará películas que responden a estas preferencias.

En este sentido, los sistemas basados en conocimiento recomiendan ítems basados en el conocimiento de dominio específico acerca de cómo ciertas características de ítems responden a las necesidades y preferencias de los usuarios y, en última instancia, la forma en que el ítem es útil para el usuario. En este tipo de sistemas la función de similitud estima cuánto las necesidades de los usuarios (descripción problema) coinciden con las recomendaciones (soluciones del problema).

- **Ventajas:** La ventaja del enfoque basado en el conocimiento es que no se basa en datos históricos. El sistema no tiene que almacenar cualquier información sobre el usuario en un largo plazo. Así que cada usuario que utiliza el sistema proporciona su preferencia y puede tener en sus recomendaciones de inmediato. Si la preferencia de los usuarios cambia, es fácil ajustarlo.

- Desventajas: La desventaja es que el sistema no aprende de sus preferencias, cada vez que va a utilizar el sistema, tendrá que indicar su preferencia. Con el fin de brindar recomendaciones correctas el conocimiento de dominio es necesario. Por ejemplo, si usted ha elegido un conjunto de preferencias para un ítem, el sistema tiene que averiguar qué ítems se adaptan mejor a las preferencias, y esto requiere un poco de ingeniería del conocimiento.

Híbridos Burke hizo una encuesta sobre varios enfoques híbridos [48]. Los sistemas de recomendaciones híbridos son otra categoría de sistemas de recomendación que intenta superar las limitaciones de los otros enfoques. Un sistema de recomendación híbrido combina dos o más técnicas de recomendación para obtener una mejor optimización del sistema y mejorar las debilidades de los recomendadores individuales. Los enfoques híbridos más populares son las de filtrado basado en contenido y colaboración.

Hay diferentes estrategias mediante las cuales la hibridación se puede lograr y se clasifica en siete categorías que se resumen en la siguiente tabla:

| Método de Hibridación | Descripción |
|-----------------------|---|
| Weighted | Las puntuaciones (o votos) de varias técnicas de recomendaciones son combinadas juntas para producir una recomendación simple. |
| Switching | El sistema switchea entre técnicas de recomendaciones dependiendo de la actual situación. |
| Mixed | Recomendaciones de diferentes recomendadores son presentadas al mismo tiempo. |
| Feature combination | Características de diferentes fuentes de datos de recomendación están incluidas dentro de un algoritmo de recomendación simple. |
| Cascade | Un recomendador refina las recomendaciones realizadas por otro. |
| Feature augmentation | La salida de una técnica es usada como la entrada de características de otra. |
| Meta-level | El modelo aprendido por el recomendador es usado como entrada de otro. |

Tabla 2.1: Categorías de métodos híbridos

2.1.3. NFC

La tecnología conocida como Near Field Communication o NFC puede considerarse como un avance del RFID o Radio Frequency Identification [59]. Básicamente, NFC permite realizar una comunicación simple, segura e intuitiva entre dispositivos. NFC aparece como una evolución en el uso de aplicaciones dentro del teléfono móvil, pues se presenta como un sistema de comunicación sencilla, una alternativa para el manejo de pagos y una opción para el almacenamiento de datos de forma más segura para los dispositivos electrónicos móviles [60]. La principal característica que hace que la tecnología NFC sea interesante y atractiva, es que complementa a otras tecnologías inalámbricas como el Bluetooth, Wifi y el mismo RFID. La principal diferencia entre NFC y los otros esquemas sin contacto, es que no está pensada para la transmisión masiva de datos, pero sí para un intercambio casi instantáneo de una poca cantidad de información y no necesita de un emparejamiento previo [60].

2.1. Contexto general

Definición de NFC

Near Field Communication (NFC), es una tecnología de comunicación de corto alcance, que permite el intercambio de datos entre dos dispositivos de manera inalámbrica. Es compatible con infraestructuras RFID, dado a que es un derivado del mismo [61]. El sistema de corto alcance se compone de dos elementos: un iniciador y un objetivo, en donde cualquier dispositivo con NFC puede adoptar las funciones o el comportamiento de una de estas partes.

Modos de operación

Teniendo en cuenta el modo de operación el NFCIP-1 y el NFCIP-2 son los protocolos mas significativos. A continuación se describen brevemente [59]:

- NFCIP-1: Combina dos protocolos de comunicación que pertenecen al RFID, tales como el MIFARE y el FeliCa, e incluye en ellos nuevos protocolos de transporte.
- NFCIP-2: Hace posible la combinación del NFC con lectores RFID logrando así una compatibilidad.

En este sentido, para establecer una comunicación existen dos modos [62]:

- Modo pasivo: Debe existir un dispositivo que reciba y otro que emita, este último dispone de fuente eléctrica propia para funcionar, y debe generar una señal para el intercambio de datos. Por otro lado el dispositivo receptor no posee baterías y debe aprovechar el campo incidente del emisor para el funcionamiento de sus circuitos.
- Modo Activo: Los dispositivos poseen energía propia, por lo que ambos son capaces de generar el campo electromagnético para la transferencia de datos.

Para los dispositivos NFC es posible hacer la comunicación con el otro par, actuando como etiqueta o haciendo de lector/escritor. Referente a esto, el NFC Forum define los siguientes modos de operación [59] (Figura 2.1):

- Peer to peer: Es utilizado cuando surge la necesidad de transmitir una reducida cantidad de datos (unos pocos kilobytes). Si se desea elevar la cantidad de datos en la transmisión, la tecnología NFC es utilizada para concretar una conexión inalámbrica con el soporte necesario para la comunicación, como por ejemplo Bluetooth [63].
- Lectura/escritura: En este modo, se tiene la capacidad de leer y escribir las etiquetas [63]. Una vez establecida la comunicación es posible el intercambio de texto (en pequeñas cantidades), una dirección de internet o un número de teléfono [64].
- Emulación de tarjeta inteligente: Un lector puede identificar a un dispositivo NFC, como si este fuera una etiqueta NFC o una tarjeta inteligente. Este modo puede ser usado para medios de pago [63], transacciones bancarias, pagos rápidos y control de acceso [64].

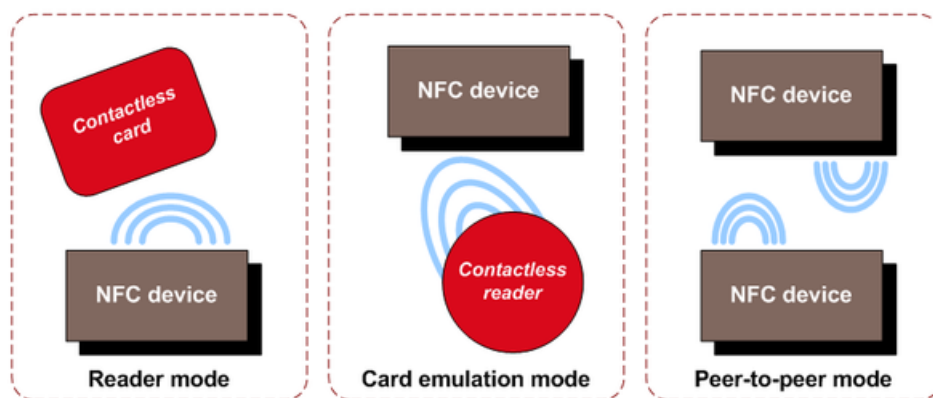


Figura 2.1: Modos de operación del NFC [65]

Etiquetas NFC

Constituyen una parte importante de la tecnología NFC, implementan un almacenamiento pasivo en la espera de que algún lector NFC requiera la información que retienen. El NFC Forum ha definido cuatro tipos de etiquetas [59]. En las especificaciones de las etiquetas se establecen las características de cada uno de los cuatro tipos, de manera tal a lograr la compatibilidad y operabilidad de los dispositivos en sus diferentes modos de lectura o escritura. En estas especificaciones se fijan varios parámetros [64] como los que se muestran a continuación en la Figura 2.2.

| | Memoria | Especificaciones | Velocidad | Lectura/Escritura |
|--------|-------------------------|------------------|--------------------------------|---|
| Tipo 1 | 96 Bytes hasta 2 KBytes | ISO -14443 A | 106 kbits/s | Si |
| Tipo 2 | 48 Bytes hasta 2 KBytes | ISO -14443 A | 106 kbits/s | Si |
| Tipo 3 | Hasta 1MB | FeliCa ISO 18092 | 212 kbits/s y 424 kbits/s [16] | Preconfigurados de fábrica como de lectura y escritura o sólo de lectura. |
| Tipo 4 | 32 KBytes | ISO -14443 A y B | 106 kbits/s y 424 kbits/s | Preconfigurados de fábrica |

Figura 2.2: Cuadro comparativo. Tipos de etiquetas NFC [66]

Formato de datos

Para que las etiquetas y los dispositivos puedan comunicarse entre sí [60], y se pueda conseguir la compatibilidad entre dispositivos NFC y RFID de los diferentes fabricantes [59], el NFC Forum

2.1. Contexto general

definió un formato de datos estandarizado.

NFC Data Exchange Format NDEF [67] Se define un formato de encapsulación de mensaje para intercambiar información entre dispositivos NFC, ya sea de un dispositivo a una etiqueta o entre dos dispositivos NFC activos, también se especifican las reglas para construir un mensaje NDEF correcto, así como una cadena ordenada de registros NDEF. NDEF no hace referencia a ningún circuito, ni arquitectura de conexión, ni se debe pensar que especifique el intercambio de información, es solamente un formato de mensaje. Este formato es el mismo para tarjetas, así como para dispositivos NFC, de esto se concluye que la información de NDEF no guarda relación con el tipo de dispositivo que participa en una comunicación. Con este formato pueden transmitirse varios tipos de información, como:

- Documentos o fragmentos XML, imágenes de diverso formato y datos encriptados.
- Cadenas de información encapsulada.
- Documentos múltiples que guardan alguna relación lógica.

Record Type Definition RTD Proporciona las pautas para la especificación de los tipos de registros, que pueden ser incluidos en mensaje NDEF. Esta especificación soporta aplicaciones específicas NFC [67]. El NFC Forum define dos tipos: NFC Forum External Types y NFC Forum Well-Known Types, siendo el primero creado para dar a otros organismos la posibilidad de especificar sus propios tipos de forma independiente [59]. Con respecto al NFC Forum Well-Known Types, es necesario decir que fue estandarizado por las especificaciones del NFC Forum, que proporcionan la pauta para el procesamiento y representación de los datos. Ellos son [59]:

- Text Record Type: Sólo texto simple, ninguna aplicación específica asignada.
- Uniform Resource Identifier (URI) Record Type: Correo electrónico, direcciones de Internet, números de teléfono u otros códigos de identificación.
- Smart Poster Record Type: Es una extensión del tipo de registro URI, que proporciona información adicional acerca del URI, como íconos o acciones recomendadas.
- Generic Control Record Type: Proporciona una estructura para cualquier actividad de control.
- Signature Record Type: Una firma está prevista para certificar la veracidad de los datos.
- Connection Handover: Ofrece traspaso de una conexión NFC a otra tecnología de comunicación con mayor rendimiento de datos (por ejemplo, Bluetooth).

2.1.4. Aprendizaje Automático

El Aprendizaje automático está relacionado con el campo de la inteligencia artificial, y es una consecuencia natural de la intersección de la informática, la estadística y la ciencia cognitiva; por lo tanto, las definiciones vendrán de todos esos campos. El aprendizaje automático se ocupa de las aplicaciones que mejoran con la experiencia; por lo tanto, se tendrá que ganar un poco de comprensión del aprendizaje, aprendizaje automático, la predicción, el modelado y algoritmos de aprendizaje. La definición formal del aprendizaje automático es “se dice que un programa de ordenador aprende de la experiencia

E con respecto a alguna tarea T y alguna medida de desempeño P , si su desempeño en T , medida por P , mejora con la experiencia E ”[68]. A continuación se definen todos los conceptos subyacentes que son necesarios para entender la estructura fundamental de los problemas de aprendizaje.

Aprendizaje

La palabra “aprendizaje” tiene muchos significados diferentes. Ejemplos de ello son “conocimiento o habilidad que se adquiere por la instrucción o el estudio” “modificación de la tendencia del comportamiento por la experiencia”[69]. La palabra puede ser usada para describir el acto de memorizar algo o aprender hechos a través del análisis, la práctica y la organización del conocimiento. Los métodos de aprendizaje difieren tanto en términos de los tipos de hipótesis con las que trabajan y los algoritmos que utilizan para encontrar una buena hipótesis dados los datos. El objetivo del aprendizaje es encontrar una hipótesis, h , que describe la relación entre las entradas y las salidas. Se tienen un conjunto de hipótesis, y un algoritmo de aprendizaje debe elegir una. Para todos los métodos de aprendizaje, se tiene un conjunto de datos de entrenamiento que se emplea como la base para el aprendizaje y la producción de una salida o respuesta. Los modelos de aprendizaje son ampliamente implementados para la predicción del comportamiento del sistema y predecir las tendencias futuras [70].

El proceso de aprendizaje

[71] Se puede visualizar el aprendizaje como si se estuviera tratando de encontrar la definición de una función dada a un grupo de ejemplos de su entrada y salida. Se tiene un conjunto de ejemplos de entrenamiento, que se componen de variables de entrada x (o características) y salidas y u objetivos. Se alimenta el entrenamiento establecido en el algoritmo y el uso de una hipótesis (h) para mapear x a y (ver Figura 2.3).

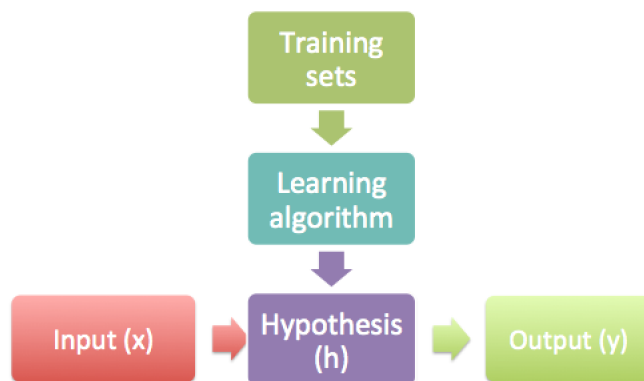


Figura 2.3: Estructura de un método de aprendizaje y su notación. Adaptado de [71]

Pasos a seguir para el aprendizaje: 1. Elegir el tipo de ejemplo de entrenamiento: Este paso debe especificar los datos de entrada para el algoritmo. 2. La recopilación de un conjunto de entrenamiento: El conjunto debe ser característico del mundo real y basado en mediciones. 3. La determinación de la representación de las características de entrada de la función aprendida: La entrada se transforma en un vector de características, que contiene un conjunto de características que son descriptivas del objeto y son una buena representación de lo que se quiere que el algoritmo aprenda. Debe contener información

2.1. Contexto general

suficiente para que el algoritmo pueda aprender. 4. Definición de la estructura de la función aprendida y algoritmo de aprendizaje: El rendimiento del algoritmo de aprendizaje es ajustado y optimizado usando un conjunto de validación (un subconjunto de conjunto de entrenamiento).

Tipos de aprendizaje

[71] Hay tres métodos principales a través de los cuales podemos aprender: supervisado, no supervisado y de refuerzo .

- **Supervisado:** Aprender de ejemplos etiquetados. Dado un conjunto de entrada y la salida deseada correspondiente, predecir salidas para entradas futuras, es decir, encontrar una regla que hace un buen trabajo de predecir la salida asociada con una nueva entrada. Ejemplos de ello son la clasificación (aprender a predecir etiquetas de clase), de regresión (aprender a predecir los resultados con valores reales) y la predicción de series de tiempo.
- **No supervisado:** Aprender de los ejemplos no etiquetados. Sólo entradas dadas, descubre automáticamente las representaciones, las características y la estructura, entre otros aspectos de la entrada. Los ejemplos incluyen agrupar el conjunto de ejemplos en clusters naturales y de compresión, entre otros. Clustering es agrupar patrones de entrada de acuerdo con clusters o agrupaciones naturales. La cuestión es lo que define “agrupación natural” el número de grupos que debe ser utilizado. Estos algoritmos son útiles para el procesamiento de imágenes, la segmentación del mercado, y las relaciones sociales de la red.
- **Refuerzo:** Trial y el aprendizaje de error. Dadas las secuencias de entradas, acciones de un conjunto fijo y recompensas/castigos, aprender a seleccionar las secuencias de acción de una manera que maximiza la recompensa esperada. Un agente interactúa con el mundo haciendo observaciones, toma acciones y es recompensado cuando se ha aprendido correctamente y castigado cuando no. El agente de aprendizaje realiza una secuencia de acciones.

Algoritmos de Aprendizaje Automático

Hay dos tipos principales de algoritmos: los clasificadores y predictores on-line. La principal diferencia entre ellos es que los algoritmos clasificadores toman un conjunto de diferentes valores de las variables de datos de contexto en un punto de tiempo como entradas y derivan una clase (estado contexto) de ellos [72], mientras que los algoritmos en línea toman la historia de los valores de una única variable de datos contexto y pronostican el próximo a venir [73].

Los clasificadores son funciones que permiten obtener el valor de una clase c basado en un conjunto de datos entrenados de entrada de instancias etiquetadas A con el formato $[a_1, a_2, \dots, a_n]$ [72] [74], donde a_i son las características (también llamados atributos).

Los predictores en línea se basan en un proceso estocástico X que, dada una secuencia de símbolos observados $[x_1, x_2, \dots, x_n]$, predice el siguiente símbolo x_{n+1} , [75], donde x_i es el símbolo observado en un punto dado del tiempo.

Hay muchas formas de representar la función de un clasificador y modelar los procesos estocásticos basados en diferentes algoritmos. Se presentan a continuación una muestra representativa de diferentes algoritmos de aprendizaje automático utilizados en la literatura que son muy populares y han demostrado buenas tasas de resultados para la clasificación utilizada en el diagnóstico de una condición médica de los síntomas, para la toma de decisiones y la predicción, para las predicciones

atmosféricas de tormentas eléctricas, por ejemplo, o para crear perfiles precisos de los usuarios que interpretan la información del usuario [76] [70] [77], y la predicción en línea se ha utilizado para la comprensión de datos, análisis de secuencias biológicas, el habla y el modelado del lenguaje, y para el reconocimiento de actividad o la predicción de ubicación [78] [79] [80] [81].

A continuación se describen cuatro tipos de clasificadores de diferentes naturalezas: basados en reglas, basado en redes Bayesianas, basados en instancia, los basados en función lineal y en árboles de decisión.

Clasificadores basados en reglas Los Clasificadores basados en reglas consisten de un sistema conformado por reglas de la forma “si-entonces”, hechos y un intérprete que controla la aplicación de estas reglas, teniendo en cuenta los hechos. Estas reglas se usan para formular las declaraciones condicionales que conforman la base de conocimientos completa, y cada una de ellas asumen la forma de “si x es A entonces y es B ”, donde x es el atributo, A es el valor que se necesita, y es la clase, y B es el nombre de la clase. La parte de la regla antes “then.” es llamada premisa, y la parte que sigue se llama la conclusión [82]. Estos algoritmos pueden ser representados como árboles o tablas.

- En el caso del algoritmo tree C4.5, cada camino desde la raíz a una de sus hojas se puede transformar en una regla simplemente “con la conjunción las pruebas a lo largo de la ruta para formar la parte antecedente, y tomando la predicción de la clase de la hoja como la clase valor”[83].
- Los algoritmos decision table mapean el conjunto de características que se incluyen en la tabla de los casos etiquetados desde el espacio definido por las características [84].

Clasificadores basados en redes Bayesianas Los clasificadores de Bayes aplican la regla de Bayes (Ecuación 2) para calcular la probabilidad de un atributo de pertenecer a una clase en la que se clasifica un conjunto de atributos E en una clase C dada la clase etiquetada del conjunto de datos con la máxima probabilidad [72]. Se asume que todas las características son independientes entre sí [85].

$$Probability(C|E) = Probability(C) * Probability(E|C)(2)$$

Donde C es la etiqueta de clase y E es el conjunto de atributos.

Dos algoritmos clasificadores de Bayes populares son las Redes Bayesianas y Naïve Bayes.

- Los algoritmos de Redes Bayesianas son grafos acíclicos dirigidos con una distribución de probabilidad condicional para cada nodo. Cada vértice en el grafo representa una variable aleatoria, y los bordes son las correlaciones directas entre las variables [85].
- En los algoritmos de Naïve Bayes también se utilizan grafos con una estructura simple que tiene el nodo de clasificación como el nodo padre de todos los otros nodos que no tienen conexiones adicionales [85]

Clasificadores basados en instancia La salida de los Clasificadores basados en Instancia o Clasificadores Lazy es un concepto de función de Descripción que mapea las instancias (atributos pares) a categorías. Una descripción de concepto basado en instancia incluye un conjunto de instancias almacenadas y también puede contener alguna información sobre sus actuaciones pasadas durante la

2.1. Contexto general

clasificación, como el número de predicciones de clasificación correctas e incorrectas. El ejemplo más simple de clasificador basado en instancia es *IB1* que utiliza la función de similitud que se describe en la ecuación 3 para encontrar coincidencias entre una instancia de formación y una instancia de prueba dada [86] mediante el cálculo de la distancia entre los casos (que son atributos pares).

$$Similarity(x, y) = \sqrt{\sum_{i=1}^n f(x_i, y_i)}(3)$$

Donde x y y son instancias (atributos de pares de valores), n es el número de atributos y la función $f(x_i, y_i)$ es igual a $(x_i - y_i)^2$ para los atributos con valores numéricos y es igual a $(x_i \neq y_i)$ para atributos con valores Booleanos y simbólicos [86].

Clasificadores basados en funciones lineales Los clasificadores de función se aplican a los datos que son linealmente separables en el espacio de características en dos clases diferentes, y su objetivo es encontrar la margen del plano hiper que separa los puntos de datos basado en la clase a pertenecer. Hay muchos planos hiper que podrían clasificar los datos, y la mejor elección es el que representa la mayor separación entre las dos clases [87].

Un ejemplo de los clasificadores de funciones lineales son Máquinas de Vectores de Soporte (SVM), que operan mediante la adopción de un conjunto de datos de entrada y predicen a cuál de las dos clases posibles puede pertenecer, un punto de datos es visto como un vector n -dimensional (una lista de n números), y se quiere saber si se pueden separar estos puntos con un hiper plano dimensional ($n-1$). El hiper plano se construye utilizando la función descrita en la Ecuación 4 [87].

$$f(x) = \sum_{i=1}^n (w_i x_i) + b(4)$$

Donde x es el atributo, x_i es la instancia ith , w es el peso, y b es el sesgo.

Clasificadores basados en Árboles de decisión Un árbol de decisión es un clasificador expresado como una partición recursiva del espacio de instancia. El árbol de decisión consta de nodos que forman un árbol de raíz, lo que significa que es un árbol dirigido con un nodo llamado “raíz” que no tiene bordes entrantes. Todos los otros nodos tienen exactamente un borde entrante. Un nodo con bordes salientes es llamado un nodo interno o de prueba. Todos los otros nodos se denominan hojas (también conocido como terminales o nodos de decisión). En un árbol de decisión, cada nodo interno divide el espacio de instancia en dos o más sub-espacios de acuerdo con una cierta función discreta de la entrada de valores de atributos. En el caso más simple y más frecuente, cada prueba considera un solo atributo, de tal manera que el espacio de ejemplo se divide de acuerdo con el valor del atributo. En el caso de atributos numéricos, la condición se refiere a un rango [88].

Los inductores de árboles de decisión son algoritmos que construyen automáticamente un árbol de decisión de un determinado conjunto de datos. Normalmente, el objetivo es encontrar el árbol de decisión óptimo, reduciendo al mínimo el error de generalización. Sin embargo, otras funciones objetivo pueden ser también definidas, por ejemplo, minimizando el número de nodos o minimizando

la profundidad media. La inducción de un árbol de decisión óptimo a partir de un conjunto de datos dado se considera que es una tarea difícil [89]. En términos generales, estos métodos se pueden dividir en dos grupos: de arriba hacia abajo y de abajo hacia arriba, con clara preferencia en la literatura para el primer grupo.

Hay varios árboles de arriba hacia abajo de toma inductores como ID3 [76], C4.5 [90], CART [91]. Algunos consisten en dos fases conceptuales: crecimiento y poda (C4.5 y CART). Otros inductores realizan únicamente la fase de crecimiento. Se debe tener en cuenta que estos algoritmos son codiciosos por naturaleza y construyen el árbol de decisión de arriba hacia abajo de manera recursiva (también conocido como “divide y vencerás”). En cada iteración, el algoritmo considera la partición del conjunto de entrenamiento con el resultado de una función discreta de los atributos de la entrada. La selección de la función más adecuada se realiza de acuerdo con algunas medidas de división. Después de la selección de una división apropiada, cada nodo subdivide más el conjunto de entrenamiento en subconjuntos más pequeños, y hasta que no haya ganado la medida de división suficiente o un criterio de parada no está satisfecho.

2.1.5. SR Basados en conocimiento u ontologías

Un tipo de sistema de recomendación es el que utiliza el conocimiento acerca de los usuarios y de los ítems para aplicar un enfoque basado en el conocimiento para la generación de una recomendación, discutiendo acerca de lo que los ítems cumplen con los requisitos del usuario [58].

El modelado de usuario es típicamente basado en conocimiento o basado en el comportamiento. Los enfoques basados en conocimiento construyen modelos estáticos de los usuarios y ajustan dinámicamente a los usuarios el modelo más cercano. Los enfoques basados en Comportamiento utilizan el comportamiento de los usuarios como un modelo, a menudo utilizando técnicas de aprendizaje automático para descubrir patrones útiles de comportamiento [92].

Las ontologías se utilizan tanto para estructurar la web, como en la búsqueda de categorización de espacio, y para proporcionar una base común para el entendimiento entre los sistemas. En profundidad las representaciones ontológicas también se observan en los sistemas basados en conocimiento, que utilizan las relaciones entre entidades Web (marcadores, páginas web, los autores de páginas, etc.) para inferir hechos sobre situaciones dadas. En este sentido, un sistema de gestión de reglas de recomendación basado en ontologías [93] implementa el sistema de recomendación que combina la gestión de reglas con el fin de mejorar la coherencia funcional del sistema.

Definición de Ontología

Las ontologías proveen una comprensión compartida y consensuada del conocimiento de un dominio que puede ser comunicada entre personas y sistemas heterogéneos. Las ontologías fueron desarrolladas en el área de Inteligencia Artificial (IA) para facilitar el intercambio y reutilización del conocimiento [94]. Una ontología es una jerarquía de conceptos con atributos y relaciones que tiene una terminología consensuada para definir redes semánticas de unidades de información interrelacionadas, proporcionando un vocabulario de clases y relaciones para describir un dominio. Por tanto, una ontología debe ser capaz de proveer un buen entendimiento del dominio que representa, esto incluye relaciones de términos y conceptos, sus definiciones o significados, sus relaciones con cada uno de ellos y las características del dominio [95].

2.1. Contexto general

Componentes de las Ontologías

Los componentes de una ontología varían de acuerdo al dominio de interés y a las necesidades de los desarrolladores. Por lo general entre los componentes se encuentran los siguientes [96] [97]:

- **Clases:** son la base de la descripción del conocimiento en las ontologías ya que describen los conceptos (ideas básicas que se intentan formalizar) del dominio. Las clases usualmente se organizan en jerarquías a las que por lo general se les aplican mecanismos de herencia, es decir un concepto de nivel superior generaliza a otro de nivel inferior [98].
- **Relaciones:** Representan las interacciones entre los conceptos del dominio. Las ontologías por lo general contienen relaciones binarias; el primer argumento de la relación se conoce como el dominio y el segundo como el rango.
- **Funciones:** Son un tipo concreto de relación donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de una ontología.
- **Instancias o individuos:** Representan objetos o elementos concretos de un concepto.
- **Taxonomía:** Conjunto de conceptos organizados jerárquicamente. Las taxonomías definen las relaciones entre los conceptos pero no los atributos de éstos.
- **Axiomas:** Se usan para modelar sentencias que son siempre ciertas. Los axiomas permiten, junto con la herencia de conceptos, inferir conocimiento que no esté indicado explícitamente en la taxonomía de conceptos. Los axiomas definidos en una ontología pueden ser estructurales o no estructurales: un axioma estructural establece condiciones relacionadas con la jerarquía de la ontología, conceptos y atributos definidos; un axioma no estructural establece relaciones entre atributos de un concepto y son específicos de un dominio. Los axiomas se utilizan también para verificar la consistencia de la ontología.
- **Propiedades (Slots):** Son las características o atributos que describen a los conceptos. Las especificaciones, rangos y restricciones sobre los valores de las propiedades se denominan facets. Para un concepto dado, las propiedades y las restricciones sobre éstos son heredadas por las subclases y las instancias de la clase.

Herramientas y APIs de programación de ontologías

Protégé Protégé es un software libre de código abierto implementado en Java, desarrollado en la Universidad de Stanford, que permite la construcción de ontologías de dominio. Es capaz de operar como una plataforma para acceder a otros sistemas basados en conocimiento o aplicaciones integradas, o como una librería que puede ser usada por otras aplicaciones para acceder y visualizar bases de conocimiento. La herramienta ofrece una interfaz gráfica que permite al desarrollador de ontologías enfocarse en la modelación conceptual sin que requiera de conocimientos de la sintaxis de los lenguajes de salida. Protégé ha sido utilizado como el ambiente de desarrollo primario para muchas ontologías, y se ha convertido en la herramienta más utilizada en el mundo para trabajar con OWL [99].

OWL API OWL API [100] es una interfaz Java y la implementación para el Lenguaje de Ontologías Web del W3C (OWL). OWL API también ofrece enlaces específicos para razonadores OWL DL: Fact++ y Pellet. Incluye los siguientes componentes:

- Parser RDF/XML parser
- Parser OWL/XML
- Analizador de sintaxis funcional OWL
- Soporte para la integración con los razonadores como Pellet y FACT++.

Jena Ver apartado Jena, en la sección “Razonadores de Programación Lógica”.

Formalismos y razonamiento con ontologías

Para que los programas software sean capaces de realizar procesos de inferencia sobre la información almacenada en las ontologías, éstas deben de fundamentarse en algún formalismo lógico. La utilización de estos formalismos lógicos, ofrecen la posibilidad de realizar procesos de razonamiento e inferencia que conducen a la adquisición de nuevo conocimiento a partir del ya existente. Así, se puede decir que una lógica define un lenguaje formal para expresar conocimiento acerca de algún dominio y las proposiciones a ser derivadas de este conocimiento, es decir, fija el conjunto de primitivas que pueden emplearse para modelar el contexto, esto es, la sintaxis, y establece la semántica formal de cada proposición en el lenguaje correspondiente (valor de verdad de cada sentencia respecto a cada mundo posible) y está equipada con un “cálculo”, es decir, un procedimiento formal (computable) para derivar nuevos hechos (descritos en el lenguaje correspondiente) a partir de un conjunto de proposiciones dado en el lenguaje. Los tres formalismos lógicos más relevantes son: lógica de primer orden, lógica descriptiva y programación lógica [101].

Las tareas de razonamiento tienen como objetivo la explotación significativa de las bases de conocimiento a través de la validación o la deducción de nuevos conocimientos [102]. Entre estos motores o razonadores se tienen:

Pellet Es un razonador open source para OWL-DL completo con muy buen rendimiento [103], y se encuentra construido en Java. Pellet soporta toda la expresividad de la lógica descriptiva, implementa estrategias de T-Box, soporte nominal (clases definidas por extensión), absorción, ramificación semántica. Su razonamiento individual, de tipos de datos (DataType), y la optimización de las respuestas a las consultas de A-Box, hacen que sea adecuado para aplicaciones web semánticas. Pellet soporta la gama estándar de las consultas derivativas y varios servicios de gestión de razonamiento a través de su propia API, mediante el suministro de enlaces y el apoyo de toolkits comunes (por ejemplo, Jena [104], WonderWeb de OWL API [105] y DIG [106]). También se ha ido más allá tanto en el conjunto estándar de servicios de inferencia (consistencia, satisfacibilidad, clasificación, y la realización) y las recomendaciones sugeridas por la W3C (la coherencia, la vinculación, y respuesta a consultas conjuntas) para introducir varios servicios estándar, que se cree que son casi indispensables para el uso práctico.

El sistema proporciona una fina capa de acceso a través de la interfaz de programación de servicio (SPI), que ofrece funciones convenientes para acceder a los servicios de razonamiento [103]. Este razonador implementa las mejores técnicas de optimización, lo que hace que su desempeño sea bueno,

2.1. Contexto general

en especial cuando debe evaluar ontologías con mayor complejidad y expresividad; sin embargo no es tan eficiente como RacerPro o FaCT ++ en clasificaciones [103].

FaCT++ Es un razonador DL que soporta la lógica SHOIQ(D) y ha sido desarrollado bajo el proyecto europeo “WonderWeb”[107] [108]. Está implementado en C++ y corresponde a una nueva versión del razonador DL FaCT, que fue originalmente implementado en Lisp [35]. FaCT++ es un buen razonador para la T-Box de una ontología. Sin embargo, hay dos desventajas importantes de FaCT++, por un lado, no tiene soporte para otros tipos de dato que no sean string o integer, como sí ocurre, por ejemplo, con Pellet, y por otro lado, tampoco tiene soporte para el razonamiento con la A-Box de la ontología. Entre las ventajas que ofrece FaCT++ se tienen: posee licencia GPL, que todavía tiene mantenimiento y sigue en desarrollo (actualmente se está investigando cómo aumentar la capacidad de razonamiento del motor de inferencia y tiene soporte de OWL2) [108] y trabaja de forma eficiente con la T-Box de ontologías de tamaño grande y mediano.

Racer Pro Por sus siglas, Renamed Abox and Concept Expression Reasoner (RACER) es un razonador DL para la lógica descriptiva SHI, desarrollado en LIPS. Inicialmente fue creado por la Universidad de Hamburgo, pero luego se convirtió en software propietario y su nombre comercial es RacerPro [109] [110]. Soporta OWL DL excepto para los nominales (clases definidas por una enumeración de sus miembros, que implementa como definiciones parciales) y para tipos de datos no estándar; maneja largas A-Boxes en combinación con largas y expresivas T-Boxes y provee servicios de inferencia sofisticados [98] [103].

Programación Lógica

En ésta, se trabaja de una forma descriptiva, estableciendo relaciones entre entidades e indicando no el cómo se deben de hacer las cosas, sino el qué hacer. La programación lógica se basa en un subconjunto de LPO denominada “Lógica de Horn”[111]. Sin embargo, la semántica de la programación lógica es diferente de la LPO. En este caso, la semántica está basada en los modelos mínimos de Herbrand. Por analogía con la formulación implicativa y deductiva, la notación utilizada para este tipo de cláusulas es una lista de literales negados como antecedente y el literal afirmado como consecuente separados por el signo de implicación: $A(x), D(x), E(x) \text{ y } F(x, y)$. De esta forma, un programa lógico consiste en reglas de la forma “si A entonces B”.

El uso de motores de reglas, conocido también como motores de inferencia, es un paradigma general basado en datos, y por tanto, es declarativo para deducir nuevas conclusiones a partir de un conjunto de datos y reglas de inferencia. Una regla de inferencia generalmente se expresa a través de una cláusula “Ifz una cláusula “Then”. Hay dos grandes tipos de sistemas basados en reglas: los sistemas de encadenamiento hacia adelante y los de encadenamiento hacia atrás. El encadenamiento hacia adelante empieza con los primeros hechos y sigue utilizando las reglas para sacar nuevas conclusiones, dados los hechos. En un sistema de encadenamiento hacia atrás, se comienza con algunas hipótesis (o metas) que se están tratando de probar, y sigue buscando reglas que le permiten concluir la hipótesis y establece nuevos sub-objetivos [112].

Lenguajes de reglas La definición de reglas juega un papel importante en los procesos de inferencia de ontologías: de la buena definición de las reglas depende el éxito para la generación de nuevo conocimiento [90]. A través de las reglas se expresa el comportamiento de las instancias dentro de un dominio. Se detalla a continuación algunos de los lenguajes de reglas:

SWRL SWRL, por sus siglas en inglés Semantic Web Rule Language, se basa en la combinación de OWL DL- OWL Lite, OWL DL y RuleML [91]. Para extender el OWL utiliza las reglas Horn-Like (de tipo si.. entonces.. sino...) expresadas en términos de OWL (clase, literales, instancias, propiedades, etc.). Un editor de este lenguaje es Protegé [113] [114]. SWRL ofrece una sintaxis abstracta que incluye axiomas y hechos, por esto, las reglas son parejas formadas por un antecedente (cuerpo) y un consecuente (encabezado). Cada consecuente puede estar formado por cero o más átomos, y éstos pueden ser instancias, literales, variables de instancias o variables de datos [115].

DL Safe Rules Puede considerarse como un subconjunto de SWRL (Semantic Web Rule Language), en la que las variables pueden solo asumir valores concretos. Surgieron como una extensión de OWL DL, con el objetivo de asegurar que un razonador no llegue a un estado de no decisión [116]. Las reglas DL Safe utilizan reglas Horn con predicados binarios y unarios [117]. Estas reglas tienen una expresividad limitada; por ejemplo, no permiten la negación o disyunción en el cuerpo de la regla [96]. Son soportadas por el razonador KAON2.

RuleML Por sus siglas en inglés Rule Markup Language, es un lenguaje para representar reglas de negocio en la web y fue definido por Rule Markup Initiative. Cubre un amplio rango de reglas que incluye reglas de derivación, reglas de transformación y reglas de reacción [118]. La idea de RuleML es permitir realizar tareas de deducción e inferencia sobre la web, mapeos entre ontologías y comportamientos dinámicos como servicios y agentes. La sintaxis de RuleML está basada en las definiciones de XML esquema y su núcleo o kernel es Datalog, que constituye un sublenguaje de la lógica de Horn.

Razonadores de Programación Lógica

A continuación, se presenta una el razonador más habitual para la programación lógica.

Jena Es un framework de Java open source para construir aplicaciones de web semántica sobre modelos RDF [119]. Jena proporciona a los desarrolladores un API para leer y escribir en formatos XML/RDF, N3 y N-Triples (formatos para grafos abreviados) [120]. Adicionalmente, permite el tratamiento del lenguaje OWL, al utilizar el modelo semántico de RDF, y proporciona conexión de forma externa a través del interfaz DIG hacia razonadores DL, como Pellet o FaCT++ [107]. Jena posee la capacidad para conectarse con motores de inferencia externos tipo Plug-in a través de una interface DIG; sin embargo, tiene su propio subsistema de inferencia que consiste en un motor híbrido con encañamiento hacia-adelante y hacia-atrás utilizando el algoritmo RETE (algoritmo de reconocimiento de Patrones). Los razonadores que ofrece son [107]:

- Razonador transitivo, que implementa propiedades transitivas y simétricas de rdfs: subPropertyOf and rdfs: subclassOf [121].
- Razonador para RDF(S)
- Razonador para OWL
- Razonador para DAML (Darpa Agent Markage Language) [122]. Motor de reglas multipropósito, que puede ser utilizado para el procesamiento de RDF, deducción de nuevo conocimiento y la transformación de grafos RDF.+

2.2. Trabajos Relacionados

Esta sección presenta un análisis de los trabajos relacionados más sobresalientes en el área de acción del presente proyecto, resaltando que hasta el momento, no se ha encontrado trabajos que tengan una aproximación igual a la presentada en este documento. Por lo anterior, a continuación se expondrá el estado actual de la investigación en guías de turismo móviles sensibles al contexto, Sistemas de Recomendaciones para el entorno turístico móvil, alternativas de obtención de información contextual basada en ubicación de los usuarios móviles y sistemas de recomendaciones basados en conocimiento ya que son las áreas más cercanas al dominio de la presente investigación.

2.2.1. Guías de turismo móviles sensibles al contexto

La Computación Ubicua ofrece nuevas oportunidades y desafíos en términos de servicios personalizados y sensibles al tiempo, la ubicación y el dispositivo, que se pueden lograr mediante el uso de la personalización, es decir, la adaptación de una aplicación hacia el contexto actual. Las raíces de la personalización son múltiples y se pueden encontrar en las interfaces de usuario que pueden ser adaptativas o incluso inteligentes y de asesoramiento, en el filtrado de la información y sistemas de recomendación, en hipertexto e hipermedia adaptativos y la informática móvil. Uno de los dominios de aplicaciones especialmente adaptadas para proporcionar acceso ubicuo sobre la base de la personalización es el dominio del turismo, permitiendo el acceso con cualquier medio de comunicación, en cualquier momento, desde cualquier lugar. Tales aplicaciones de apoyo al turista en movimiento por medio de los servicios de localización son a menudo llamadas Guías de Turismo Móviles, las cuales proporcionan al turista información personalizada sobre puntos de interés (POI) (por ejemplo, ambiente, atracciones del lugar o la gastronomía), o asistir al turista en la organización de un viaje individual [123].

Una serie de dichas guías de turismo móviles se han propuesto, ofreciendo una amplia gama de funcionalidades con respecto a la sensibilidad al contexto y la adaptación, descritas a continuación, las cuales son relevantes para la presente investigación, debido a la importancia del uso de información contextual en el sistema de recomendaciones:

COMPASS [124] la abreviatura de Context-Aware Mobile Personal Assitant - ofrece a los turistas recomendaciones y servicios sensibles al contexto, específicamente información de restaurantes. El Contexto de Ubicación es considerado como el principal criterio para seleccionar los servicios pertinentes en el entorno cercano del usuario, mientras que el Contexto del usuario comprende un perfil suministrado de forma manual indicando los intereses del usuario que es actualizado por el sistema en forma automática basado en la retroalimentación de los usuarios para un POI específico.

CRUMPET [125] [126] es un proyecto de la UE relativa a la “Creación de Servicios Móviles personalizados fáciles de usar para el Turismo”, basado en la tecnología de agentes. El usuario puede solicitar información y recomendaciones sobre los lugares de interés turístico, restaurantes y excursiones. CRUMPET considera la ubicación, el dispositivo, la red y las propiedades de contexto de usuario. El contexto de ubicación física se presenta en forma de coordenadas del sensor GPS y todas las propiedades de contexto se adquieren de forma dinámica y automática.

El sistema GUIDE [127] [128] se deriva del área de los servicios basados en la localización. El objetivo es proporcionar información actualizada y sensible al contexto de una ciudad a los turistas a través de una PDA. Aunque el objetivo del sistema es la prestación de servicios basados en la localización, un modelo de contexto lógico más amplio en términos de perfiles es proporcionado, distinguiendo entre el llamado contexto personal en términos de la información sobre el usuario (por ejemplo,

las preferencias, ubicación actual y antecedentes de atracciones ya visitadas) y el llamado contexto ambiental, que comprende información sobre lugares de interés (por ejemplo, los vínculos entre las atracciones cercanas, horarios de apertura y cierre, relevancia para los intereses de los usuarios).

Gulliver Genie [129] [130] es un prototipo que resulta de las áreas de inteligencia artificial y sistemas de agentes que se centra en la entrega de contenidos inteligentes teniendo en cuenta la ubicación y las necesidades de los turistas durante su viaje de una manera proactiva. Gulliver considera ubicación, en términos de la posición actual del usuario, la orientación y el movimiento, así como el contexto del usuario y del dispositivo. Un receptor GPS proporciona la ubicación actual del usuario y una brújula electrónica proporciona la orientación del usuario de forma automática. Esta información de contexto es supervisada dinámicamente por un agente, y otro agente gestiona el contexto del usuario que comprende los datos demográficos como la edad, el género, el idioma y la nacionalidad, junto con un perfil de interés que contiene preferencias acerca de la literatura, el arte, entre otros.

Entre otros prototipos de Guías turísticas móviles basadas en contexto se encuentran: LoL@ [131] - la abreviatura de Local Location assistant la cual es una aplicación móvil basada en la ubicación con GPRS/UMTS, proporcionando a los turistas información turística multimedia sobre la ciudad de Viena. MobiDENK [132] - siglas en alemán para los monumentos móviles - soporta tanto la navegación e información actualizada sobre los POIs. El prototipo m-ToGuide [133] se ha desarrollado dentro del Proyecto Europeo IST promoviendo el uso de las redes celulares 2.5/3G con servicios basados en la localización y PinPoint [134] es un marco para el desarrollo de aplicaciones web sensibles al contexto basado en una arquitectura externa y se ha implementado una guía turística para móviles basada en la Web utilizando dicho framework.

En efecto, los anteriores trabajos relacionados orientan claramente sobre la información de contexto que es empleada en un sistema turístico, convergiendo todos en que la ubicación es una de las variables fundamentales para dichos sistemas. Sin embargo, es de resaltar que aunque emplean como enfoque de personalización: sistemas basados en agentes, sistemas basados en ubicación, servicios móviles personalizados, entre otros, no emplean Sistemas de Recomendaciones para guiar al turista.

2.2.2. Sistemas de Recomendaciones para el entorno turístico móvil

Las tecnologías de recomendación (SR) han sido ampliamente investigadas durante los últimos años, las cuales emplean un amplio rango de técnicas estadísticas, Aprendizaje Automático y recuperación de información [135]. En un contexto general algunas investigaciones condensan algunos conceptos, técnicas y algoritmos relevantes [41] en el campo de sistemas de recomendaciones, de igual manera, algunas plataformas como Mahout de Apache [136] o Lenskit [137] han surgido como soporte en la construcción de SR tanto a nivel comercial como académico. Los sistemas de recomendaciones son usualmente clasificados en las siguientes categorías basado en cómo son hechas las recomendaciones: basadas en contenido, en las que se le recomienda al usuario ítems similares a otros que ha preferido en el pasado; colaborativas, se le recomiendan ítems que personas con gustos y preferencias similares han elegido en el pasado, y enfoques híbridos en los que combinan métodos colaborativos, basados en contenido, entre otros [135].

En el ámbito turístico, varios trabajos han abordado temáticas orientadas hacia el diseño conceptual de Sistemas de Recomendaciones Turísticas, que van desde metodologías para el desarrollo de sistemas de recomendaciones turísticos [13], enfoques sistemáticos de revisión del estado del arte de los TSR [138], y técnicas de modelamiento de contexto en RS, junto con las direcciones de investigación futuras [124].

En cuanto a los algoritmos empleados en sistemas de recomendaciones turísticos, algunos trabajos

2.2. Trabajos Relacionados

incorporan correlaciones entre ubicaciones en la técnica de Filtrado Colaborativo basado en su historia de ubicaciones y en la de otros obtenida con GPS como se presenta en [139], así mismo en [140] se propone un método para la generación de itinerarios turísticos en los sistemas de recomendación basados en algoritmos de conocimiento, en [141] se plantea una plataforma de sistema de recomendación con un enfoque híbrido, combinando Filtrado Colaborativo y filtrado basado en contenido en función de la ubicación y tiempo para visitar el destino, en [142] se presenta un enfoque multidimensional (MD) para los sistemas de recomendación y su aplicación en un filtro colaborativo estándar u otras técnicas de recomendación, en [143] se presenta un enfoque único de filtrado que combina información demográfica disponible en los datasets de los sistemas de recomendación y se proponen dos algoritmos híbridos resultantes de mejorar demográficamente el Filtrado Colaborativo basado en Items y basado en Usuarios. Finalmente [45] presenta un enfoque que aprende perfiles de usuario basándose en un marco común que emplea diferentes tipos de información y una combinación de un filtro demográfico, un filtro basado en contenido y un filtro colaborativo.

Por último, aunque las anteriores investigaciones dan luces acerca de los algoritmos de recomendación a emplear en sistemas turísticos, no han sido empleados algoritmos en los que se prioricen las recomendaciones de acuerdo a patrones de consumo turístico, las que finalmente si el usuario decide seguir, se convertirá en una motivación o incentivo para el consumo en el destino.

2.2.3. Alternativas de obtención de información contextual basada en ubicación de los usuarios móviles

La ubicación es una parte esencial de la información contextual, que ha resultado ser útil en muchas aplicaciones, particularmente las de determinación de la posición, navegación, rutas, seguimiento, logística y control de dispositivos de computación ubicua. De hecho, una serie de servicios basados en la localización han sido investigados hasta ahora, pero la mayoría de los servicios existentes inherentemente dependen de determinados sistemas de detección, tales como GPSs y etiquetas RFID [144].

En este sentido, normalmente, la ubicación de un usuario es detectada y luego se realizan suposiciones sobre el contexto más generales con base a dicha posición, y en esta línea varios enfoques se han empleado para determinar la ubicación del usuario, la tecnología GPS es una solución común en ambientes abiertos y al aire libre (outdoor). Sin embargo, se ha demostrado que una señal GPS está disponible sólo el 4,5 por ciento del tiempo durante el día de un usuario típico [145]. Esto sugiere que el promedio de los usuarios pasan mucho de su tiempo en el interior, donde el servicio de GPS está normalmente restringido. El sistema de posicionamiento Wi-Fi es una alternativa eficaz al GPS para interiores [146] [147], aunque este método proporciona un grado razonable de precisión, el mapa de radio debe ser construido fuera de línea a un costo adicional para obtener información sobre la ubicación exacta.

En este sentido, una aproximación denominada LifeMap [148] ha sido desarrollada, la cual es un proveedor de contexto basado en smartphones que funciona en tiempo real, el acelerómetro, la brújula digital, Wi-Fi y GPS para realizar un seguimiento y automáticamente identificar puntos de interés con precisión a nivel interior (indoor). En [21] se presenta otra aproximación que desarrolla y despliega una guía sensible al contexto para la ciudad de Lancaster, Reino Unido, que ofrece viajes a la medida para los visitantes, adaptando su comportamiento a los cambios en la ubicación de un usuario, en función de los intereses del visitante, la ubicación, el tiempo disponible, las limitaciones financieras, limitaciones de movilidad, y las condiciones climáticas locales, adoptando un enfoque centrado en la red inalámbrica 802.11.

Igualmente [22] propone UbiCicero, una guía de museo multi-dispositivo sensible a la ubicación equipada con un lector RFID capaz de explotar de manera oportunista pantallas grandes cuando los usuarios se encuentran cerca. Al tomar en cuenta la información dependiente del contexto, incluyendo la posición actual del usuario y de la historia de comportamiento, así como el tipo de dispositivo disponible, se proporciona información más personalizada y relevante para el usuario, lo que permite una experiencia global más rica.

Sin embargo, una alternativa a las tecnologías de obtención de información contextual a partir de la detección de la ubicación, es NFC, la cual permite una interacción natural del usuario con su entorno y aunque es emergente, ya se han realizado algunas investigaciones en este ámbito, tal como se muestra en [149], que identifica cómo los sistemas basados en NFC podrían utilizarse para mejorar los flujos de trabajo de soluciones móviles y la facilidad de uso, los resultados de este estudio se basan en la ejecución de seis casos piloto en 2005-2007, y concluye que el uso de NFC ofrece tanto ventajas como desventajas y, por lo tanto, para estimar el potencial de la NFC, sus pros y sus contras tienen que ser sopesados para cada caso en cuestión, en [150] se desarrollan técnicas de detección de proximidad seguras basadas en la información recogida por los sensores ambientales disponibles en los teléfonos móviles NFC, tales como datos de audio y luz y [151] muestra un proyecto denominado “Ciudad NFC”, en el cual la tecnología NFC se utiliza para demostrar los servicios instantáneos. NFC hace posible conectar dispositivos electrónicos e iniciar servicios con sólo tocar una etiqueta (o un dispositivo) con un teléfono. En el proyecto de la Ciudad NFC, éste se utiliza en varias aplicaciones experimentales. Una aplicación presentador usa esto para crear una presentación (en un proyector o pantalla) sin tener que conectar cables o transferir archivos de forma manual utilizando memorias USB o similar. Una aplicación de chat establece una charla sencilla habilitando el contacto de dos teléfonos móviles NFC. Una aplicación de fiesta transfiere imágenes y modifica la lista de reproducción de música, tocando una etiqueta NFC con el teléfono. Una aplicación de guía proporciona información y sugieren sitios turísticos de la misma manera.

Específicamente en el ámbito turístico en [152] se propone una solución orientada a ayudar al usuario a encontrar la ubicación de los puntos de interés dentro de la ciudad y desplazarse a través de ellos, el uso de los teléfonos móviles con la tecnología de comunicación de campo cercano (NFC) incorporada y Smart Posters diseminadas a lo largo de la ciudad.

En síntesis, la mayoría de investigaciones se han orientado hacia prototipos que ofrecen servicios de enrutamiento basado en la ubicación geográfica de los usuarios móviles, obtenida típicamente de receptores GPS, o técnicas de seguimiento de ubicación alternativas (Wi-Fi, celular-id, RFID, etc.) [148] [21] [22] y no se han explorado alternativas de interacción de contexto por medio de la tecnología NFC para alimentar un sistema de recomendaciones turístico móvil.

2.2.4. Aproximaciones de recomendación basadas en conocimiento y sensibles al contexto

Los sistemas de recomendación brindan asesoramiento a los usuarios acerca de los ítems que desearan comprar o examinar. Las recomendaciones formuladas por estos sistemas pueden ayudar a los usuarios a navegar a través de grandes espacios de información de descripciones de los productos, artículos de noticias u otros ítems. El tipo más conocido de sistemas de recomendaciones es el tipo de filtrado colaborativo. Estos sistemas agregan datos sobre los hábitos o preferencias de compra de los clientes, y realizan recomendaciones a otros usuarios en función de la similitud de los patrones de compra en general. Sin embargo, un tipo de sistema de recomendación que utiliza el conocimiento acerca de los usuarios y de los ítems para aplicar un enfoque basado en conocimiento para la genera-

2.2. Trabajos Relacionados

ción de una recomendación, razonando acerca de lo que los ítems que cumplen con los requisitos del usuario.

Un sistema de recomendación basada en el conocimiento evita algunos inconvenientes: no tiene un problema de aceleración ya que sus recomendaciones no dependen de una base de valoraciones de los usuarios, no tiene para recopilar información acerca de un usuario en particular porque sus juicios son independientes de los gustos individuales. Estas características hacen que los recomendadores basados en conocimiento no sólo son sistemas valiosos por sí mismos, sino también altamente complementarios a otros tipos de sistemas de recomendación.

Es necesario una ingeniería de conocimiento para la construcción de un sistema de recomendación basada en el conocimiento. Este es el “precio de la entrada inevitable de un enfoque basado en conocimiento, un precio que no se incurre por el método de conocimiento débil, como el filtrado de colaboración o de aprendizaje automático. Sin embargo, estos métodos débiles sufren el problema de aceleración se mencionó anteriormente.

A diferencia de otros sistemas de recomendación, estos sistemas no dependen de grandes masas de datos estadísticos sobre determinados ítems calificados o usuarios particulares. Además, los sistemas de recomendación basados en conocimiento ayudan a los usuarios a explorar y con ello a entender un espacio de información. Los usuarios son una parte integral del proceso de descubrimiento del conocimiento, elaborando sus necesidades de información en el curso de la interacción con el sistema. Una necesidad es suficiente conocimiento general sobre el conjunto de ítems y sólo un conocimiento informal de las necesidades de uno; el sistema sabe acerca de las ventajas y desventajas, los límites y las estrategias útiles de búsqueda en el dominio, es por esto que los sistemas de recomendación basados en conocimiento son fuertemente complementarios a otros tipos de sistemas de recomendación [58].

Por su parte, un enfoque de recomendación novedoso como una alternativa para seleccionar el mejor conjunto de recursos y evitar la sobrecarga cognitiva es presentado en [153], el cual está basado en conocimiento que toma el contexto en consideración para ayudar en la selección de los mejores recursos de acuerdo a la situación actual del usuario.

De manera similar, en [154] se propone un sistema para llevar a cabo recomendaciones adaptativas sensibles a la situación de información para ayudar a los usuarios de teléfonos móviles en un entorno de campus. Su principal aporte es un modelo para describir cualquier contexto de usuario y de la situación o de los recursos, una arquitectura del sistema, sus componentes y las tecnologías subyacentes. Esta arquitectura y el modelo propuesto se utilizan para realizar recomendaciones que se adaptan al interés de usuario, localización, hora y agenda.

Adicionalmente, en [155] se desarrolla un método híbrido de sistemas de recomendación sensible al contexto, basado en ontologías en el entorno de una Infraestructura de Datos Espaciales para usuarios móviles. El componente recomendador del sistema implementa filtros espaciales, semánticos y de colaboración.

Ahora bien, en [156] se presenta un sistema de recomendación inteligente para los turistas interesados en experiencias relacionadas con el sector del vino. El sistema de recomendación y la ontología de dominio permite la clasificación de los ítems, considerando que cada ítem es una instancia de una (o varias) de las clases de nivel más bajo de la ontología y emplea el dominio ontológico para representar las preferencias de los usuarios.

Seguidamente, en [157] se presenta una plataforma de recomendación híbrido que proporciona información acerca de los recursos turísticos en función del perfil de usuario, ubicación, horario y la cantidad de tiempo para visitar puntos de interés aislados o combinados en una ruta. El sistema de recomendación mediante técnicas híbridas combinan el filtrado colaborativo con la recomendación basada en el contenido, y que permite el razonamiento sobre las folcsonomías. Estas técnicas se adaptan

a las necesidades de servicios móviles en el ámbito turístico.

De manera similar, en [158] se presenta un enfoque de un sistema turístico inteligente ubicuo de información basado en ontologías que propone una ontología de dominio turístico que consta de los conceptos de contenidos turísticos y lugares, y un recorrido por la ontología de aplicación para varios servicios turísticos inteligentes.

Así mismo, en [159] se presenta un enfoque basado en ontologías para encontrar recomendaciones razonables para los sitios (Puntos de interés) como restaurantes, hoteles y lugares turísticos. Se extiende una ontología OWL existente con el fin de mantener las relaciones semánticas entre diferentes tipos de sitios. La ontología se utiliza para hacer recomendaciones sobre una aplicación de realidad aumentada para móviles basado en las entradas del usuario en su dispositivo.

En igual sentido, en [160] se propone un enfoque basado en ontologías para la semántica de recomendación de contenidos hacia el e-learning sensible al contexto. El recomendador toma conocimiento sobre el alumno (contexto de usuario), el conocimiento sobre el contenido, y el conocimiento sobre el dominio que se aprendió en consideración. La ontología se utiliza para modelar y representar estos tipos de conocimiento. La recomendación consiste en cuatro pasos: cálculo relevancia semántica, el refinamiento de la recomendación, la generación del camino de aprendizaje, y la recomendación de aumento.

Finalmente, en [161] se propone el sistema de recomendación es el sistema predice la preferencia del usuario para sugerir artículos por el uso de la técnica de la información del sistema de filtrado. Este sistema de gestión de reglas para la recomendación basada en ontologías implementa el sistema de recomendación que combina la gestión de reglas con el fin de mejorar la coherencia funcional del sistema.

En resumen, los enfoques de sistemas de recomendaciones basados en conocimiento emplean ontologías para realizar el proceso de recomendación, y sirven como base para el planteamiento del sistema de la presente propuesta, sin embargo, hasta la presente, no se tiene conocimiento sobre propuestas que lleven estas aproximaciones al dominio de recomendaciones en entornos turísticos móviles usando NFC para incentivar el consumo turístico; lo cual justifica la pertinencia de la presente propuesta.

Capítulo 3

Diseño del Modelo Contextual

Los capítulos 3, 4 y 5 presentan la Propuesta que busca habilitar y maximizar posibles movimientos a realizar por parte de un turista optimizando su tiempo dentro de un destino turístico, lo cual permite incentivar o motivar el consumo turístico del mismo, por medio de un sistema de recomendaciones.

En este sentido, para el diseño del presente sistema se han tomado las Directrices de diseño para un sistema de recomendación presentada en el trabajo [162], dentro del cual se resalta que la construcción de un recomendador presenta dos principales desafíos: 1) Decisiones de diseño del sistema a construir y 2) encontrar la mejor manera de representar los datos para que puedan ser eficientemente usados por el recomendador y proporcionar una mayor precisión que un sistema de recomendaciones simple.

El diseño de un sistema de recomendación significa tomar decisiones que se pueden clasificar en los siguientes dominios:

- Algoritmos: qué métodos de recomendación utilizar?
- Arquitectura: cómo se implementará el sistema, tendrá que ser centralizada o distribuida?
- Perfil de usuario: ¿cuál es el modelo de usuario? ¿Es necesaria la adaptación de perfil?

Estas opciones se ven limitadas en gran parte por el entorno del recomendador. El entorno del recomendador se puede estudiar a través de tres dimensiones: Usuarios: qué son los usuarios, cuáles son sus objetivos? Datos: ¿cuáles son las características de los datos en que se basan las recomendaciones? Aplicación: ¿cuál es la aplicación general del recomendador en el sistema?

Las dos primeras dimensiones se abordarán en el presente capítulo y en el capítulo 4 se aborda el diseño del motor de recomendaciones.

3.1. Diseño

Las aplicaciones context-awareness apoyan de manera inteligente las tareas del usuario, actuando autónomamente en nombre de los usuarios. El comportamiento de las aplicaciones context-awareness no sólo depende de sus estados internos e interacciones con el usuario, sino también del contexto detectado durante su ejecución. Para esto es necesario definir un modelo de la información de contexto, el cual generalmente varía según el tipo de información que puede representar. Aunque algunos modelos tienen en cuenta el estado actual de un usuario, por ejemplo, “en una reunión”, otros modelos tienen en cuenta el medio ambiente físico, por ejemplo, su ubicación. Para ello, se necesita modelar el

contexto que permita capturar diversas características de la información contextual, incluyendo la variedad de tipos, dependencias, la calidad, e historias. Además, se requieren métodos de procesamiento y gestión para apoyar el descubrimiento y la reutilización de la información contextual.

El objetivo de esta sección es conceptualizar un modelo de contexto que pueda ser utilizado por el sistema propuesto.

3.1.1. Modelado de datos en el contexto

Para modelar la información contextual, deben entenderse tanto la definición y el método para procesar datos de contexto. En esta sección, se define el contexto y su conocimiento y se discute cómo agregar datos del mismo para que puedan ser utilizados por el sistema.

Definición de Contexto

Para crear aplicaciones sensibles al contexto, se debe primero entender el significado de contexto desde el punto de vista de un entorno turístico móvil. En este sentido, se exploran las variables o dimensiones contextuales disponibles en dicho entorno. El uso del término contexto, o context-awareness, por lo general se refiere a un tipo general de sistemas que pueden detectar un entorno físico en continuo cambio y proporcionar servicios de interés para el usuario basada en élste [163]. Usando esta definición básica, varios autores [164] [165] [166] [167] se han centrado en diferentes aspectos del context-awareness, tales como su naturaleza, cómo modelar datos contextuales y las interacciones entre los usuarios y el mismo.

- **Definición de contexto.** El contexto es cualquier información pertinente que se puede utilizar para caracterizar una entidad en la interacción con un usuario en un punto dado en el tiempo. En este mundo centrado en el usuario, una entidad puede ser una persona, dispositivo, aplicación, o casi cualquier otra cosa. Este trabajo se centra principalmente en un sistema de recomendaciones en un entorno turístico móvil.
- **Estado de Contexto.** Un estado de contexto es la información prominente que caracteriza cómo se utiliza el sistema de recomendaciones y que puede afectar su funcionalidad. Es la suma de múltiples variables o dimensiones de contexto cuya combinación tiene diferentes significados que pueden afectar al sistema. Una variable o dimensión de contexto es una señal de contexto, como por ejemplo la ubicación.
- **Adquisición de Contexto.** Para la adquisición contexto, se obtiene una instantánea del contexto del usuario mediante la combinación de las dimensiones definidas en el modelo de contexto: ubicación, tiempo, perfil, información del usuario, entre otros (véase la sección 3.1.2).
- **El uso de Contexto.** Se utiliza la información de contexto para enriquecer las recomendaciones generadas por el sistema y aumentar la experiencia del usuario en la misma.

3.1.2. Modelo Contextual

Para conceptualizar el modelo contextual es necesario analizar los datos de contexto existentes y modelarlos para su uso en el sistema, específicamente existen algunas variables o dimensiones que se utilizarán en algunos algoritmos de aprendizaje para inferir y predecir perfiles contextuales. El enfoque del sistema centrado en el turista, hace que se deba considerar la interacción del mismo con

3.1. Diseño

un entorno turístico ubicuo basado en la tecnología NFC. En general, estas interacciones podrían incluir información de ubicación detectada por medio de NFC, información de tiempo, intereses del usuario, características propias de las actividades turísticas a ser recomendadas, entre otros. Es por esto que para construir el sistema se debe tener un modelo de contexto apropiado para representar, manipular y acceder a la información de contexto.

Existen diferentes enfoques para modelar la información de contexto, dos ejemplos comunes son: el enfoque basado en roles de objetos, que es un método para el diseño y consulta de modelos de contexto a nivel conceptual; y el enfoque basado en la ontologías, que utiliza las ontologías, que se definen como “una especificación formal y explícita de una conceptualización compartida” [168], para describir los elementos de contexto y sus relaciones. La información contextual en este trabajo se modela utilizando el enfoque basado en ontologías, ya que es muy adecuado para el intercambio de conocimiento. Dado que proporciona una especificación formal de la semántica de datos de contexto y esto es especialmente importante en entornos móviles y ubicuos donde diferentes entidades heterogéneas y distribuidas deben interactuar para el intercambio de información de contexto de los usuarios [169].

Con el fin de definir mejores modelos de ontologías es importante considerar la reutilización de ontologías existentes, ya que ayuda a reducir al mínimo la introducción de errores y a mejorar la reutilización del modelo desarrollado. En este sentido, algunas ontologías existentes se emplearon y se extendieron para completar las especificidades de dominio del presente proyecto.

Principales variables de contexto

En la conceptualización del contexto de presente trabajo se analizaron y agruparon los datos de contexto que comparten el mismo tipo de información contextual, combinando las variables o dimensiones en tres grupos, que corresponden con la tarea que realiza el turista, la ubicación y el tiempo en que se produce la interacción.

- **Cuándo:** El día (o semana o fin de semana) y el marco de tiempo (mañana, tarde, atardecer o noche) se etiquetan como la fecha y tiempo [38].
- **Dónde:** ubicación física del turista obtenido a partir de la interacción con la etiqueta NFC del sitio que se encuentra visitando.
- **Qué:** Todas las demás variables o dimensiones que tienen que ver con las tareas que realiza el sistema de recomendaciones, específicamente lo que debe tener en cuenta para proporcionar las actividades turísticas a realizar en dicho momento.

Cuándo: Fecha y hora: La dimensión de tiempo es generalmente capturada implícitamente y proporciona información sobre la hora y la fecha. El grupo de fecha y hora es importante debido a que el contexto general consiste de muchas descripciones parciales de una situación que varían con el tiempo. En el presente proyecto, se va a usar como una variable de contexto, dividiendo un día en diferentes períodos de tiempo (es decir ranuras: mañana, tarde, atardecer y noche), en la que por lo general se diferencian las actividades humanas. De manera análoga para la fecha, se empleó el atributo día de la semana (Lunes, Martes, Miércoles, Jueves, Viernes, Sábado, Domingo) que se pueden clasificar en si es un Día de la semana (WeekDay) o un Día de Fin de semana (WeekendDay), tal como se muestra en las Figuras 3.1 y 3.2.

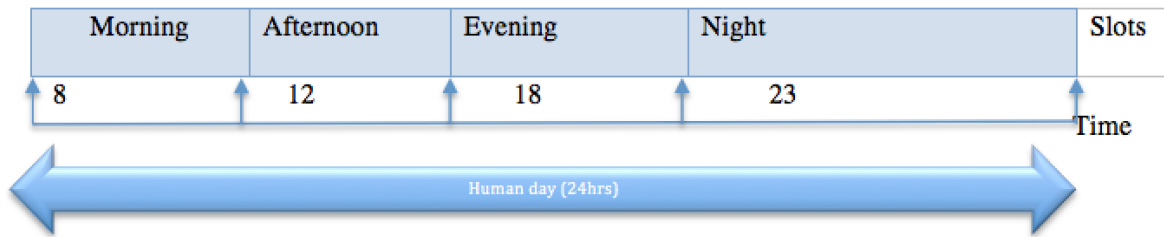


Figura 3.1: Partición del día en zonas horarias.

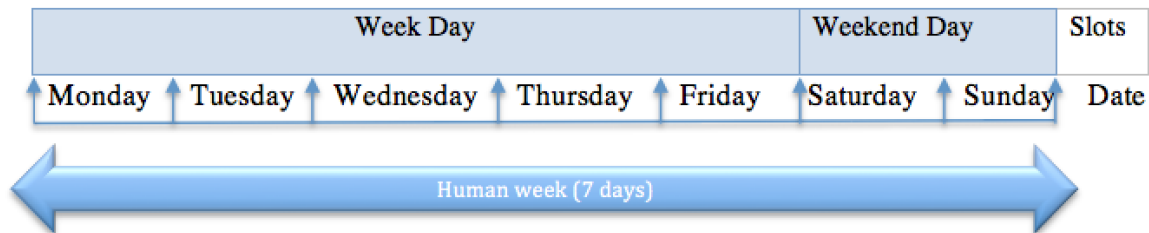


Figura 3.2: Partición de la semana.

Dónde: Ubicación: La ubicación del usuario generalmente es detectada a menudo implícitamente a través de GPS o sensores de localización Wi-Fi, o una combinación de ambos [170]. Esta dimensión modela la información necesaria sobre la posición geográfica del usuario actual, que pueden ser coordenadas geométricas o simbólicas. Las coordenadas geométricas comprenden información sobre los puntos espaciales tales como latitud, longitud y altitud y se modelan mediante la ontología de Geo Posicionamiento W3C WGS84. Las coordenadas simbólicas son identificadores para las ubicaciones (por ejemplo, número de sala, el nombre del auditorio) y se modelan a través de los conceptos de Entorno y Tipo de Entorno, que se refieren a la información sobre el entorno actual (por ejemplo, el aula, la biblioteca) y el tipo de este Entorno (por ejemplo, público, privado), respectivamente. Para el presente proyecto, la ubicación del usuario es obtenida por medio de las Smart Tags, que representan una ubicación específica como se muestra en la figura 3.3, en este caso, el punto de interés (o lugar) turístico, y se recuperan las coordenadas físicas del mismo. Esta dimensión mantiene importantes relaciones con el usuario y las dimensiones de lugares.

Qué: Tareas del sistema de recomendación: En este ítem se clasifican las demás dimensiones relacionadas con las tareas que realiza el sistema de recomendaciones, relacionadas con disponibilidad de tiempo, intereses, gustos y preferencias del usuario, así como también el tipo de actividades a recomendar, el horario de disponibilidad, entre otras.

Modelado de variables o dimensiones de contexto

Como se muestra en la figura 3.4, existen varias variables o dimensiones que deben ser tenidas en cuenta en el momento de realizar las recomendaciones, principalmente tienen que ver con los gustos y preferencias del usuario, su disponibilidad y el horario y tipo de actividades a recomendar en dicho

3.1. Diseño



Figura 3.3: Etiqueta NFC representativa de la ubicación de un Lugar Turístico.

momento.

El elemento principal del modelo es el usuario, que está relacionado con las actividades a recomendar, el perfil contextual, su tiempo disponible, sus intereses y con cinco dimensiones de la información de contexto: la ubicación, el tiempo, la tipología, el motivo de viaje y el rango de edad. Por otro lado las actividades a recomendar se realizan en un Sitio o Punto de Interés y cuentan con una disponibilidad (horario) y un duración, cada una de estas dimensiones y algunas de sus relaciones se resumen en la siguiente figura y se explican mejor a continuación.

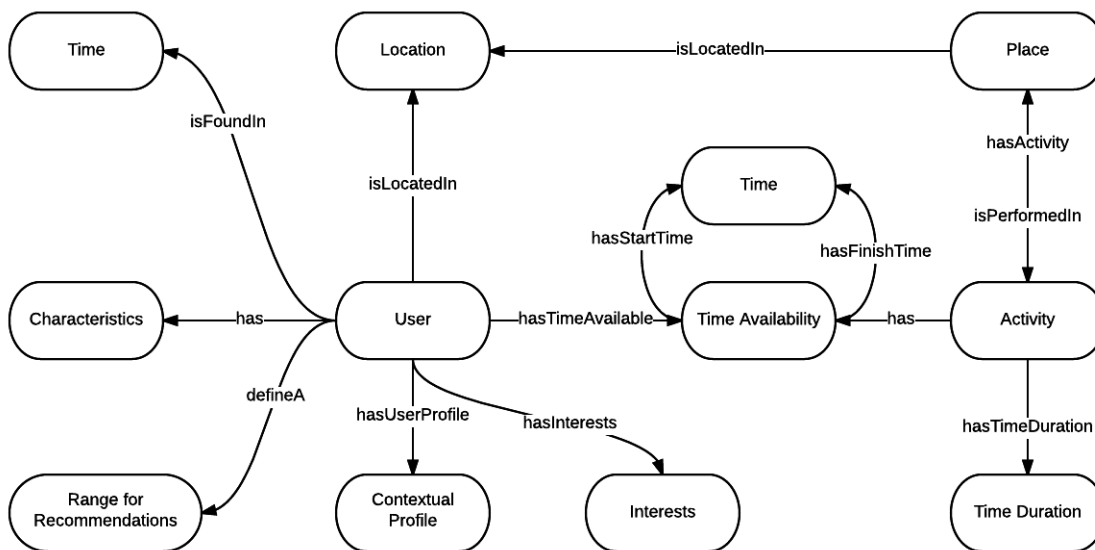


Figura 3.4: Modelo de variables o dimensiones contextuales.

Estas variables o dimensiones proporcionan varios puntos de vista complementarios en las interacciones del usuario. Por ejemplo, el turista puede tener un motivo de viaje Eventos, haber viajado de manera individual y tener más de 50 años, definir 8 horas como tiempo disponible para realizar la

visita y definir que está interesado en turismo relacionado con Arte & Entretenimiento y Colegio & Educación. La Tabla 3.1 enumera varias variables de contexto y los valores que pueden tomar.

| Variable Contextual | | Descripción | Valor |
|---------------------|----|--|-----------------------------|
| Motivo Viaje | E | Eventos | Si o No (1 o 0) Una opción |
| | TN | Trabajo/Negocios | |
| | TO | Turismo/Ocio | |
| | VF | Visita a Familiares | |
| | O | Otros | |
| Tipología | FA | Familia | Si o No (1 o 0) Una opción |
| | G | Grupo | |
| | I | Individual | |
| Rango Edad | ME | Menor de 18 | Si o No (1 o 0) Una opción |
| | J | 18 a 30 | |
| | A | 30 a 50 | |
| | AM | Mayor a 50 | |
| Ubicación | UU | Ubicación de usuario | Coordenadas geográficas |
| | UA | Ubicación de actividad | Coordenadas geográficas |
| Hora del día | H | Hora | Hora en formato de 24 horas |
| Rango | R | Rango definido por el usuario para las recomendaciones | Kilómetros |
| Tiempo disponible | TD | Disponibilidad de tiempo del usuario | Tiempo definido en horas |
| Duración | DA | Duración de la actividad | Tiempo definido en horas |
| Intereses | AE | Arte y Entretenimiento | Si o No (1 o 0) |
| | CE | Colegio y Educación | Si o No (1 o 0) |
| | VN | Vida Nocturna | Si o No (1 o 0) |
| | AL | Aire libre | Si o No (1 o 0) |
| | T | Tiendas | Si o No (1 o 0) |
| | O | Otros | Si o No (1 o 0) |

Tabla 3.1: Valores que pueden tomar las Variables contextuales.

3.1. Diseño

A continuación se explican en detalle cada una de las variables o dimensiones contextuales:

- **Dimensión de características:** esta dimensión hace referencia a la información personal del usuario que se emplea para la predicción del perfil contextual. Para el caso del presente trabajo, como está enmarcado dentro de los datos obtenidos del proyecto de Trazabilidad Turística del Departamento del Cauca, dentro de los parámetros que caracterizaban a un usuario estaban: tipología, motivo de viaje, nacionalidad, país, ciudad, días de visita, posible gasto, rango de edad, género y transporte, de los cuales fueron elegidos 3: tipología, motivo de viaje y rango de edad como variables contextuales a emplear para predecir el Perfil del Usuario. Esta dimensión se explicará con mayor detalle en la sección 3.1.2.
- **Dimensión de Perfil contextual:** Ver en detalle la sección 3.1.2.
- **Dimensión de Intereses:** Ver en detalle la sección 3.1.2.
- **Dimensión de Actividades:** Hace referencia a las actividades turísticas a recomendar en el sistema, dentro del dominio turístico y para el modelo de Datos, los diferentes tipos de actividades a realizar en los sitios para realizar las recomendaciones, se empleó la ontología denominada ETP-Tourism [171], de las cuales fueron escogidos los siguientes:

| Type of Activity | Subtype |
|------------------|--|
| Adventurer | BoatExcursion DesertExcursion Fishing Hiking ScudaDiving |
| Recreation | Dancing Shopping Sightseeing Sunbathing |
| Sport | Byking Golf Jogging Swimming Waterskiing |
| Training | Craftsmanship LanguageCourses MusicLessons |
| Wellbeing | Massage TaiChi Yoga |

Tabla 3.2: Tipos de Actividades

Se agregaron los siguientes tipos adicionales: Art, Culture, Education y Religion.

- Dimensión de Sitios o Puntos de Interés: La dimensión de Lugares hace referencia a los Puntos de Interés turísticos que el turista puede visitar y se encuentran geográficamente referenciados y etiquetados con las Smart Tags, las cuales poseen la información de ubicación que permite situar al usuario.
- Dimensión de Tiempo Disponible: Se refiere al rango de tiempo con el que cuenta el usuario para realizar la visita en la ciudad y el horario en que se encuentra abierto el Sitio o Establecimiento turístico para el cual está disponible su visita.
- Dimensión de Duración: Está directamente relacionada con el tiempo que gastaría el usuario en realizar una actividad, específicamente se usa para maximizar el tiempo durante la visita del turista en la ciudad.
- Dimensión de Rango para las recomendaciones: es el radio en kilómetros alrededor definido por el usuario para el que desea visualizar las recomendaciones.

A continuación se muestra el vector de contexto que resulta en la Tabla 3.3.

| Week Day | Time | Contextual Profile | Interests | Location | Time Availability | Range for Recommendations |
|----------|------|--------------------|-----------|----------|-------------------|---------------------------|
|----------|------|--------------------|-----------|----------|-------------------|---------------------------|

Tabla 3.3: Vector de Contexto

Diseño del Modelo de Datos

El modelo de datos se refiere a las características de los datos en que se basan las recomendaciones, estudiando cuidadosamente los ítems que el sistema va a manipular. Generalmente cuando los ítems y sus descripciones ya existen previamente, el diseño del sistema de recomendación tiene pocas posibilidades de influir en éstos o cambiarlos. En este punto es importante destacar que es necesario identificar las principales características de los datos que pueden influir en el diseño y los resultados del sistema de recomendación.

Dentro de las características a identificar se encuentran: los tipos de datos que el recomendador deberá manipular (estructurados, semi-estructurados o no estructurados), calidad y cantidad de los mismos (alta, media o baja), volumen de los ítems (muchos, pocos), diversidad de los ítems (heterogénea u homogénea), cambio constante de los ítems o estabilidad de los mismos, valoraciones o calificaciones de los usuarios de manera implícita o explícita, y si son binarias o no [162]:

Tipo de datos:

- Datos no estructurados: en este caso un ítem puede ser representado únicamente por datos de tipo no estructurado, que se refiere a la información que no es fácilmente utilizable por un programa informático, por lo que se requiere un pre-procesamiento para extraer palabras clave significativas, o conceptos que ayudan a distinguir cada ítem de otro.
- Datos semi-estructurados: los ítems son descritos generalmente por varios metadatos genéricos, que corresponden a las principales características del artículo y el texto de forma libre.

3.1. Diseño

- Datos estructurados: los ítems pueden ser descritos por medio de un modelo bien estructurado que podría pertenecer a una norma o un estándar de facto.

La calidad/cantidad de los metadatos:

- Calidad: para el diseño del sistema de recomendaciones es necesario conocer la calidad de los metadatos disponibles. En general, una descripción de metadatos del ítem se considera de alta calidad si ésta permite que un ítem pueda ser distinguido de otro, en este sentido, se debe entender el nivel adecuado de detalles en la descripción de los ítems para capturar las diferencias entre fuentes de información.
- Cantidad: La cantidad de metadatos es un factor importante a tener en cuenta: muy pocos metadatos pueden llevar a recomendaciones inexactas, mientras que demasiados metadatos pueden conducir a procesamiento inútil. Además, la descripción de metadatos puede variar en términos de profundidad (grado de precisión) y ancho (variedad de descripción). El riesgo con la descripción superficial es proponer ítems a los usuarios que no se corresponden exactamente con lo que esperan.

Volumen de ítems: Además de la cantidad de metadatos por ítem, es necesario tener en cuenta el volumen de ítems en el conjunto de datos, ya que representa un factor importante en la determinación de la elección del sistema de recomendación.

Distribución de los ítems: También es esencial tener en cuenta cómo los ítems se distribuyen entre el conjunto de datos. Por ejemplo, si en un conjunto de datos de películas, una proporción muy alta de los ítems se anota con el concepto de “Acción”, estos metadatos pueden no ser lo suficientemente discriminativos (demasiados ítems de contenido seleccionados si al usuario le gustan las películas de acción, demasiado pocos si no lo hace). En estos casos, y si es posible, el nivel de profundidad de la anotación de los datos debe ser repensada con el fin de obtener mejores metadatos de calidad.

Naturaleza y estabilidad de los ítems del conjunto: Las noticias, libros o programas de televisión son intrínsecamente diferentes, y por lo tanto los usuarios no se comportan de la misma manera en función de la naturaleza de los ítems. También se debe considerar la estabilidad del conjunto de los ítems, es decir, con qué frecuencia se introducen nuevos ítems o ítems que desaparecen.

Valoraciones de los ítems por los usuario: Otra consideración importante es tomar en cuenta las valoraciones de los usuarios o no. Si no hay puntuación de usuarios en relación con los ítems del sistema, esto excluye toda la familia de los métodos de filtrado colaborativo. Las calificaciones de los usuarios acerca de los ítems pueden ser explícita (por ejemplo en Amazon, los usuarios pueden indicar lo mucho que disfrutaron de un libro o un CD), y puede ser expresado en diferentes escalas (como binario o multi-nivel). Si esto no está directamente disponible, es posible calcular indicadores de realimentación implícita (por ejemplo, el hecho de comprar un ítem puede ser considerado como un indicador implícito de interés del usuario), que se puede utilizar ya sea en algunos algoritmos de recomendación (como el filtrado colaborativo), o como una entrada a un módulo de adaptación del perfil de usuario que actualizará intereses de los usuarios en base a gustos y disgustos de ítems específicos.

Conjunto de Datos

En el análisis contenido en este capítulo se emplearon los datos que fueron obtenidos en la experiencia del Estudio de caso realizada bajo el Proyecto de Trazabilidad Turística de la Universidad del Cauca con las siguientes características:

- Usuarios: Registro de Usuarios por medio de una Tarjeta Turística actuando como identificación. La información obtenida de un usuario es: tipología, motivo de viaje, nacionalidad, país, ciudad, días de visita, posible gasto, rango de edad, género y transporte.
- PATs (Puntos de Atención al Turista): Sitios a visitar por parte del turista, la gran mayoría de estos puntos están constituidos por hoteles, restaurantes, artesanías y sitios de interés cultural y de valor histórico. La información de éste se caracterizó por nombre, latitud, longitud.
- Número de Visitas: 1742 registros.
- Número de Usuarios: 1246 registros.
- Número de PAT's: 21 registros.
- Ratings Binarios: Presente o Ausente
- Este conjunto de datos es disperso ya que se obtuvieron máximo 5 registros por usuario.

Modelo de datos

Tipo de datos: Se decidió optar por los datos semi-estructurados. Eso significa que los sitios son descritos por los metadatos genéricos correspondientes a sus características y también un poco de texto libre (descripción). Por ejemplo, un restaurante puede ser clasificado por categoría y puede tener varios atributos que lo describe (título, tipo de turismo al que pertenece).

Calidad / Cantidad de metadatos: La calidad general de los metadatos es medio. La estructura de los datos (semi-estructurada) permite capturar una gran cantidad de detalles, pero no con la mayor precisión los datos estructurados.

Volumen de sitios: El volumen de sitios es medio, ya que Popayán no cuenta con una cantidad enorme de sitios a visitar, además se empleará la base de datos con la que se cuenta del Proyecto Trazabilidad Turística.

Ratings de elementos de usuario: Las calificaciones o ratings de los usuarios para el sistema serán implícitas y unarias, es decir si un usuario visita el sitio se calificará como 1, pero para el usuario será explícitamente ya que existirá una sección en la aplicación donde podrá calificar la recomendación como tal en una escala de 1 a 5.

Existirá una sección en la aplicación que le permitirá al usuario indicar explícitamente qué tipo de Turismo le interesa (relacionado directamente con los patrones de consumo a maximizar).

Dimensiones del rating: Será unidimensional ya que se manejará un solo rating o calificación para cada ítem.

3.1. Diseño

Timestamp en los ratings: Se realizará la inclusión de información de tiempo en los ratings con propósitos históricos.

Persistencia de las recomendaciones desplegadas al usuario: Se ha definido el formato para un archivo de log que almacenará todas las interacciones.

Densidad de los ratings (porcentaje promedio de ítems rateados calificados por usuario): En principio baja, ya que el dataset es disperso, pero con tendencia a incrementarse si los usuarios califican las actividades recomendadas.

Distribución de las propiedades del dataset: Más usuarios que ítems y con máximo 5 calificaciones por usuario.

Diseño del Modelo Contextual de Usuario

Es necesario y fundamental entender al usuario completamente para el éxito de cualquier sistema de recomendación. En esta sección se discuten las propiedades del lado del usuario que pueden tener un impacto en el diseño y las opciones que se deben tener en cuenta en el momento de diseñar el sistema de recomendaciones. La identificación de esas propiedades se reduce a entender quiénes son los usuarios y cuáles son las expectativas y metas que se encuentran detrás de las motivaciones de los mismos en el momento de utilizar el sistema recomendaciones.

Para entender quiénes son los usuarios se deben tener en cuenta los tres principales aspectos: comprensión de sus características de identificación clave, niveles de habilidad y experiencia previa con sistemas similares. Para el caso del presente trabajo, el aspecto más importante es:

Identificar las características de los usuarios La identificación de un grupo de usuario se puede realizar a partir de la información demográfica, como la edad y el género, área de trabajo, nacionalidad, idiomas que hablan, y profundos conocimientos cualitativos de investigación del usuario son un importante punto en el desarrollo de los modelos de usuario para un recomendador. A partir de la comprensión de estos factores se puede construir una relación con los usuarios y obtener una apreciación de sus necesidades. Así por ejemplo el desarrollo para grupos de usuarios puede permitir (1) la construcción de recomendadores simples basados en la demografía, esto se utiliza comúnmente en soluciones publicitarias focalizadas en agrupar los clientes en segmentos; (2) definir los estereotipos de los usuarios: técnicas para estereotipar permiten la definición de un conjunto de características diferenciales para un grupo de usuarios; cuando un nuevo usuario se introduce en el sistema, se puede asignar a un estereotipo predefinido, en base a sus datos personales, lo que permite la activación de un conjunto de preferencias predeterminadas que pueden ser perfeccionadas en el tiempo gracias a métodos de adaptación de perfil de usuario. En este sentido, las soluciones personalizadas explotan las características del usuario que pueden ser utilizadas en combinación con técnicas más sofisticadas para proporcionar un primer paso sencillo en un proceso de filtrado híbrido, o para un proceso de algoritmo de filtrado basado en contenido mediante el uso de perfiles estereotipados.

Seguidamente para entender las motivaciones, expectativas y objetivos del usuario, en el diseño del motor de recomendación se deben identificar las tareas de los usuarios y entender si la aplicación las soporta. Identificar y comprender la motivación usuario puede dar lugar a mejoras fundamentales de recomendación y de la experiencia del usuario. Las percepciones de los usuarios capturadas dentro

del diseño del modelo de usuario permiten considerar el posible rango de tareas que la aplicación deberá soportar.

En la mayoría de los casos el usuario podrá tener muchas motivaciones para el uso de un sistema, y en el proceso del diseño del sistema de recomendaciones se deben considerar las formas de encontrar la naturaleza del objetivo del usuario, ya sea explícita o implícitamente. Existen algunos casos en los que la tarea objetivo no es perceptible (por ejemplo, porque la aplicación está dedicada al entretenimiento). En tales casos puede ser difícil construir un buen recomendador, ya que la satisfacción del usuario puede estar más fuertemente relacionada con factores externos como el estado de ánimo del usuario, lo cual se encuentra por fuera del control del sistema. En este caso, la extensión, la cantidad o la casualidad pueden ser las cualidades más deseables del recomendador.

Modelo de Usuario

A continuación se muestra el modelo de usuario diseñado:

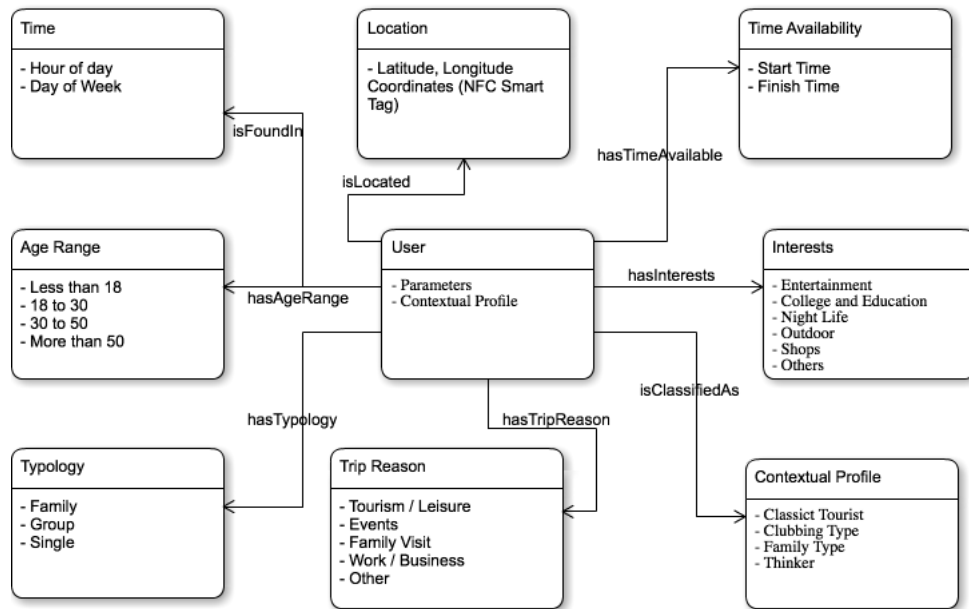


Figura 3.5: Modelo de Usuario propuesto.

3.1. Diseño

Características de los usuarios Para el presente proyecto las dimensiones del usuario que tienen una importancia primordial basado en el conjunto de datos existentes y en la interacción con su contexto a través de la tecnología NFC son:

- Ubicación: determinada por medio de coordenadas latitud y longitud de la ubicación geográfica de la etiqueta NFC, la cual representa la localización física del sitio que el usuario está visitando en ese momento.
- Dimensión de información de tiempo: relacionada con la hora del día y el día de la semana.
- Dimensión de Motivo de viaje: Dentro de las características del turismo se encuentran los elementos motivadores del viaje: “ocio, negocios, salud, descanso, entre otros”. En este sentido, de acuerdo a las opciones definidas previamente en el conjunto de datos utilizado, los motivos o razones de viaje están definidas como: Turismo/Ocio, Eventos, Visita Familiares, Trabajo/Negocios, Otro.
- Dimensión de Rango de edad: Hace referencia a la edad del turista y según lo que se tiene definido en el conjunto de datos, la edad el turista puede estar en los siguientes rangos definidos: Menor a 18, 18 a 30, 30 a 50, Mayor a 50.
- Dimensión de Tipología: Esta dimensión se refiere a la forma en que viaja el turista, en este sentido según el conjunto de datos, el turista puede viajar en Familia, en Grupo o Individual.
- Dimensión de Intereses: Esta dimensión se relaciona directamente con los patrones o características de consumo ajustables para motivar o incentivar consumo turístico, y en este sentido un patrón de consumo es la relación con los bienes y servicios que caracterizan un estilo de vida. Bajo esta premisa se han definido los intereses de un turista basado en las categorías en las que se clasifica un establecimiento, tomadas de la clasificación que realiza Foursquare: Arte y Entretenimiento, Colegio y Educación, Vida Nocturna, Aire libre, Tiendas, Otros.
- Dimensión de su tiempo disponible: Esta dimensión se relaciona con el tiempo en horas con el que el usuario cuenta para una visita, y puede ser configurado por el mismo y variado en cualquier momento mediante la aplicación.
- Dimensión del perfil contextual: Esta dimensión hace referencia al perfil en el cual el turista es clasificado por el sistema, se han definido los siguientes perfiles: Turista clásico, Nocturno, Familiar y Pensador. En la sección 4.5.1 se definirá en detalle el proceso seguido para realizar dicha clasificación.

Motivaciones de los usuarios y expectativas: Dentro de las expectativas de los usuarios está el poder encontrar servicios personalizados y relevantes mediante los dispositivos móviles, gracias a su naturaleza ubicua que los convierte en una plataforma atractiva para asistir a los turistas a elegir los puntos de interés para visitar en función de su ubicación física y de otros parámetros contextuales. Los objetivos de los turistas son: al llegar a un destino turístico poder optimizar su tiempo disponible gracias al sistema, el cual le proporcionará recomendaciones de actividades que se puedan realizar en dicho intervalo de tiempo, de igual forma poder visualizar la oferta turística con la que cuenta un destino gracias al sistema, lo que incrementará el número de visitas que realizaría si llegara al destino sin conocerlo y finalmente poder observar recomendaciones acordes a sus gustos, preferencias, intereses y perfil turístico.

De esta forma, lo que se espera es que las expectativas de los turistas aumenten cuando descubren progresivamente los beneficios de sistema de recomendaciones.

3.2. Resumen

A lo largo de este capítulo se presentó la conceptualización del modelo de contexto a ser utilizado por el sistema propuesto, analizando los datos de contexto existentes y su posible modelado para su uso en el sistema, específicamente considerando la interacción del usuario con un entorno turístico ubicuo basado en la tecnología NFC. En este sentido se presentaron las dimensiones o variables contextuales a ser tenidas en cuenta en el momento de realizar las recomendaciones, principalmente en relación con los gustos y preferencias del usuario, su disponibilidad y el horario disponible y tipo de actividades a recomendar en dicho momento y para luego presentar el diseño del modelo de datos (relacionado con las características de los datos en que se basan las recomendaciones o ítems que el sistema va a manipular) y de usuario (relacionado con las propiedades del lado del usuario que conllevan a entender quiénes son los usuarios y cuáles son las expectativas y metas) a considerar.

Capítulo 4

Enfoque del Sistema de Recomendaciones Propuesto

Este capítulo propone un enfoque de recomendación diseñado para motivar o incentivar el posible consumo de oferta turística por medio de la habilitación y maximización de posibles movimientos a realizar por parte de un turista optimizando su tiempo en un destino, y generando el modelo conceptual del sistema de recomendaciones a construir para dar planteado en el objetivo específico 2.

Los trabajos más importantes presentados en la sección 2.2.2 que se abordaron durante la etapa de generación de base inicial de conocimiento y a partir del análisis de los trabajos mencionados y la discusión generada en torno a ello en los diferentes espacios de socialización, se generan las siguientes reflexiones:

- Los sistemas de recomendación basados en el conocimiento utilizan la información de dominio para recomendar elementos que asisten a las necesidades y preferencias del usuario.
- Los sistemas sensibles al contexto del usuario actual permiten el uso de la información de contexto, además del conocimiento del dominio y los intereses del usuario, añadiendo una dimensión extra de análisis para realizar recomendaciones. Esto conduce a recomendaciones más específicas, que contribuyen a evitar la sobrecarga cognitiva usuario.
- Los sistemas que utilizan la información de contexto y el perfil de usuario en apoyo de su conducta adaptativa se conocen como sistemas adaptativos sensibles al contexto y se han utilizado ampliamente en entornos ubicuos que son ambientes provistos de muchas computadoras disponibles que trabajan de forma invisible para el usuario.

A partir del estudio realizado, se hace un planteamiento del enfoque de recomendación para motivar o incentivar el posible consumo de oferta turística por medio de la habilitación y maximización de posibles movimientos a realizar por parte de un turista optimizando su tiempo en un destino. A continuación se presenta el esquema con las fases correspondientes a desarrollar:

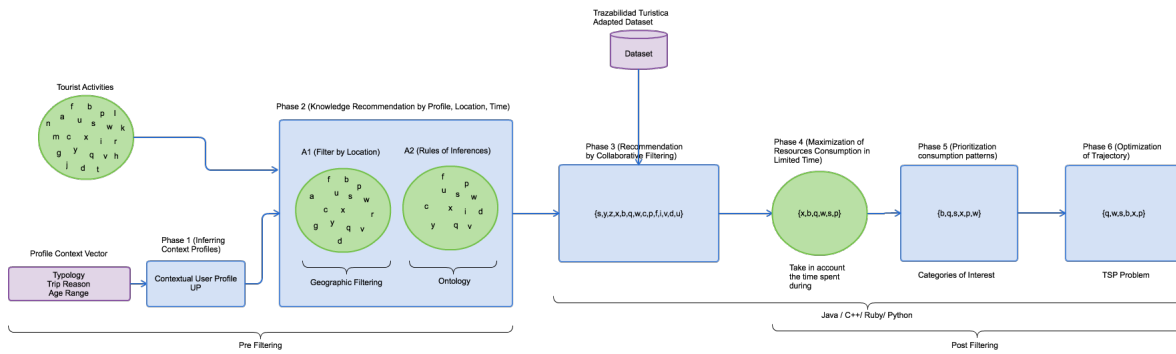


Figura 4.1: Enfoque del sistema propuesto.

A continuación se describen cada una de las fases del enfoque de recomendación propuesto.

4.1. Diseño del Modelo de Aplicación

Un sistema de recomendaciones es una pieza compleja de software y es por naturaleza parte de un gran sistema. Éste puede ser una característica menor o el punto principal; la aplicación puede pre-existir o ser construida junto con el recomendador, pero en cualquier caso el diseño de un sistema de recomendaciones debe estar integrado dentro del diseño de la aplicación que lo contiene. Esta sección estudia los principales factores de la aplicación que contiene el sistema y que pueden influenciar en el diseño del recomendador sobre dos líneas principales: el rol del recomendador y la influencia de la implementación de la aplicación. En orden de entender el rol del sistema de recomendaciones se debe dar respuesta a las siguientes preguntas:

- Cual es el propósito del motor de recomendaciones?
- Cual es el tipo de recomendaciones que se van a proporcionar?
- Las recomendaciones estarán integradas con otras características de la aplicación?
- Cuales métricas se usaran para medir el rendimiento del motor de recomendaciones?

4.1.1. Propósito del motor de recomendaciones:

El recomendador puede ser un servicio proporcionado por la aplicación que lo contiene (application hosting), pero puede tener menor importancia que otros servicios proporcionados por la aplicación. Además, un recomendador dentro del contexto de negocios puede tener fácilmente acceso a ítems difíciles de encontrar como es el caso de muchos sitios de e-commerce, retornando los servicios que mejor se ajustan a sus necesidades. Finalmente, el propósito de un sistema de recomendaciones es incrementar la eficiencia del sistema permitiendo al usuario obtener directamente el contenido que está buscando.

4.1.2. Tipos de recomendaciones:

Un recomendador puede proporcionar varias clases de recomendaciones, desde un simple ítem (o una simple lista de ítems) o una secuencia (por ejemplo un sistema de recomendaciones de viajes).

4.1. Diseño del Modelo de Aplicación

Los recomendadores de un simple ítem o una lista simple de ítems no tienen en cuenta cómo la escogencia de un ítem por el usuario en un punto de tiempo puede influenciar en la escogencia de los siguientes ítems. La escogencia del tipo de recomendaciones debe ser dirigida por la necesidad o presencia de un orden lógico en las recomendaciones, por ejemplo, en un sistema de recomendaciones de viajes, un viaje puede ser visto como una secuencia de puntos de viaje (como visitar un museo, luego una playa, etc.), los cuales pueden ser conectados a través de varias características lógicas como la geografía, cultura, ocio, entre otros, en orden de proporcionar una buena experiencia de viaje. Las recomendaciones de secuencia de ítems pueden ser particularmente útiles cuando los usuarios son nuevos en un dominio y necesitan una trayectoria en la selección de diversos ítems, ayudándolos en el desarrollo de sus objetivos personales: el orden lógico de las recomendaciones ayuda progresivamente en su curva de aprendizaje proporcionando los siguientes ítems más apropiados.

4.1.3. Integración con el contenido de navegación:

Cómo las recomendaciones se integraran con otras características de contenido de navegación? En la mayoría de los casos, se ofrecerá a los usuarios otro medio para buscar contenido por medio de recomendaciones. Una buena integración de estos diferentes métodos de navegación pueden en gran medida mejorar la experiencia de usuario. Los usuarios pueden solicitar recomendaciones completamente separadas de la búsqueda de contenido. Esto puede ser una buena elección si las recomendaciones están destacadas como una característica principal de la aplicación. Dichas recomendaciones pueden aparecer en la página principal de un sitio web o en la pantalla principal de una aplicación. Esto también puede ser beneficioso para tener recomendaciones dependientes del actual contexto de interacción. El caso típico es recomendar ítems que son similares a los que el usuario está actualmente buscando. En este caso, el sistema de recomendaciones estará habilitado para proporcionar recomendaciones adaptadas al contexto, por ejemplo, la actual categoría cuando esta buscando un artículo de noticias (si la actual categoría es deportes entonces el recomendador recomendará ítems en esta categoría).

En este sentido, es importante considerar si el sistema de recomendaciones es una parte opcional u obligatoria del modelo de interacción. Esto tiene fuertes implicaciones en la confiabilidad esperada del sistema: las fallas para completar una mayor tarea en un sitio web porque la única manera de completar dicha tarea es usando un sistema de recomendaciones, el cual ofrece recomendaciones inexactas puede ser una fuente de una mayor insatisfacción del usuario. Sin embargo dentro del diseño del sistema donde el recomendador actúa en paralelo a métodos de navegación tradicionales, el impacto del mismo recomendador puede ser muchas veces menos severo.

4.1.4. Métricas de rendimiento:

La escogencia de las métricas no solamente permite evaluar el recomendador una vez que se construye sino también la influencia de la escogencia de los algoritmos a implementar. Las principales métricas que se pueden considerar son las siguientes:

- Métricas de Exactitud como Accuracy, Precision y Recall: son criterios técnicos los cuales pueden ser usados para evaluar el sistema de recomendaciones. Accuracy es el margen de error del recomendador mientras realiza la recomendación de los ítems. Precision se refiere a la proporción del top de recomendaciones que son buenas recomendaciones. Recall representa la proporción de las buenas recomendaciones que aparecen en el top de recomendaciones. En la sección 6.1.2 del capítulo de evaluación se describirán detalladamente.

- Clarity y explainability: ésta muestra que tan importante es que el usuario entienda cómo las recomendaciones han sido determinadas. Un buen nivel de transparencia puede ser mas difícil de lograr con algunas familias de algoritmos.
- Serendipia: los usuarios pueden ser gratamente sorprendidos por algunas de las recomendaciones o es conveniente permitir recomendaciones obvias?.
- Toma de riesgos: relacionado con el anterior criterio, se realizarán recomendaciones solamente para ítems que el usuario tiene una alta probabilidad de tener preferencia?. Ítems mas riesgosos pueden ser recomendados si el objetivo es permitir al usuario descubrir contenido que no sería consciente sin la ayuda del sistema. Pero el diseñador de un recomendador debe tener cuidado de los riesgos que asume en la exploración de nuevos ítems. El costo de la insatisfacción de los usuarios puede ser tenido en cuenta antes de tomar dichos riesgos.
- Velocidad de respuesta/rendimiento: en muchos casos, la reactividad de la aplicación es una gran preocupación y puede ser algunas veces más importante que la exactitud de los resultados. Conociendo cómo muchas recomendaciones son necesarias por unidad de tiempo permite escoger los mejores algoritmos o decidir si las recomendaciones se realizan pre-calculadas.
- Fiabilidad: Mide que tan crítica es la salida de un recomendador en el contexto de la aplicación?. Por ejemplo el diseño de un recomendador para un sitio web de e-commerce puede no ser acerado de la misma manera como una solución para un sistema de donantes compatibles de órganos en un hospital.

Adicionalmente a las características vistas por los usuarios, algunos aspectos de la implementación también tienen una gran influencia en el diseño del recomendador.

4.1.5. Únicos o múltiples dispositivos:

La misma aplicación puede ser accedida desde un único o múltiples dispositivo(s) (por ejemplo un sistema de recomendaciones de noticias en un móvil, PC). Esto debería ser estudiado si las recomendaciones dependen del contexto de usuario. Pero en términos de implementación, también se plantean preguntas adicionales:

Las preferencias coleccionadas son fusionadas o deberían seguir siendo separadas para habilitar recomendaciones contextuales?. Donde deberían ser almacenadas las preferencias?. Si las preferencias son almacenadas en un servidor, éstas son transmitidas al servidor en tiempo real (implicando una conexión constante) o en lotes?. Respondiendo estas preguntas es importante incluso si el acceso desde múltiples dispositivos no está inicialmente planeado, ya que se está convirtiendo en la norma que aplicaciones web son derivadas de versiones móviles.

4.1.6. Únicos o múltiples usuarios:

Inversamente, el mismo dispositivo puede ser usado por varios usuarios. Adicionalmente, los usuarios que interactúan con la aplicación anfitriona pueden estar registrados o ser anónimos y pueden interactuar frecuentemente u ocasionalmente. Ésto impacta la arquitectura de un recomendador (requiere diferentes mecanismos de identificación, por ejemplo login vs. cookies), escogencia del algoritmo (por ejemplo perfil de sesión en caso de usuarios ocasionales anónimos vs. perfil persistente en caso de usuarios registrados), y parámetros de algoritmos (ejemplo, el grado de adaptabilidad del sistema

4.2. Inclusión de contexto en Sistemas de Recomendaciones

al usuario - ejemplo la rapidez de la adaptación del perfil: a largo plazo frente a corto plazo - que suele depender de la frecuencia del uso).

4.1.7. Infraestructura de la aplicación:

La infraestructura de la aplicación contenedora impone restricciones fuertes de los tipos de algoritmos de recomendación que pueden ser usados en su implementación específica. En particular, la escalabilidad de la solución tiene que ser cuidadosamente estudiada. Dos casos principales pueden ser identificados, si la aplicación es accedida a través de un navegador o si una aplicación se ejecuta localmente en el dispositivo del usuario.

- Aplicación basada en navegador. En el caso de una aplicación basada en navegador, el procesamiento que se realiza en el cliente es mínimo. Como toda la información está disponible en un simple punto, cualquier tipo de algoritmo puede ser usado. Sin embargo, la escalabilidad se enfocará en el diseño.
- Aplicación distribuida. Cuando el dispositivo del usuario ejecuta una aplicación local, diferentes arquitecturas pueden ser usadas. Para determinar la arquitectura distribuida más adecuada, los siguientes criterios pueden ser estudiados:
 - Poder de procesamiento de dispositivos relevantes. Es el dispositivo del cliente capaz de soportar tareas interactivas?. Puede el lado del servidor computar recursos que son extendidos según sea necesario para el número de usuarios?.
 - Conectividad de red. Es la conexión de red permanente?. Tiene la transferencia de datos un costo para los usuarios?. Para dispositivos móviles, cual es el impacto de la conexión en la vida de la batería?.
 - Fuente de datos. Cómo son los datos de los que proceden las recomendaciones? Desde una base de datos (recuperación de información) o un stream de datos (filtrado de información)?. Estado real de la pantalla: un punto que es fácilmente pasado por alto en etapas tempranas de diseño es la cantidad de espacio en pantalla que las recomendaciones usarán. En muchos casos esto es muy limitado por el diseño de la interfaz de usuario de la aplicación. Esto no es solo una cuestión cuantitativa y puede influenciar en la naturaleza de las recomendaciones que son proporcionadas. Por ejemplo, si está destinada para proporcionar recomendaciones más exploratorias, es necesario tener un suficiente número de recomendaciones y así suficiente espacio. El estudio de la aplicación de la cual hace parte el sistema de recomendaciones impone un primer conjunto de restricciones en el diseño del recomendador, pero no es suficiente para construir un sistema apropiado. Esto requiere adicionalmente conocimiento acerca de ambos: el usuario y los datos.

4.2. Inclusión de contexto en Sistemas de Recomendaciones

En la exploración de los paradigmas para incluir información de contexto en los sistemas de recomendaciones, se encontró que en [20] se argumenta que la información contextual relevante es importante en los sistemas de recomendación y que es necesario tomar en cuenta esta información al proporcionar recomendaciones, y por ende se discute la noción general de contexto y la forma en que se puede modelar en sistemas de recomendación. A continuación se explica dicha metodología,

la cual será aplicada en la presente propuesta en la toma de decisiones con respecto a qué paradigma emplear.

Los sistemas de recomendación surgieron como un área de investigación independiente a mediados de la década de 1990, cuando los investigadores y practicantes comenzaron centrándose en problemas de recomendación que se basan explícitamente en la noción de calificaciones (ratings) como una manera de capturar las preferencias del usuario para los diferentes ítems. Por ejemplo, en el caso de un sistema de recomendación de películas, John Doe podrá asignar una calificación de 7 (sobre 10) para la película “Gladiator”, es decir, establecer $R_{movie}(\text{John Doe}, \text{Gladiator}) = 7$. El proceso de recomendación general comienza con la especificación del conjunto inicial de calificaciones que es proporcionado explícitamente por los usuarios o se infiere de forma implícita por el sistema. Una vez que estas calificaciones iniciales se especifican, un sistema de recomendación intenta estimar la función de clasificación R

$$R : User \times Item \rightarrow Rating$$

para los pares (user, ítem) que no han sido calificados por los usuarios. Aquí Rating es un conjunto totalmente ordenado (por ejemplo, números enteros no negativos o números reales dentro de un cierto rango), y User e Item son los dominios de usuarios e ítems respectivamente. Una vez que la función R se calcula para todo el espacio $User \times Item$, un sistema de recomendación puede recomendar el ítem mejor calificado (o k ítems de mejores calificados) para cada usuario. Este tipo de sistemas son llamados sistemas tradicionales o de dos dimensiones (2D), ya que consideran sólo las dimensiones de Usuarios e Ítems en el proceso de recomendación.

En otras palabras, en su formulación más común, el problema de recomendación se reduce al problema de estimar ratings para los ítems que no han sido vistos por un usuario. Esta estimación se basa generalmente en las calificaciones otorgadas por este usuario a otros ítems, las calificaciones dadas a este ítem por otros usuarios, y posiblemente en alguna otra información, así (por ejemplo, la demografía de los usuarios, las características de ítems). Mientras que una cantidad sustancial de investigaciones se han realizado en el área de sistemas de recomendaciones, la gran mayoría de los enfoques existentes se centran en recomendar ítems a usuarios o usuarios a ítems y no toman en consideración cualquier información contextual adicional, tales como el tiempo, el lugar, la compañía de otras personas. Es por esto que en esta sección se toman como base los sistemas de recomendaciones sensibles al contexto (CARS), que tratan de modelar y predecir los gustos y preferencias del usuario mediante la incorporación de la información contextual disponible en el proceso de recomendación como categorías adicionales explícitas de datos. Estas preferencias y gustos a largo plazo por lo general se expresan como calificaciones y se modelan como la función no sólo de ítems y usuarios, sino también del contexto. En otras palabras, las calificaciones se definen con la función de clasificación como

$$R : User \times Item \times Context \rightarrow Rating$$

donde User e Item son los dominios de usuarios e ítems respectivamente, Rating es el dominio de las calificaciones, y Context especifica la información contextual asociada a la aplicación.

4.2.1. Paradigmas para incluir contexto en sistemas de recomendaciones

El uso de la información contextual en sistemas de recomendación se puede remontar al trabajo de Herlocker y Konstan [172], cuya hipótesis de que la inclusión de conocimientos sobre la tarea del usuario en el algoritmo de recomendación en ciertas aplicaciones puede conducir a mejores recomen-

4.2. Inclusión de contexto en Sistemas de Recomendaciones

daciones. Diferentes enfoques para el uso de información contextual en el proceso de recomendación pueden ser clasificados en dos grupos:

Recommendation via context-driven querying and search

Este enfoque ha sido utilizado por una amplia variedad de sistemas de recomendaciones móviles y turísticos [173] [174] [124]. Los sistemas que utilizan este enfoque suelen utilizar información contextual (obtenida ya sea directamente del usuario, por ejemplo, mediante la especificación del actual estado de ánimo o interés, o del medio ambiente, por ejemplo, la obtención de la hora local, el clima, o la ubicación actual) para consultar o buscar en un determinado repositorio de recursos (por ejemplo, restaurantes) y presentar los mejores recursos que coinciden (por ejemplo, los restaurantes cercanos que están actualmente abiertos) para el usuario.

Contextual preference elicitation and estimation

El otro enfoque representa una tendencia más reciente en la literatura de sistemas de recomendación sensibles al contexto [142] [175] [176] [177]. Las técnicas que siguen a este segundo enfoque intentan modelar y aprender preferencias del usuario, por ejemplo, mediante la observación de las interacciones de éste y otros usuarios con los sistemas o por la obtención de retroalimentación de preferencias del usuario en diversos ítems recomendados previamente. Para modelar las preferencias sensibles al contexto de los usuarios y generar recomendaciones, estas técnicas típicamente adoptan los métodos de filtrado colaborativo existente, basado en contenido, o recomendación híbrida con los ajustes de recomendación context-aware o aplican diversas técnicas de análisis de datos inteligentes de minería de datos o aprendizaje de máquina de (tales como clasificadores bayesianos o máquinas de vectores soporte). Esta propuesta está basada en el segundo enfoque ya que es la tendencia más reciente y el presente sistema está basado en un sistema de recomendación híbrido. Más específicamente, el proceso de recomendación basado en Contextual preference elicitation and estimation puede adoptar una de las tres siguientes formas, basadas en la utilización del contexto, como se muestra en la Figura 4.2:

- **Contextual Prefiltering** (o contextualización de entrada de recomendación): En este paradigma de recomendación (presentada en la figura 4.2a), la información contextual impulsa la selección de datos o la construcción de datos para ese contexto específico. En otras palabras, la información sobre el contexto actual c se utiliza para seleccionar o construir el conjunto pertinente de los registros de datos (por ejemplo, ratings). Entonces, las calificaciones se pueden predecir utilizando cualquier sistema de recomendación 2D tradicional sobre los datos seleccionados.
- **Contextual Postfiltering** (o contextualización de salida de recomendación): En este paradigma de recomendación (presentada en la figura 4.2b), la información contextual se ignora inicialmente, y las calificaciones son predichas con el uso de cualquier sistema de recomendación tradicional 2D en los datos completos. A continuación, se ajusta el conjunto resultante de recomendaciones (contextualizada) para cada usuario utilizando la información contextual.
- **Contextual Modeling** (o contextualización de la función de recomendación): En este paradigma de recomendación (presentada en la figura 4.2c), la información contextual se utiliza directamente en la técnica de modelado como parte de la estimación de calificación.

En resumen se tiene que:

- Contextual Prefiltering: Información contextual filtra un conjunto de datos antes de aplicar un algoritmo de recomendación tradicional.
- Contextual Postfiltering: Recomendaciones sobre todo el conjunto de datos y se ajustan usando información contextual.
- Contextual Modeling: Uso de información contextual directamente en la función de recomendación como predictor explícito de un rating para un ítem.

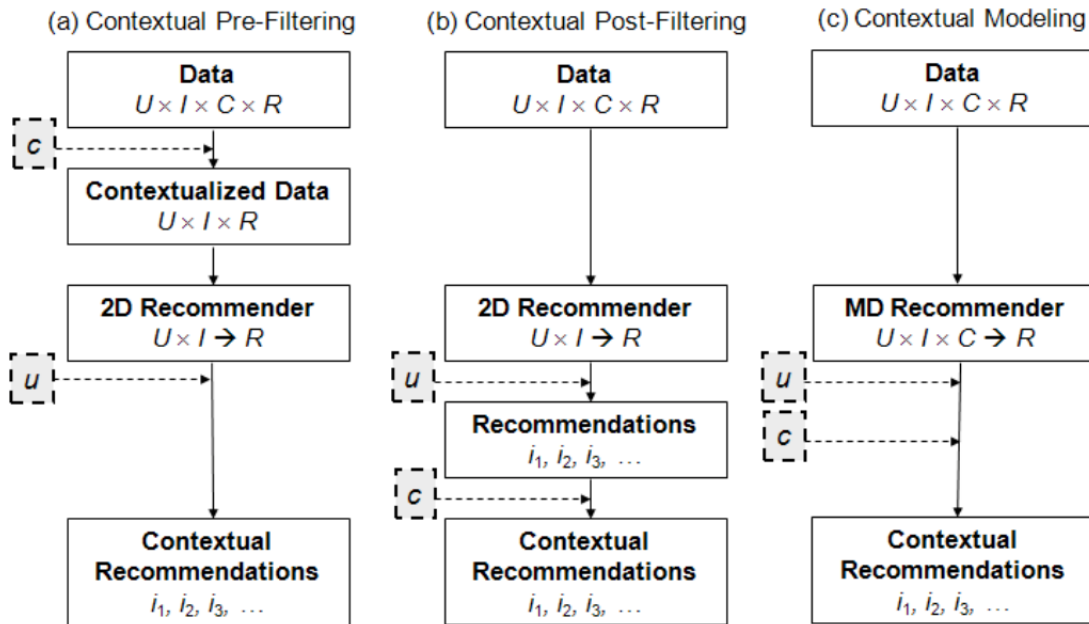


Figura 4.2: Paradigmas para incluir contexto en Sistemas de Recomendaciones.

4.3. Métodos de Hibridación

De acuerdo con lo discutido anteriormente, el enfoque propuesto está basado en “Contextual preference elicitation and estimation”, adoptando la forma de utilización de contexto “Contextual Modeling”. En este sentido, es importante describir las técnicas de hibridación que combinan dos o más técnicas de recomendación para obtener un mejor rendimiento. Más comúnmente, el filtrado colaborativo se combina con alguna otra técnica, a continuación se presentan algunos de los métodos de combinación que se pueden emplear [48]:

4.3.1. Weighted

En un weighted recomendador híbrido la puntuación de un ítem recomendado se calcula a partir de los resultados de todas las técnicas de recomendaciones disponibles presentes en el sistema. Por

4.3. Métodos de Hibridación

ejemplo, el híbrido simple combinado sería una combinación lineal de las puntuaciones de recomendación. La ventaja este método es que todas las capacidades del sistema se ejercen en el proceso de recomendación de una manera sencilla y es fácil de realizar la asignación de pesos. Sin embargo, la suposición implícita en esta técnica es que el valor relativo de las diferentes técnicas es más o menos uniforme en todo el espacio de posibles ítems.

4.3.2. Switching

Este método se basa en la sensibilidad a nivel de ítem en la estrategia de hibridación: el sistema utiliza algún criterio para cambiar entre las técnicas de recomendación. Por ejemplo, un sistema usa un híbrido de contenido/colaboración en el que un método de recomendación basado en contenido se emplea por primera vez, y si éste no puede hacer una recomendación con la suficiente confianza, entonces una recomendación colaborativa es realizada. Esto hace que se introduzca una complejidad adicional en el proceso de recomendación ya que los criterios de conmutación deben determinarse, y esto introduce otro nivel de parametrización.

4.3.3. Mixto

Donde es práctico hacer gran número de recomendaciones simultáneamente, puede ser posible utilizar un híbrido “mixto”, donde se presentan recomendaciones de más de una técnica, en ese sentido, las recomendaciones de las dos técnicas son combinadas en el resultado sugerido al final. Por lo general, la recomendación requiere un ranking de ítems o la selección de una única mejor recomendación, y en ese momento debe emplearse algún tipo de técnica de combinación.

4.3.4. Combinación de características

Otra forma de conseguir la fusión de un contenido/colaborativo es tratar la información de colaboración como datos de entidad simplemente adicionales asociados con cada ejemplo y utilizar técnicas basadas en el contenido sobre este conjunto de datos aumentado. El híbrido de combinación de características permite que el sistema considere los datos de colaboración sin depender exclusivamente de él, por lo que reduce la dependencia del sistema del número de usuarios que han valorado un ítem.

4.3.5. Cascada

A diferencia de los métodos de hibridación anteriores, el híbrido cascada implica un proceso por etapas. En este método, una técnica de recomendación se emplea primero para producir un ranking de los candidatos y una segunda técnica refina la recomendación del conjunto de candidatos. Cascada permite que el sistema evite el empleo de la segunda técnica de menor prioridad, la técnica de los ítems que ya han sido bien diferenciados por la primera o que están lo suficientemente mal valorados y que nunca van a ser recomendados. El segundo paso de la cascada se centra sólo en aquellos ítems para los que hace falta una discriminación adicional, es más eficiente que, por ejemplo, un weighted híbrido que se aplica para todos los ítems.

4.3.6. Característica de Aumento

Esta técnica se emplea para producir una calificación o clasificación de un ítem y la información se incorpora entonces en el procesamiento de la siguiente técnica de recomendación. Mientras que

ambas técnicas tanto de cascada y de aumento que secuencian dos recomendadores, con el primer recomendador tiene una influencia sobre el segundo, son fundamentalmente muy diferentes. En un híbrido de aumento, las características utilizadas por el segundo recomendador incluyen la salida del primero. En un híbrido en cascada, el segundo recomendador no utiliza ninguna salida del primero en la producción de sus clasificaciones, pero los resultados de los dos recomendadores se combinan de una manera priorizada.

4.3.7. Meta nivel

Otra forma en que dos técnicas de recomendación pueden ser combinadas es utilizando el modelo generado por uno como entrada para otro. Esto difiere de la característica de aumento: en un híbrido de aumento, utilizamos un modelo aprendido para generar características para la entrada a un segundo algoritmo; en un híbrido meta-nivel, todo el modelo se convierte en la entrada. Un híbrido meta-nivel que se centra exclusivamente en la recomendación, es descrito por Pazzani [45] como “la colaboración a través de contenidos”. Estos modelos, esencialmente vectores de términos y pesos, a continuación, se pueden comparar entre los usuarios para hacer predicciones.

El beneficio del método de meta-nivel, especialmente para el híbrido contenido/colaborativo es que el modelo aprendido es una representación comprimida de los intereses de un usuario, y un mecanismo de colaboración que sigue, puede operar en esta representación de información densa más fácilmente que en los datos de calificación en bruto.

A continuación se resumen los métodos descritos:

| Método de Hibridación | Descripción |
|-----------------------|---|
| Weighted | Las puntuaciones (o votos) de varias técnicas de recomendaciones son combinadas para producir una recomendación simple. |
| Switching | El sistema switchea entre técnicas de recomendaciones dependiendo de la actual situación. |
| Mixed | Recomendaciones de diferentes recomendadores son presentadas al mismo tiempo. |
| Feature combination | Características de diferentes fuentes de datos de recomendación están incluidas dentro de un algoritmo de recomendación simple. |
| Cascade | Un recomendador refina las recomendaciones realizadas por otro. |
| Feature augmentation | La salida de una técnica es usada como la entrada de características de otra. |
| Meta-level | El modelo aprendido por el recomendador es usado como entrada de otro. |

Tabla 4.1: Métodos de Hibridación

4.4. Modelo de Aplicación

El enfoque a usar en el motor de recomendación es un enfoque híbrido que combina a base de filtrado de conocimiento y filtrado colaborativo. La razón de la escogencia de dicho enfoque a parte de las limitaciones de cada uno por separado encontrado en la literatura, es por las características

4.5. Fases del enfoque propuesto

intrínsecas propias del proyecto, para lo cual es necesario emplear el perfil de usuario, las características contextuales, y las características de los ítems a recomendar.

4.4.1. Propósito del motor de recomendación

El objetivo principal del sistema es incentivar el consumo de oferta turística por medio de la habilitación y maximización de posibles movimientos a realizar por parte de un turista optimizando su tiempo en un destino.

4.4.2. Tipo de recomendaciones

El tipo de recomendaciones que el motor de recomendación va a generar son actividades a realizar en sitios turísticos (puntos de interés).

4.4.3. Uno o varios dispositivos

La versión actual del motor de recomendación es para dispositivos móviles.

4.4.4. Usuarios individuales o múltiples

El motor de recomendación generará recomendaciones para múltiples usuarios, cada uno con una aplicación local en su dispositivo móvil.

4.4.5. Métrica de rendimiento

Precision, recall y utilidad.

4.4.6. Infraestructura de aplicación

La aplicación se basará en el sistema operativo Android con comunicación HTTP hacia el servidor.

4.5. Fases del enfoque propuesto

El enfoque a usar para el motor de recomendaciones es un enfoque híbrido el cual combina un filtro basado en conocimiento y uno colaborativo. Los problemas de la sobre-especialización y el nuevo usuario también se superan mediante las valoraciones de los usuarios. Los problemas de cold start y nuevo rater en filtrado colaborativo que son el resultado de la escasez de datos (escasez en ratings de usuario), haciendo difícil para el filtro colaborativo proporcionar recomendaciones exactas, son superadas usando las características de conocimiento el cual introduce datos adicionales suficientes para usar el recomendador híbrido. En orden de decidir cual arquitectura y cuál modelo de usuario usar, en el diseño del sistema se debe tener en cuenta el ambiente del sistema de recomendaciones. Este ambiente puede ser estudiado sobre los 3 modelos anteriormente descritos: modelo de aplicación, modelo de usuario y modelo de datos. Dado que el filtrado colaborativo es una técnica bien establecida para las preferencias de las predicciones, y está muy bien soportado por Apache Mahout, se utilizará esta técnica. Sin embargo, en un sistema de recomendación basado en la colaboración, ítems co-calificados por un par de usuarios pueden ser muy pocos y la correlación entre dos usuarios puede ser insuficiente, por tanto la información basada en conocimiento de cada usuario se explota para

soportar el conocimiento del turista (contexto del usuario), conocimiento de los ítems y el dominio en consideración para soportar la definición y relaciones semánticas y facilidades de razonamiento. Se emplea el método de hibridación: “Característica de aumento”, en el que la salida de una técnica se utiliza como una característica de entrada a otra.

4.5.1. Inferencia de Perfiles Contextuales

En esta sección se explicará la inferencia del perfil del usuario a partir de la utilización de 3 variables contextuales del Modelo del Usuario, las cuales son capturadas implícitamente a través de un formulario de la aplicación móvil, y formateadas de manera que puedan ser entendidas computacionalmente. En primer lugar se transforman los datos en vectores contextuales que almacenan varias variables que pueden ser interpretadas mejor para producir un vector de contexto con valores claros y bien definidos. Para modelar las variables de contexto que afectan la identificación del perfil de un turista, se muestra el vector resultante en la siguiente tabla:

| | | |
|-----------|--------------|------------|
| Tipología | Motivo Viaje | Rango Edad |
|-----------|--------------|------------|

Tabla 4.2: Vector de perfil de contexto

Para calcular el valor de las variables, se sincronizan respecto al tiempo en porciones de tiempo discretas, creando un vector de perfil contextual $V(t)$ el cual es parametrizado por el tiempo t . La duración de la porción de tiempo determina la cantidad de datos disponibles del contexto para el paso de inferencia o razonamiento del perfil contextual. Los estados de contenido del vector pueden ser calculados en la siguiente ecuación:

$$v(t) = \{CS_1(t), CS_2(t)..CS_n(t)\} \quad (1)$$

Donde CS es la variable contextual, $CS_i(t)$ es el valor de i de la variable contextual para la porción de tiempo t . Ciertas variables no pueden ocurrir simultáneamente y luego tiene valores binarios, que puede tomar valores de 1 o 0. Por ejemplo, el rango de edad puede ser Menor a 18 o 18 a 30.

Razonamiento de perfil contextual

Las variables contextuales se convierten en perfiles contextuales cuando son combinadas: por ejemplo, turistas que llegan acompañados por familiares buscan planes de tipo familiar, mientras que los jóvenes interesados en entretenimiento o diversión prefieren actividades nocturnas. En este sentido, se usan algoritmos de clasificación para inferir perfiles contextuales, por ejemplo la clase a la cual se ajusta la combinación de las variables de perfil de contexto descritas anteriormente. Las combinaciones de dichas variables y usar los clasificadores para los cuales los resultados de predicción son mejores, indicando en ese modo que tan relevantes son las clases. Se ha etiquetado el conjunto de datos basado en los perfiles contextuales objetivo y creado el vector de perfil de contexto, para tenerlos en cuenta como entrada para inferir los diferentes perfiles contextuales: este proceso puede ser realizado con técnicas de Aprendizaje Automático. La figura 4.3 muestra el diagrama conceptual de este proceso:

Se deduce el perfil contextual de los datos disponibles, los cuales son clasificados en patrones que pueden ser interpretados como perfiles turísticos. En éste análisis, se considera el perfil contextual proporcionado como un recurso, la máquina interpreta el conjunto de datos del contexto del usuario

4.5. Fases del enfoque propuesto

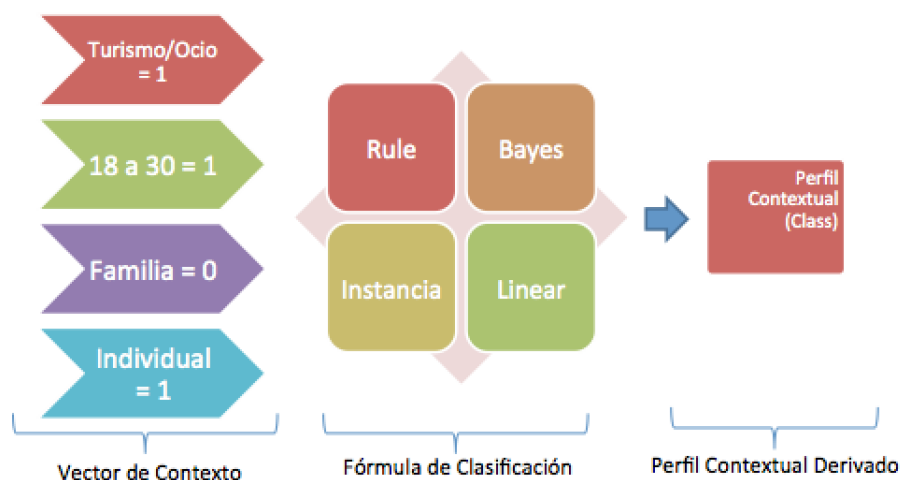


Figura 4.3: Identificación de Perfil Contextual.

como una manera que se usa para entender el comportamiento del usuario [178]. Cuando se clasifican perfiles contextuales, se continúa usando el modelo clasificado que fue creado después de los datos entrenados con los datos de usuario, y se usa para predecir el perfil contextual de un nuevo turista a partir de dicho modelo aprendido.

Definición de Perfiles Contextuales

Para definir perfiles contextuales a usar, se analizan los escenarios que pueden suceder cuando un turista llega a un destino. Se definen los perfiles a derivar, para realizar esto, se analizan las 3 variables definidas previamente relacionándolas a varios perfiles contextuales que son específicas para el entorno turístico como se detalla a continuación:

| |
|---|
| Escenarios |
| Turista llega a un destino acompañado de su familia |
| Turista llega a un destino en Grupo |
| Turista llega a un destino de manera individual |
| El motivo de viaje de un turista es Turismo/Ocio |
| El motivo de viaje de un turista es Eventos |
| El motivo de viaje de un turista es Visita a Familiares |
| El motivo de viaje de un turista es Trabajo/Negocios |
| El motivo de viaje de un turista es Otros |
| El turista tiene un rango de edad Menor a 18 años |
| El turista tiene un rango de edad entre 18 y 30 años |
| El turista tiene un rango de edad entre 30 y 50 años |
| El turista tiene un rango de edad mayor a 50 años |

Tabla 4.3: Ejemplo de posibles escenarios para un Turista

Estas variables son combinadas para crear un posible escenario turístico basado en lo valores de

Capítulo 4. Enfoque del Sistema de Recomendaciones Propuesto

dichas variables, lo cual es llamado “context stamp”[179]. De acuerdo a esto se han definido cuatro principales perfiles contextuales: Classic Tourist, Clubbing-type, Family Type y Thinker, los cuales fueron tomados de la ontología de Usuario del proyecto Travel Guide Ontology [171] que contiene conceptos relacionados con los viajeros. Esta ontología describe intereses de los usuarios y actividades, grupos de edad, las compañías de viajes preferidas, y otros datos sobre los diferentes tipos de usuarios y perfiles de usuario, para lo cual ampliaron el módulo de gestión del conocimiento PROTON con los conceptos y las relaciones de dominio-relacionados para la ontología del usuario. Cada perfil es un escenario turístico para el usuario, dependiendo de las dimensiones del modelo de usuario que se emplean para caracterizar a un turista, como se muestra en la tabla 4.4:

| Perfil Contextual | Definición |
|-------------------|--|
| Classic Tourist | Turista interesado en conocer mejor otras culturas, viajando a un destino turístico con el fin de ampliar sus horizontes, adquirir conocimientos de otras culturas, y aprender una lengua extranjera. [180] |
| Clubbing-type | Turista que viaja principalmente para visitar clubes nocturnos u otros bares y lugares de ocio. [181] |
| Family Type | Turistas que les interesan actividades orientadas a la familia como: visitas a lugares históricos, experiencias relacionadas con el agua (lago/mar), la montaña/aventuras al aire libre, las experiencias de los museos urbanos, reuniones familiares y compras. [182] |
| Thinker | Turista interesado en visitar sitios relacionados con arte, museos, instituciones educativas, eventos, trabajo o negocios, entre otros. |

Tabla 4.4: Definición de Perfiles Contextuales

Se aplica el modelo de usuario usando la formula de vector de perfil contextual $V(t)$ (Ecuación 1). Se puede ver una imagen del entorno de trabajo unificado de variables de perfil contextual en la tabla 4.5, la cual es definida basada en todas los posibles valores que pueden tomar las variables y en el previo modelo de usuario definido.

| Variable de Tipología | | | Variable Rango de Edad | | | | Variable Motivo de Viaje | | | | |
|-----------------------|-----------------|------------|------------------------|---------------|---------------|--------------------|--------------------------|----------------------------------|--------------------------------|--------------|------------|
| Fami- lia | Indivi- dual | Gru- po | Me- nor a 18 | 18 a 30 | 30 a 50 | Ma- yor a 50 | Turis- mo/ Ocio | Visi- ta a Fami- liares | Tra- bajo/ Nego- cios | Even- tos | O- tros |
| 1 o 0 | 1 o 0 | 1 o 0 | 1 o 0 | 1 o 0 | 1 o 0 | 1 o 0 | 1 o 0 | 1 o 0 | 1 o 0 | 1 o 0 | 1 o 0 |

Tabla 4.5: Vector de Perfil contextual

Reglas de Inferencia

Cuando se infieren perfiles contextuales, se usa un algoritmo clasificador de Aprendizaje Automático para permitir automáticamente derivar un perfil contextual basado en el vector de contexto. Para entrenar estos algoritmos, se han definido etiquetas de clase correspondientes a dicho perfil. Se han definido un conjunto de reglas para etiquetar las clases y entrenar los clasificadores en la inferencia de perfiles contextuales. Para etiquetar las clases, se ha seguido el enfoque tomado en el Proyecto

4.5. Fases del enfoque propuesto

MavHome [78]. Cuando se definen etiquetas de contexto, se han tenido que realizar un conjunto de supuestos relacionados al entorno turístico y al comportamiento del turista cuando llega a un destino, los cuales son listados a continuación:

- Los turistas que llegan a un destino acompañados por sus familiares, amigos, compañeros buscan planes de tipo familiar.
- Los turistas que no vienen en plan familiar y su interés es por trabajo, estudio, eventos, etc. prefieren actividades culturales e intelectuales.
- Los turistas jóvenes cuyo interés sea la diversión o el ocio están más dispuestos a realizar actividades nocturnas.

Con relación a lo anterior, se han definido un conjunto de reglas no exhaustivas para etiquetar los perfiles contextuales basado en las diferentes combinaciones que los valores pueden tomar:

- `If (motivo_viaje[Turismo/Ocio] = 1) AND rango_edad[18 a 30] = 1) AND (Tipologia[Individual] = 1 OR Tipologia[Grupo] = 1) => ClubbingType`
- `If (motivo_viaje[Visita a Familiares] = 1) OR (Tipologia[Familia] = 1) => FamilyType`
- `If (motivo_viaje[Eventos] = 1 OR motivo_viaje[Trabajo/Negocios] = 1) => Thinker`
- `If (motivo_viaje[Turismo/Ocio] = 1) AND rango_edad[18 a 30] = 0) AND (Tipologia[Individual] = 1 OR Tipologia[Grupo] = 1) => ClassicTourist`

A continuación se construye un clasificador basado en técnicas de Aprendizaje Automático para inferir perfiles contextuales. Se definen los vectores basados en cada variable de contexto y sus posibles valores, explicando cómo se etiquetan (soporte de supuestos). Se emplea el conjunto de datos descrito en la sección 3.1.2 para probar los algoritmos con los datos de la vida real y evaluar su desempeño para elegir el más adecuado para esta fase de la propuesta. Se utilizan diferentes clasificadores de los diferentes tipos definidos en la sección 2.1.4 soportados ampliamente por WEKA:

- Bayesian family: Bayes Net, Naïve Bayes, WAODE, HNB y AODE
- Functions Classifiers: Logistic, MultilayerPerceptron, RBFNetwork, SimpleLogistic y SMO
- Lazy Classifiers: IB1, IBk, KStar, LBR y LWL
- Rule Classifiers: ConjunctiveRule, DecisionTable, DTNB, JRip, NNge, OneR, PART, Prism, Ridor y ZeroR.
- Tree Classifiers: BFTree, DecisionStump, FT, Id3, J48, J48ggraft, LADTree, LMT, NBTree, RandomForest, RandomTree, REPTree, SimpleCart y UserClassifier.

Se usa el conjunto de reglas descrito anteriormente para etiquetar los conjuntos de datos en los cuatro perfiles contextuales (clases) definidos: Classic Tourist, Clubbing-type, Family Type y Thinker. Las métricas a tener en cuenta para la evaluación de los algoritmos son:

- Precisión (o confianza) denota la proporción de casos positivos que son predichos correctamente positivos reales, es decir, la proporción de casos que se clasifican correctamente como pertenecientes a una clase entre todos los casos clasificados como pertenecientes a esa clase.
- Recall denota la proporción de casos positivos reales que se predice correctamente positivo, es decir, la proporción de casos clasificados como pertenecientes a una clase entre todas las instancias que pertenecen a esa clase, este valor indica cuánto de la clase fue capturado la hora de clasificar.

Se emplean estas medidas para evaluar el desempeño de los clasificadores para elegir el que tiene el mejor equilibrio entre estas métricas calculando los promedios ponderados de los valores obtenidos para cada clase.

Herramientas de programación

Se ejecutaron los algoritmos en una computadora portátil Dell Vostro 1220 con Windows 7 OS, usando la tercera edición de WEKA software simulador de código abierto, publicado bajo la Licencia Pública General GNU, para ejecutar las clasificaciones [183]. Es un ambiente que permite el acceso a una variedad de técnicas de Aprendizaje Automático, con ambas funcionalidades pre y post-procesamiento, y contiene una “colección de herramientas de visualización y algoritmos para el análisis de datos y el modelado predictivo, junto con interfaces gráficas de usuario”[184]. WEKA proporciona un entorno integrado en el que se puede acceder a una variedad de técnicas de aprendizaje automático a través de una interfaz interactiva y proporciona herramientas de análisis que permiten comprender los resultados de las pruebas más en profundidad y ha sido elegido como uno de los favoritos para las pruebas de minería de datos [70]. WEKA permite cargar los vectores de contexto, con el formato .csv, y elegir los atributos para poner a prueba, así como el número de casos a utilizar cuando se entrena el clasificador. En este sentido se evalúa la Precision y Recall de los clasificadores anteriormente mencionados para elegir el modelo con mejor comportamiento y posteriormente clasificar la otra mitad del conjunto de datos usando las siguientes opciones disponibles por WEKA para evaluar el conjunto de datos:

- Training set: se utiliza el conjunto de entrenamiento (el archivo que se carga en el preproceso) para la prueba.
- Cross validation: la validación cruzada es el método para evaluar los clasificadores realizando una partición del conjunto de datos en k partes.
- Percentage split: los resultados de clasificación serán evaluados en una prueba de conjunto que es una parte de los datos originales. La división por defecto (que se muestra en el área de texto junto a la opción) es del 66 %, lo que significa que el 66 % de los datos va para training y el 34 % para testing.

Se ponen a prueba la precisión de los clasificadores, los gráficos de la Figura 4.4, Figura 4.5, Figura 4.6, Figura 4.7 y la Figura 4.8 informan de la Precision de clasificación de cada algoritmo sobre todo el conjunto de datos:

4.5. Fases del enfoque propuesto

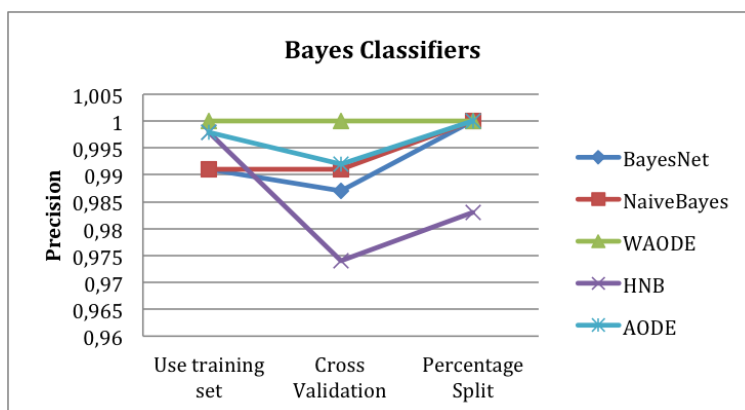


Figura 4.4: Evaluación de Precision para Bayes Classifiers.

Como se puede observar en la anterior figura el mejor clasificador de este tipo (Bayesianos) es WAODE con un resultado promedio del 100 % y para el resto de clasificadores se obtuvo un valor de 99 % para el valor de Precision en el conjunto de datos.

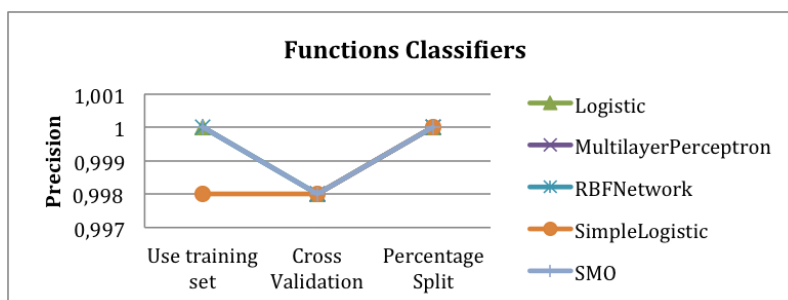


Figura 4.5: Evaluación de Precision para Functions Classifiers.

En la anterior figura ningún clasificador de este tipo (Función) alcanza un resultado promedio del 100 %, ya que se obtiene una tasa de precisión media de 93 % en el conjunto de datos.

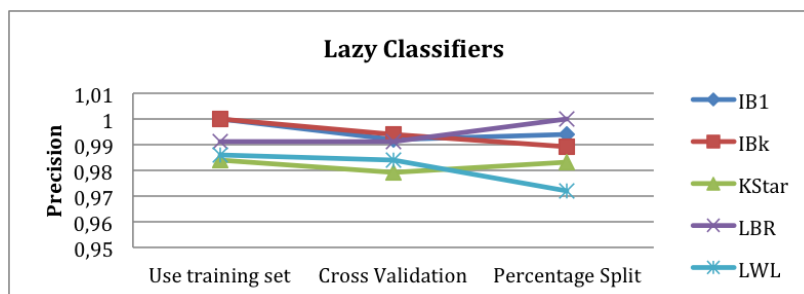


Figura 4.6: Evaluación de Precision para Lazy Classifiers.

Para este tipo (Lazy) ningún clasificador alcanza un resultado promedio del 100 %, ya que se obtiene una tasa de precisión media de 98 % en el conjunto de datos.

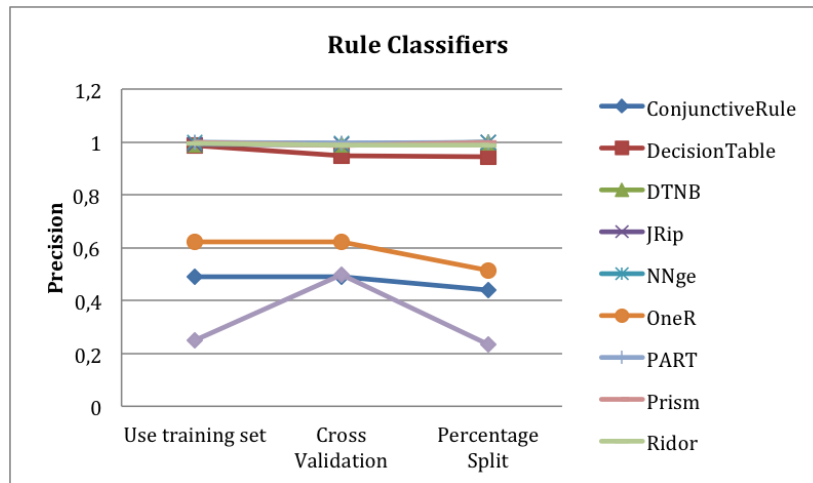


Figura 4.7: Evaluación de Precision para Rule Classifiers.

En la anterior figura ningún clasificador de este tipo (Reglas) alcanza un resultado promedio del 100 %, ya que se obtiene una tasa de precisión media de 83 % en el conjunto de datos.

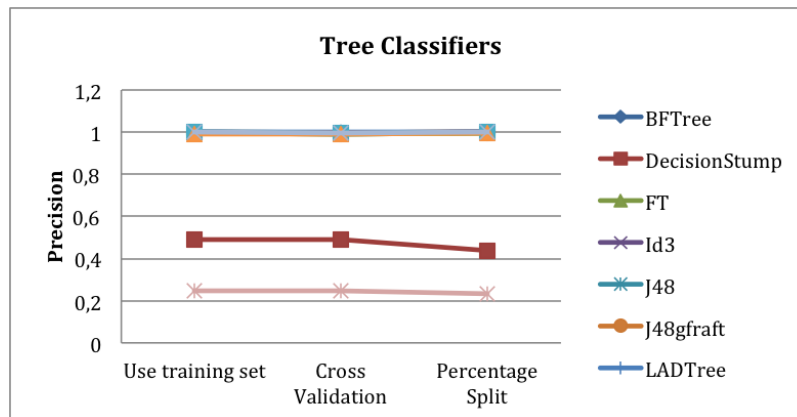


Figura 4.8: Evaluación de Precision para Tree Classifiers.

Como se puede observar en la anterior figura el mejor clasificador de este tipo (Árbol) es LADTree con un resultado promedio del 100 % y para el resto de clasificadores se obtuvo un valor de 90 % para el valor de Precision en el conjunto de datos.

Se ponen a prueba el Recall de los clasificadores, los gráficos de la Figura 4.9, Figura 4.10, Figura 4.11, Figura 4.12 y la Figura 4.13 informan del Recall de clasificación de cada algoritmo sobre todo el conjunto de datos.

4.5. Fases del enfoque propuesto

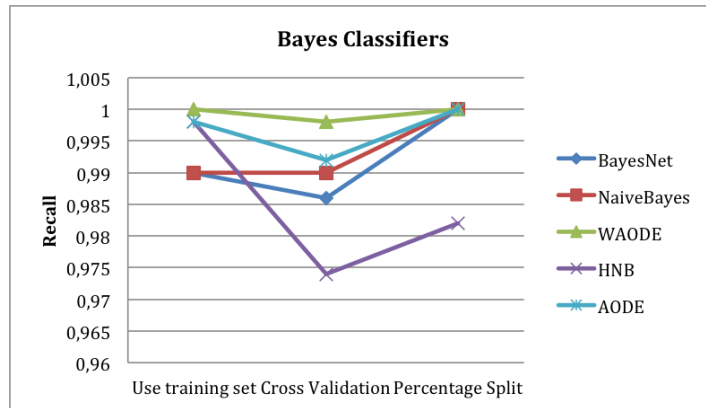


Figura 4.9: Evaluación de Recall para Bayes Classifiers.

Como se puede observar en la anterior figura todos los clasificadores de este tipo (Bayesianos) obtienen un resultado promedio del 99 % para el valor de Recall en el conjunto de datos.

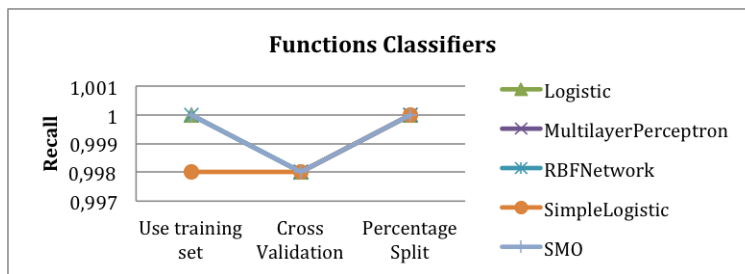


Figura 4.10: Evaluación de Recall para Functions Classifiers.

En la anterior figura ningún clasificador de este tipo (Función) alcanza un resultado promedio del 100 %, ya que se obtiene una tasa de Recall media de 93 % en el conjunto de datos.

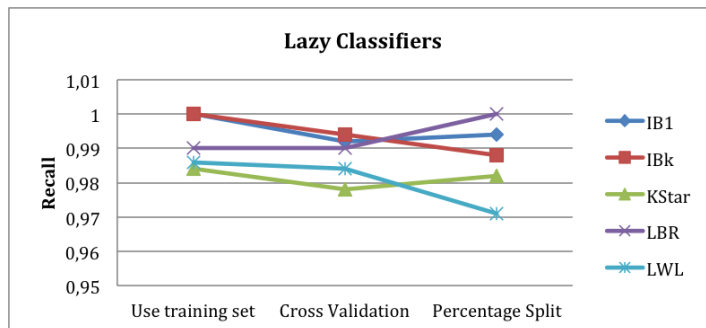


Figura 4.11: Evaluación de Recall para Lazy Classifiers.

Para este tipo (Lazy) ningún clasificador alcanza un resultado promedio del 100 %, ya que se obtiene una tasa de Recall media de 98 % en el conjunto de datos.

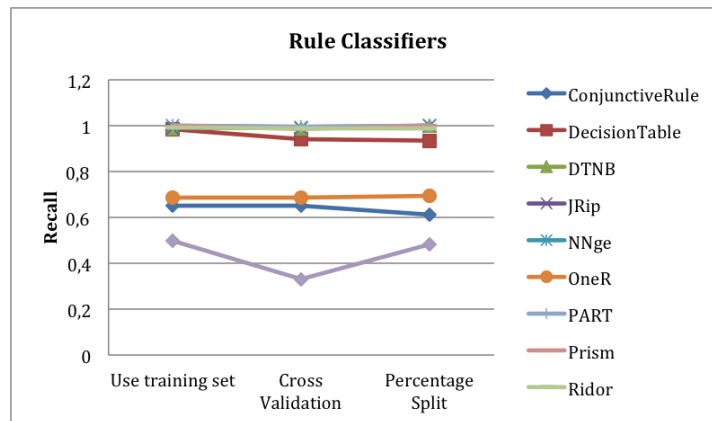


Figura 4.12: Evaluación de Recall para Rule Classifiers.

En la anterior figura ningún clasificador de este tipo (Reglas) alcanza un resultado promedio del 100 %, ya que se obtiene una tasa de Recall media de 86 % en el conjunto de datos.

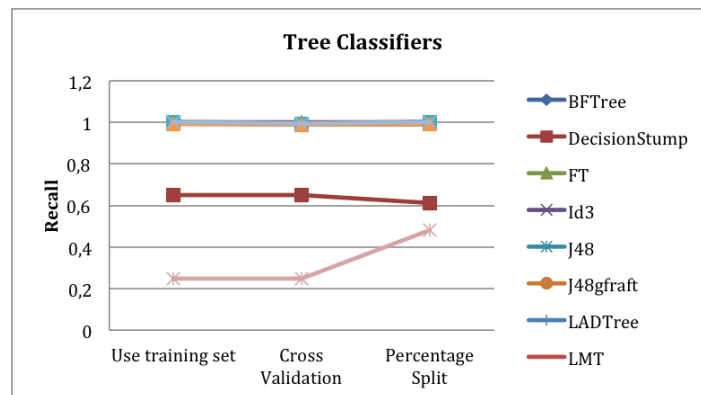


Figura 4.13: Evaluación de Recall para Tree Classifiers.

Como se puede observar en la anterior figura el mejor clasificador de este tipo (Árbol) es LADTree con un resultado promedio del 100 % y para el resto de clasificadores se obtuvo un valor de 92 % para el valor de Recall en el conjunto de datos.

Los resultados de las pruebas muestran que para todos los clasificadores la tasa de precisión media es de 93 % en el conjunto de datos de usuario. Los mejores clasificadores son los WAODE y LADTree con un resultado promedio de 100 %, FT y PART con un resultado promedio de 99.99 %. Gracias a los resultados experimentales realizados se puede proporcionar un punto de control para el modelo de contexto propuesto y la inferencia del perfil contextual con el fin de evaluar el mejor clasificador, los cuales indican que WAODE y LADTree tienen la mejor combinación de rendimiento. Consecuentemente, se emplea el modelo de WAODE previamente cargado por medio del API Weka en java para su implementación como predictor de Perfiles contextuales y obtener el perfil del usuario al que se le van a realizar las recomendaciones.

4.5. Fases del enfoque propuesto

4.5.2. Recomendación basada en Conocimiento

Esta fase es la encargada de realizar el proceso de recomendación empleando un motor de inferencia, guiado por las reglas construidas que se razonan sobre la base de conocimiento (KB) representada por la ontología OWL, la cual recoge las dimensiones del modelo contextual anteriormente planteado: el conocimiento sobre el turista (contexto de usuario), el conocimiento de contenidos y el conocimiento sobre el dominio en consideración. La salida de este proceso es un conjunto de recursos pertinentes, para este caso, esos recursos son actividades turísticas relacionados con un turista.

Representación del conocimiento

Los Sistemas de recomendación basados en contenidos están diseñados principalmente para recomendar objetos basados en texto y emplear técnicas para representar características del objeto que se adquieren directamente de las descripciones textuales de los objetos [49] [185]. En algunos casos, los objetos en los sistemas se describen con vectores de características que se construyen sobre la base de las ocurrencias de las palabras en las descripciones de texto de los objetos. Cada palabra se considera como una característica independiente que caracteriza el objeto [135]. Dicha representación, donde el contenido expresado en lenguaje natural se indexa directamente en el uso de palabras como las características, tiene algunas limitaciones. El problema con la indexación directamente con texto no estructurado es que las realizaciones sintácticas y léxicas de las frases pueden variar, esto se puede mejorar mediante el modelado de los conceptos y sus relaciones en las frases y el uso de las estructuras resultantes como características. Tales características pueden ser representados mediante anotaciones estructuradas basadas en ontologías, que se discute en las siguientes secciones.

Ontologías

En informática, una ontología se puede definir como una especificación de un vocabulario de representación que consiste en las definiciones de clases, relaciones, funciones y otros objetos de un dominio compartido del discurso [186].

Lenguajes de Ontologías y Reglas Los lenguajes desarrollados por la comunidad de la Web Semántica se han adoptado para apoyar la representación conceptual de ontologías y esquemas de anotación. El Marco de Descripción de Recursos (esquema) (RDF (S)) [187] y OWL [188], se puede utilizar para describir formalmente los conceptos y propiedades entre ellos. Las construcciones en estos lenguajes tienen la semántica predefinida. Los Recursos (conceptos y propiedades) se pueden definir y describen el uso de estas construcciones. El lenguaje RDF (S), y la semántica definidas por ellos, se pueden utilizar para describir, por ejemplo, la subsunción y las relaciones de tipo. Los recursos pueden ser identificados con identificadores uniformes de recursos (URI). RDF se ha definido como un modelo general para la descripción de recursos web, utilizando una variedad de formatos de sintaxis. El modelo de datos RDF es similar al enfoque de modelado conceptual como los diagramas entidad-relación [189], ya que se basa en la idea de hacer declaraciones sobre los recursos en forma de declaraciones de sujeto-predicado-objeto, también conocido como triples en la terminología RDF. Una vez que se construye una definición de ontología, sus individuos pueden ser formados como valores de datos reales que se asignan a las propiedades de tipo de datos, y los enlaces a los individuos de otras clases se asignan a propiedades de objetos. Así, se ha demostrado [190] que OWL tiene limitaciones en el caso de representar relaciones entre propiedades complejas. Esto ha sido superado con el trabajo conjunto entre OWL y un lenguaje de reglas. Como resultado de eso, Semantic Web Rule Language

(SWRL) se ha introducido [190] y combina OWL con RuleML (el sub-idioma de Rule Markup Language). En SWRL, un axioma de regla consiste en un antecedente (cuerpo) y un consecuente (cabeza). El elemento básico tanto de antecedente y consecuente es un átomo. SWRL identifica cinco átomos básicos que se han construido en base a conceptos definidos en la ontología. Los átomos son:

- $C(x)$ - se utiliza para comprobar si un determinado individuo x es una instancia del concepto C , por ejemplo, la $Track(Yesterday)$ comprueba si $Yesterday$ es la instancia del concepto $Track$;
- $P(x, y)$ - permite comprobar si dos individuos x y y se relacionan entre sí a través de una propiedad P , por ejemplo, $genre(Yesterday, rock)$ está “buscando” la propiedad $genre$ entre los individuos $Yesterday$ y $rock$;
- $Q(x, z)$ - verifica si una Data Property Q de un individuo x tiene un valor de z ;
- $sameAs(x, y)$ - se mantiene si los individuos x y y son los mismos;
- $differentFrom(x, y)$ - se mantiene si los individuos x y y son diferentes.

Todos los átomos presentados anteriormente se pueden utilizar con variables en lugar de individuos. El átomo de $P(x, y)$ se puede utilizar de la siguiente manera - $genre(?t, rock)$ para representar la una pregunta: ¿qué pistas pertenecen al género del $rock$?

Perfiles de usuario

La personalización en el contexto de los sistemas de recomendación se puede definir como el proceso de personalización de los contenidos a las necesidades específicas e individuales de cada usuario [191]. Para esto es necesario conocer el perfil de usuario, el cual se utiliza para identificar los intereses de los usuarios y preferencias. Existen varios enfoques adecuados para la construcción de un perfil de usuario [192] [193] [194], principalmente se pueden dividir en dos categorías: 1) basada en el conocimiento, y 2) basada en el comportamiento [195]. El primero considera un modelo de usuario estático y utiliza cuestionarios y entrevistas para que coincida con un modelo de usuario a uno de los modelos ya existentes, mientras que el segundo construye un modelo de un usuario en función de los patrones descubiertos a partir del comportamiento del usuario aplicando técnicas de aprendizaje automático. La mayoría de los sistemas de recomendación utilizan un enfoque basado en el comportamiento y modelan un usuario de un modo binario. Como se mencionó anteriormente el perfil contextual del turista para la presente propuesta es construido por medio de un modelo de clasificación basado en Aprendizaje Automático.

Personalización basada en Ontología

Los enfoques de perfiles de usuario basados en Ontología están diseñados para reducir la brecha semántica entre las características de bajo nivel extraídas de documentos, como la bolsa de las palabras, y las más abstractas, vistas conceptuales de intereses de los usuarios [196]. Los sistemas actuales utilizan ontologías extendiendo el modelo de la bolsa de palabras [197] [195]. Las Ontologías de dominio se utilizan para unir los conceptos en los perfiles y los objetos mediante subsunción de jerarquías para generalizar los conceptos [93]. En esta tesis, los perfiles de usuario se modelan a través de una ontología de dominio como se describió en la sección 4.5.1.

4.5. Fases del enfoque propuesto

Reglas e Inferencia

El lenguaje de ontologías puede ser visto como una especialización de la lógica de predicados la cual puede mejorar la inteligencia de máquina, manejable y está soportado por una herramienta de razonamiento eficiente mediante la implementación de la lógica y la interferencia o reglas. De acuerdo con la lógica, es más general que las ontologías ya que puede ser utilizada por agentes inteligentes para la toma de decisiones y tiene más explicación que el lenguaje de ontologías. Las reglas son responsables de lo que representa la relación entre los hechos. El sistema de inferencia consiste de reglas de la forma:

```
SI E contiene ciertos triples
  ENTONCES añadir a E ciertos triples adicionales donde E
  es un conjunto arbitrario de triples RDF
E.
```

Diseño de la Ontología

La Ontología que se emplea para la presente propuesta es diseñada basada en las dimensiones presentadas en el modelo contextual presentado en la sección 3.1.2, la cual relaciona el Usuario, su tiempo disponible y su perfil con las Actividades turísticas y su duración, los Sitios donde se realizan dichas actividades, la Información de Tiempo y Ubicación, la cual sirve como base de conocimientos para apoyar la definición de las relaciones semánticas y las instalaciones de razonamiento. El recomendador toma conocimiento sobre el turista (contexto de usuario), el conocimiento de contenidos y el conocimiento sobre el dominio en consideración y por ende se utiliza para modelar y representar este tipo de conocimiento y diseñar las reglas de recomendación sobre la misma.

La Ontología fue desarrollada usando el software Protege [198]. Protege es una herramienta de software libre que se desarrolló para la construcción y edición de las ontologías. La ontología se construyó mediante la definición de las clases o subclases y propiedades y para algunas de las dimensiones se emplearon ontologías base, específicamente tres ontologías, de las cuales se obtuvieron los conceptos que estaban directamente relacionados con el proyecto. El modelo del Perfil de Usuario se basó en “The USERMODELONTOLOGY”[199], dentro de la cual los conceptos relevantes de la ontología general de modelo de usuario GUMO se puede acceder en línea a través del navegador UbiOntology bajo la URL:

`http://www.u2m.org/UbisWorld/UbisBrowser.php}`

o en RDFS, DAML y OWL bajo

`http://www.u2m.org/UbisWorld/UbisOntology.php?window=UserModelOntology OWL`

`http://www.u2m.org/UbisWorld/UbisOntology.php?window=UserModelOntology DAML`

`http://www.u2m.org/UbisWorld/UbisOntology.php?window=UserModelOntology RDFS`

Las dimensiones básicas modelo de usuario se organizan en varios grupos de propiedades como se describe a continuación (Tabla 4.6):

| Group | Class | Id |
|------------------------|-----------------------|--------|
| UserModelPropertyGroup | MentalState | 700017 |
| UserModelPropertyGroup | PhysicalState | 700016 |
| UserModelPropertyGroup | Demographics | 700010 |
| UserModelPropertyGroup | ContactInformation | 700008 |
| UserModelPropertyGroup | Role | 700019 |
| UserModelPropertyGroup | EmotionalState | 700014 |
| UserModelPropertyGroup | Personality | 700012 |
| UserModelPropertyGroup | Characteristics | 700013 |
| UserModelPropertyGroup | AbilityAndProficiency | 700011 |

Tabla 4.6: Dimensiones de Usuario tomadas de la ontología [199]

Dentro de las dimensiones básicas del modelo de usuario que se consideraron de acuerdo con las características del enfoque propuesto son (Tabla 4.7):

4.5. Fases del enfoque propuesto

| Group | Class | Id |
|--------------------|-------------------|--------|
| Demographics | Gender | 800300 |
| Demographics | AgeGroup | 800301 |
| Demographics | Age | 800302 |
| Demographics | Birthday | 800303 |
| Demographics | Birthplace | 800304 |
| Demographics | FirstLanguage | 800306 |
| Demographics | SecondLanguage | 800308 |
| Demographics | FamilyStatus | 800312 |
| Demographics | EducationLevel | 800314 |
| Demographics | Employment | 800316 |
| Demographics | Salary | 800318 |
| Demographics | Wealth | 800320 |
| ContactInformation | GivenName | 800410 |
| ContactInformation | MiddleName | 800411 |
| ContactInformation | FamilyName | 800412 |
| ContactInformation | FullName | 800414 |
| ContactInformation | Street | 800416 |
| ContactInformation | HouseNumber | 800418 |
| ContactInformation | PostalCode | 800420 |
| ContactInformation | City | 800422 |
| ContactInformation | State | 800424 |
| ContactInformation | Country | 800426 |
| ContactInformation | TelephoneNumber | 800430 |
| ContactInformation | MobilePhoneNumber | 800432 |
| ContactInformation | FaxNumber | 800433 |
| ContactInformation | Email | 800434 |
| ContactInformation | Homepage | 800436 |

Tabla 4.7: Características de las dimensiones del modelo de Usuario tomadas de la ontología [199]

Y adicionalmente como ya se mencionó anteriormente, es necesario conocer la Tipología y el motivo de Viaje de los turistas.

Para el modelo del Perfil de Usuario se tomó como base la ontología de Usuario del proyecto Travel Guide Ontology [171] que contiene conceptos relacionados con los viajeros. Esta ontología describe intereses de los usuarios y actividades, grupos de edad, las compañías de viajes preferidas, y otros datos sobre los diferentes tipos de usuarios y perfiles de usuario, para lo cual ampliaron el módulo de gestión del conocimiento PROTON con los conceptos y las relaciones de dominio-relacionados para la ontología del usuario (Tabla 4.8):

| Type of User Profile |
|----------------------|
| Adventurer |
| Artist |
| ClassicTourist |
| Clubbing-type |
| Conformist |
| Cosmopolit |
| FamilyType |
| Historic |
| Hooliggan |
| NatureLover |
| Passivist |
| Romantic |
| Shopaholic |
| Thinker |

Tabla 4.8: Tipos de Perfil de Usuario tomados de la ontología la ontología de Usuario del proyecto Travel Guide Ontology [171]

Finalmente dentro del dominio turístico y para el modelo de Datos, es decir los diferentes tipos de actividades a realizar en los sitios para realizar las recomendaciones, se tomó como base la ontología denominada “ETP-Tourism”[200], de las cuales fueron escogidos los mencionados en la tabla 3.2.

Se agregaron los siguientes tipos adicionales (Tabla 4.9):

| Type of Activity |
|------------------|
| Art |
| Culture |
| Education |
| Religion |

Tabla 4.9: Tipos de actividades turísticas adicionales al diseño de la ontología del proyecto

Las filas sombreadas en algunas tablas son las clases que finalmente fueron empleadas en la ontología construida de acuerdo a las características y requerimientos de la presente propuesta. A continuación se describe el proceso para el diseño de la misma:

- Clases o Subclases de la ontología: Las clases se definen mediante un elemento owl: Class. La estructura de clase de la ontología se puede ilustrar en la Figura 4.14. Por ejemplo, la clase

4.5. Fases del enfoque propuesto

Adventure definida como sigue:

```
<owl:Class rdf:ID="Adventure">
<rdfs:label>Adventure</rdfs:label>
<rdfs:subClassOf rdf:resource="\#Activity" />
</owl:Class>
```

- Propiedades de la ontología: En OWL hay dos tipos de propiedades:
 - Propiedades de objeto, que se relacionan los objetos con otros objetos. He aquí un ejemplo de una propiedad de objeto que está en la forma de OWL:

```
<owl:ObjectProperty rdf:ID="hasTimeDuration">
<rdfs:domain rdf:resource="\#Activity" />
</owl:ObjectProperty>
```

- Propiedades de tipo de datos, que relacionan objetos a valores de tipo de datos. He aquí un ejemplo de una propiedad de tipo de datos que se encuentra en la forma de OWL:

```
<owl:ObjectProperty rdf:ID="hasName">
<rdfs:domain rdf:resource="\#User" />
</owl:ObjectProperty>
```

Recomendación basada en conocimiento

Las dimensiones del modelo contextual se recogieron en la ontología OWL anteriormente descrita que sirve como base de conocimientos para apoyar la definición de las relaciones semánticas y las facilidades de razonamiento. El recomendador toma conocimiento sobre el turista (contexto de usuario), el conocimiento de contenidos y conocimiento sobre el dominio en consideración, la ontología se utiliza para modelar y representar a ese tipo de conocimiento.

Fundamentalmente, un sistema de recomendación es el sistema que predice la preferencia del usuario para sugerir ítems por el uso de la técnica de la información del sistema de filtrado. Un Sistema de recomendación basado en ontologías por medio de reglas tiene como objetivo mostrar los resultados de recuperación o recomendaciones para el usuario objetivo.

El recomendador de conocimiento está formado por la base de conocimiento (KB) y el motor de razonamiento. Se construyeron reglas utilizando un motor de inferencia, tales reglas guían el proceso de recomendación y la salida de este proceso es un conjunto de recursos pertinentes, para este caso, esos recursos son actividades turísticas relacionadas con un turista.

Reglas Recomendación

Cuando se detecta un cambio en el contexto del usuario, los datos de contexto recogidos se transfieren al recomendador basado en conocimiento (KB), el cual instancia del modelo de contexto y llama al motor de razonamiento que analiza las reglas de contexto almacenadas en la base de conocimientos.

Dichas reglas fueron construidas en SWRL y tienen la forma de una implicación entre el antecedente (el cuerpo) y el consecuente (cabeza), es decir, cada vez que las condiciones especificadas en el antecedente sostiene, el consecuente también se sostiene. Siempre que el motor de Razonamiento encuentra una coincidencia entre la información instanciada y las reglas, las actividades turísticas

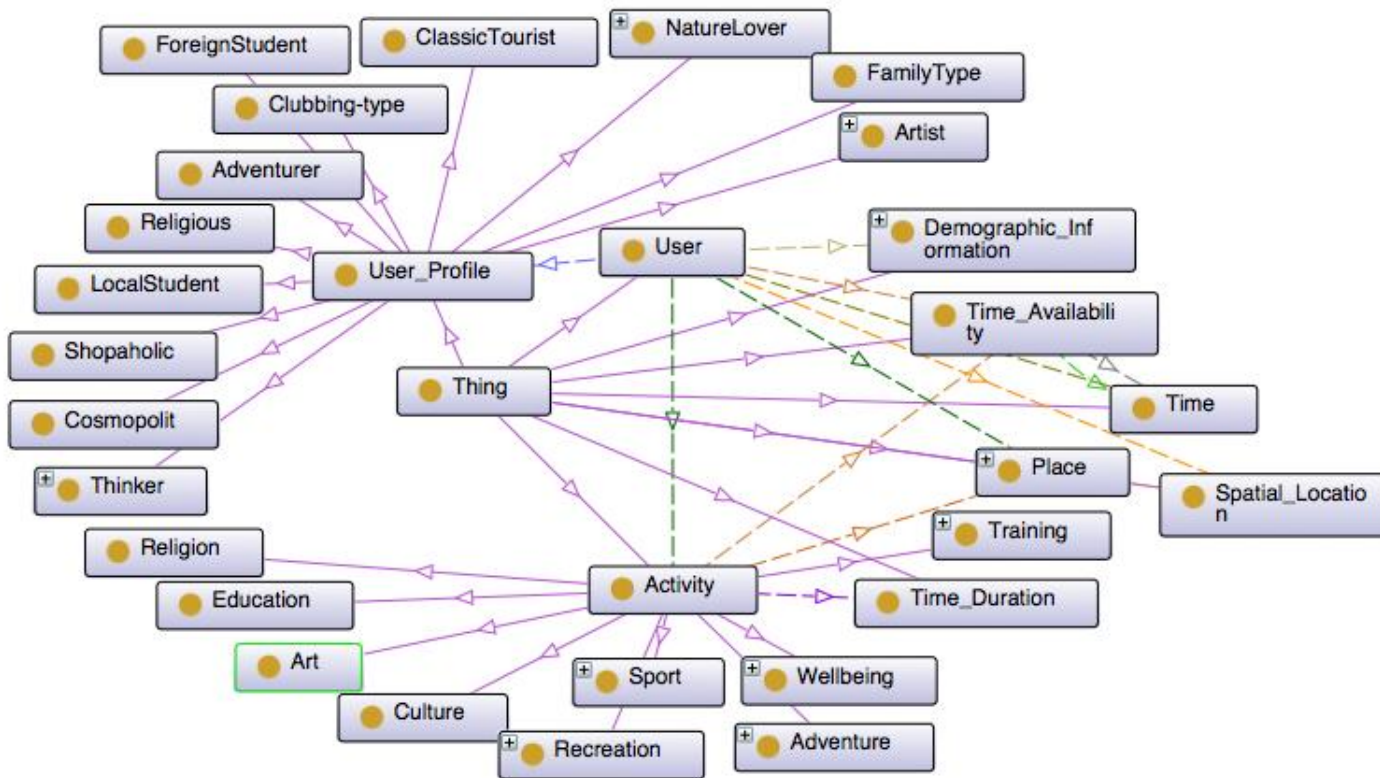


Figura 4.14: Estructura de la Ontología empleada en el sistema.

que asisten a esa especificación son seleccionados por el sistema. A continuación se ejemplifican dos clases de reglas usadas por el sistema:

Las primeras clases de reglas relacionan los tipos de perfiles turísticos con los tipos de actividades de acuerdo con la definición de perfiles contextuales especificados en la tabla 4.4:

- FamilyType(?p), Outdoor(?a), User(?u),
hasUserProfile(?u, ?p) -> hasRecommended(?u, ?a)
- Education(?a), Thinker(?p), User(?u),
hasUserProfile(?u, ?p) -> hasRecommended(?u, ?a)

Las segundas clases de reglas tienen en cuenta la disponibilidad de cada una de las actividades turísticas, para recomendar actividades que se puedan realizar en el horario en que el turista se encuentra:

```
Recreation(?a), Time(?finishTime), Time(?startTime),
Time(?userTime), Time_Availability(?ta), User(?u),
hasCurrentDate(?u, ?userTime), hasFinishTime(?ta, ?finishTime),
hasStartTime(?ta, ?startTime), hasTimeAvailability(?a, ?ta),
hasHour(?finishTime, ?hourFinishTime),
```


4.5. Fases del enfoque propuesto

```
hasHour(?startTime, ?hourStartTime), hasHour(?userTime, ?hUserTime),  
greaterThan(?hUserTime, ?hourStartTime),  
lessThan(?hUserTime, ?hourFinishTime) -> hasRecommended(?u, ?a)
```

En el cuerpo de la regla, la variable *User* corresponde con *Activity* (el tipo de actividad) en el ejemplo a través de la relación *hasUserProfile*, dicha instancia pertenece a la dimensión *UserProfile* y tiene el valor de tipo familiar (*FamilyType*).

Si todas las condiciones establecidas en el cuerpo de la regla se cumplen, entonces la parte de la cabeza de ella se emparejan con las instancias presentes en el modelo de contexto y, finalmente, los respectivos componentes se presentan al usuario. En el ejemplo anterior, si todas las condiciones se cumplen entonces el usuario se empareja con las actividades a través de la relación *hasRecommended*.

Se observa que el uso de dichas reglas con el modelo de contexto extrae la recomendación semántica y la coloca en el modelo ontológico. Una ventaja de esta característica es que el conocimiento del dominio es modelado por la ontología y las reglas pueden manipular los conceptos de la misma, incluyendo el contexto de usuario.

4.5.3. Recomendación por Filtrado Colaborativo

La siguiente fase del enfoque propuesto es el uso de filtrado colaborativo para predecir las preferencias del usuario y generar recomendaciones. El Filtrado colaborativo en su implementación más simple que recomienda al usuario activo, los ítems que otros usuarios con preferencias similares les han gustado en el pasado. Estos sistemas de filtrado colaborativo producen predicciones o recomendaciones para un usuario determinado y uno o más ítems. Los ítems pueden consistir en cualquier cosa para la que un ser humano puede proporcionar una calificación, como el arte, libros, CDs, artículos de revistas, o destinos de vacaciones. Los ratings en un sistema de filtrado de colaboración pueden adoptar una variedad de formas:

- Calificaciones escalares: pueden consistir en cualquiera de las calificaciones numéricas, como las 1 a 5 estrellas proporcionadas en MovieLens o calificaciones ordinales como muy de acuerdo, de acuerdo, neutral, en desacuerdo, totalmente en desacuerdo.
- Opciones binarias de calificaciones como: de acuerdo/en desacuerdo o bueno/malo.
- Calificaciones unarias pueden indicar que un usuario ha observado o comprado un artículo, o de otro modo calificar el ítem positivamente. La ausencia de una calificación indica que no se tiene información relativa al usuario al ítem (tal vez ha comprado el ítem en otro lugar).

Las calificaciones pueden ser recogidas a través de los medios explícitos, implícitos, o ambos. Las calificaciones explícitas son aquellas en las que se pide a un usuario proporcionar una opinión sobre un ítem. Las calificaciones implícitas son aquellas que pueden inferirse de las acciones del usuario. Por ejemplo, un usuario que visita una página de productos tal vez tiene algo de interés en ese producto, mientras que un usuario que posteriormente adquiere el producto puede tener un interés mucho más fuerte en ese producto.

Usos del filtrado colaborativo

En esta sección se consideran cuestiones relacionadas con el uso del CF mediante la exploración de las tareas del usuario que soporta, a continuación, los servicios que proporcionan los sistemas de CF.

Las tareas para las cuales el filtrado colaborativo se ha usado y que se han estudiado son:

- Ayudar a los usuarios a encontrar nuevos ítems que le podrían gustar. En un mundo de la sobrecarga de información, no es posible evaluar todas las cosas. Presentar algunos para poder elegir se ha aplicado con mayor frecuencia a los ítems de consumo (música, libros, películas), pero también puede ser aplicada a la investigación los documentos, páginas web u otros elementos calificables.
- Asesorar sobre un ítem en particular. Si se tiene un tema en particular en mente; la comunidad sabe si es bueno o malo?
- Ayudar a encontrar un usuario (o algunos usuarios) que le puedan gustar. Esto podría ayudar con la formación de grupos de discusión [201], matchmaking, o conectar a los usuarios para que puedan intercambiar recomendaciones socialmente.

Los recomendadores colaborativos explotan los métodos de vecinos más cercanos (método de vecindad basada en usuario y el método de vecindad basado en ítem). Los métodos basados en usuario se basan en la opinión de los usuarios afines para predecir una calificación, y generar recomendaciones. Los enfoques basados en ítem toman las calificaciones otorgadas a los ítems similares y generan recomendaciones. La complejidad computacional de los métodos basados en usuario escala como el número de usuarios aumenta, mientras que la de los métodos basados en ítem sube como el número de ítems crece.

Algoritmos de vecindad basada en usuario

Estos algoritmos generan predicciones para los usuarios basándose en puntuaciones de usuarios similares, los cuales son llamados usuarios vecinos similares. Si un usuario n es similar a un usuario u , se dice que n es un vecino de u . Algoritmos basados en el usuario generan una predicción para un ítem i mediante el análisis de ratings de i de los usuarios en la vecindad de u [202].

El algoritmo de recomendación basado en el usuario sale de esta intuición. Es un proceso de recomendar ítems a algún usuario, denotado por u , como se ve a continuación:

```
every item i that u has no preference for yet every other user v that has a preference for i
compute a similarity s between u and v      incorporate v's preference for i, weighted by s,
into a running average      return the top items, ranked by weighted average
```

El bucle externo simplemente considera todos los ítems conocidos (para los que el usuario no ha expresado su preferencia) como candidato a la recomendación. El bucle interno busca cualquier otro usuario que ha expresado su preferencia por este ítem candidato, y obtiene su valor de preferencia para el mismo. Al final, los valores se promedian para llegar a una estimación - un promedio ponderado. Cada valor de preferencia se pondera en el promedio por que tan similar es el usuario al usuario activo. Para el usuario más similar, su valor de preferencia es ponderado más fuertemente.

En realidad, una vecindad de la mayoría de usuarios similares se calcula primero, y sólo los ítems conocidos por los usuarios son considerados:

```
for every other user w? compute a similarity s between u and w?
retain the top users, ranked by similarity, as a neighborhood n
for every item i that some user in n has a preference for,
but that u has no preference for yet
for every other user v in n that has a preference for i?
```

4.5. Fases del enfoque propuesto

compute a similarity s between u and v
incorporate v 's preference for i , weighted by s , into a
running average

La principal diferencia es que los usuarios similares se encuentran en primer lugar, antes de ver en lo que los usuarios más similares están interesados. Esos elementos se convierten en los candidatos a la recomendación.

Vecindades de usuario

Las técnicas, la correlation-thresholding y best-n-neighbors, se han utilizado para determinar cuántos vecinos seleccionar. Con ambas técnicas, las contribuciones de las pocas excepcionalmente valiosas altas correlaciones tienden a perderse en el ruido de las muchas correlaciones más bajas cuando los algoritmos están configurados para dar cobertura aceptable.

La primera técnica, utilizado por Shardanand y Maes [203], es establecer un umbral de correlación absoluta, donde se seleccionan todos los vecinos con correlaciones absolutas mayores que un umbral dado. El establecimiento de un umbral alto limita su vecindario para que contenga muy buenas correlaciones, pero para muchos usuarios altas correlaciones no están disponibles, lo que resulta en un pequeño vecindario que no pueden proporcionar la cobertura de predicción para muchos ítems. Una configuración de un umbral de correlación bajo resulta en un gran número de correlaciones más bajas, anulando el propósito del umbral. El umbral debe estar entre -1 y 1, ya que todas las métricas de similitud devuelven valores de similitud en este rango, la literatura recomienda explorar muchos valores de datos reales para determinar el óptimo.

La segunda técnica consiste en escoger la mejor n correlación para un n dado. Esta técnica se realiza razonablemente bien, ya que no limita la cobertura de predicción. Sin embargo, la selección de un mayor n producirá demasiado ruido para aquellos que tienen altas correlaciones. Escoger un menor n puede causar predicciones pobres para aquellos usuarios que no tienen ninguna correlación alta. Una vez más la literatura recomienda evaluar varios valores y averiguar la configuración óptima para el conjunto de datos con el que se este trabajando, pero afirman que no hay valor mágico; un poco de experimentación con datos reales es necesario para delimitar el recomendador.

Métricas de similitud

Una decisión de diseño crítico en la implementación de un CF basado en Usuario es la elección de la función de similitud. Se han propuesto y evaluado en la literatura diferentes funciones de similitud.

- Correlación de Pearson: Este método calcula la correlación estadística (r de Pearson) entre dos ratings comunes de los usuarios para determinar su similitud. GroupLens y Bellcore utilizan este método [204] [205]. La correlación se calcula por la siguiente fórmula:

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

La correlación de Pearson sufre de computación de alta similitud entre los usuarios con pocas calificaciones en común. Esto puede ser aliviado mediante el establecimiento de un umbral en

el número de ítems co-rateados necesarios para el acuerdo total (correlación de 1) y escalando la similitud cuando el número de ítems co-rateados cae por debajo de este umbral [206] [41]. Los experimentos han demostrado un valor umbral de 50 útil en la mejora de la exactitud de la predicción, y el umbral pueden ser aplicado al multiplicar la función de similitud por $\min\{|I_u \cap I_v|/50, 1\}$.

- **Correlación de Spearman:** El coeficiente de correlación de Spearman es otro candidato para una función de similitud [206]. Para la correlación de Spearman, los ítems que un usuario ha calificado están clasificados de manera que los de mayor calificación están en el rango 1 y los ítems de menor calificación tienen rangos más altos. A los ítems con la misma calificación se les asigna el rango promedio para su posición. El cálculo es entonces el mismo que el de la correlación de Pearson, excepto que se utilizan los rangos en lugar de las calificaciones.
- **Similitud del Coseno:** Este modelo es algo diferente de los enfoques descritos anteriormente, ya que es un enfoque de vector-espacio basado en álgebra lineal en lugar de un enfoque estadístico. Los usuarios se representan como n -vectores dimensionales y la similitud se mide por la distancia coseno entre dos vectores de calificación. Esto se puede calcular de manera eficiente mediante la adopción de su producto escalar y dividiéndolo por el producto de su norma L2 (euclidiana):

$$s(u, v) = \frac{r_u \cdot r_v}{\|r_u\|_2 \|r_v\|_2} = \frac{\sum_i r_{u,i} r_{v,i}}{\sqrt{\sum_i r_{u,i}^2} \sqrt{\sum_i r_{v,i}^2}} \quad (3)$$

- **Distancia Euclidiana [207]:** La distancia euclidiana es una métrica familiar por la geometría de 2 y 3 dimensiones. La distancia euclidiana $r_2(x, y)$ entre dos vectores 2-dimensionales $x = (x_1, x_2)^T$ y $y = (y_1, y_2)^T$ viene dada por:

$$r_2(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} = \sqrt{\sum_{i=1}^2 (x_i - y_i)^2} \quad (4)$$

- **Coficiente de Tanimoto [202]:** Curiosamente, también hay implementaciones de similitud por el usuario que ignoran valores de preferencia completamente. No importa si un usuario expresa una preferencia alta o baja para un sólo ítem, sólo que el usuario exprese una preferencia en absoluto, el coeficiente de Tanimoto es una implementación de esto, el cual también se conoce como el coeficiente de Jaccard. Es el número de ítems para los que dos usuarios expresan alguna preferencia por éstos, dividido por el número de ítems por los que el usuario expresa cierta preferencia, como se ilustra en la figura 4.15:

En otras palabras, es la relación entre el tamaño de la intersección con el tamaño de la unión de sus ítems preferidos. Tiene las propiedades requeridas: cuando los ítems de dos usuarios se superponen completamente, el resultado es 1,0. Cuando no tienen nada en común, que es 0,0. El valor nunca es negativo, pero eso está bien. Su expresión con más de dos vectores de bits se puede escribir como:

4.5. Fases del enfoque propuesto

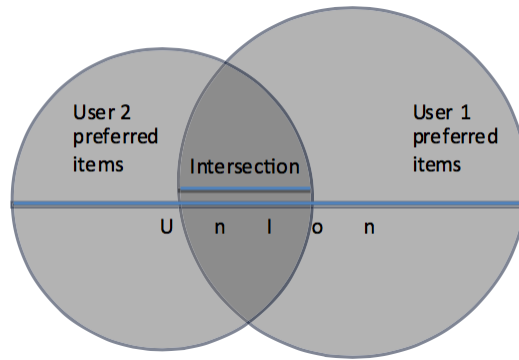


Figura 4.15: Coeficiente de Tanimoto.

$$T(A, B) = \frac{A \cdot B}{|A|^2 + |B|^2 - A \cdot B} = \frac{\sum_{i=1}^n A_i \times B_i}{\sum_{i=1}^n (A_i)^2 + \sum_{i=1}^n (B_i)^2 - \sum_{i=1}^n A_i \times B_i} \quad (5)$$

- Similitud Log Likelihood [202]: similar a la similitud basada en el coeficiente de Tanimoto, ya que es otra métrica que no toma en cuenta los valores de preferencias individuales. Al igual que el coeficiente de Tanimoto, está basado en el número de ítems en común entre dos usuarios, pero su valor es más una expresión de lo improbable que es para dos usuarios tener alta superposición, dado el número total de ítems existentes y el número de ítems para los que cada usuario tiene una preferencia. La similitud log-likelihood se calcula tratando de evaluar lo improbable que es que la superposición entre los dos usuarios es sólo debido a la casualidad. Es decir, dos usuarios distintos, sin duda evaluarán un par de películas en común, pero dos usuarios similares mostrará una superposición que se ve muy poco probable que sea una casualidad. Con algunas pruebas estadísticas, esta similitud intenta determinar qué tan probable es que dos usuarios no tienen ningún parecido en sus gustos; más improbable, más similares los dos deben ser. El valor de similitud resultante puede ser interpretada como una probabilidad de que la superposición no se debe al azar.

Diseño del Filtro Colaborativo

Muchos frameworks de código abierto se han desarrollado para la construcción, la investigación y el estudio de los sistemas de recomendación, como Apache Mahout [208], LensKit [209], Waffles [210], Recommenderlab [211]. Fueron construidos con diferentes lenguajes de programación, entre los cuales, Mahout y LensKit se basan en Java; Waffles se basa en C++; Crab se basa en Python; y el lenguaje R se adopta en Recommenderlab. Para el contexto de esta tesis, la implementación del Filtro Colaborativo se ha realizado con Apache Mahout.

La introducción detallada a Apache Mahout se puede encontrar en su sitio web [208]. Algunos de los puntos se destacan en esta sección. El motor de recomendación dentro de Apache Mahout se implementa a través de Taste, el cual puede ser considerado como un componente flexible y maduro y tipo de independiente dentro Mahout. No sólo es compatible con los enfoques básicos de CF basados en usuario basada y en ítems, sino que también proporciona interfaces extensibles para conectar y realizar recomendaciones personalizadas de los usuarios. Taste tiene cinco interfaces de paquetes como

abstracciones clave para llevar a cabo las recomendaciones: `DataModel` es un conector para extraer la información de las preferencias del usuario de la fuente de datos. `JDBCDataModel` y `FileDataModel` se usan para leer la información de base de datos y archivos, respectivamente; `Usersimilarity` e `Item-similarity` son otro paquete de interfaces para averiguar usuarios o ítems similares para los usuarios o ítems específicos, denominados vecindades. Un algoritmo de similitud es el núcleo del motor de recomendación de CF. Muchos algoritmos de similitud populares de los paquetes `Taste`, como la similitud de correlación de Pearson, distancia de similitud Euclidiana, similitud de correlación de Spearman, coeficiente de similitud Tanimoto, similitud del coseno, y así sucesivamente para satisfacer las necesidades diferentes de los usuarios; `UserNeighborhood` es especialmente para la recomendación basada en el usuario, que generan resultados de recomendación de los vecinos del usuario dado. En el modelo `UserNeighborhood`, se podría definir un número diferente de vecindad para afinar el resultado de recomendación. Por lo general, la vecindad se encuentra por `UserSimilarity`; la última interfaz es `Recommender`, que implementa la recomendación. Proporcionando un `DataModel`, un `Recommender` podría generar la predicción, haciendo uso de la `GenericUserBasedRecommender` o `GenericItemBasedRecommender`.

Parámetros de configuración del recomendador CF

El conjunto de datos a emplear para el recomendador fueron los obtenidos en la experiencia del caso de estudio realizada bajo el Proyecto de Trazabilidad Turística de la Universidad del Cauca como se mencionó en la sección 3.1.2.

Evaluación de las métricas de similitud

Los algoritmos de recomendación disponibles en Apache Mahout incluyen Filtrado Colaborativo basado en usuarios e ítems, recomendadores basados en Matrix de factorización, clustering K-means y Fuzzy K-means y así sucesivamente [208]. Las predicciones generadas por el motor de recomendación se basan en las correlaciones entre los usuarios o ítems a una amplia extensión. Casi todos los esfuerzos realizados en el enfoque de recomendación sobre el tratamiento de este problema. La forma en la medición directa de la similitud entre los usuarios o los ítems de acuerdo con los perfiles de los usuarios o las descripciones de los ítems por medio del recomendador de filtrado basado en contenido. Si se toman los comportamientos de los usuarios en cuenta y la similitud se basa calculado sobre estas preferencias, la recomendación pertenece a un recomendador CF.

Para la presente propuesta, dado que los valores de preferencia del conjunto de datos son unarios, las métrica de similitud que aplican para este tipo de datos booleanos son el coeficiente Tanimoto y la Similitud Log Likelihood, por lo cual estas dos métricas fueron las que se emplearon sobre el dataset para evaluar el número de usuarios para los que el recomendador es capaz de recomendar, el número máximo de recomendaciones por usuario, el número de usuarios para cada número máximo de recomendaciones, Precision y Recall, variando el número de vecinos más cercanos para el caso de la técnica de vecindad basada en Nearest Neighborhood y variando el umbral de vecindad para el caso de la técnica basada en Neighborhood Threshold, con el fin de elegir la configuración que mejor se comporte con el dataset disponible.

Los gráficos de la Figura 4.16, Figura 4.17, Figura 4.18 y la Figura 4.19 informan de la evaluación para la Similitud Log Likelihood basada en la técnica de vecindad Nearest Neighborhood, variando el número de vecinos más cercanos tomando valores entre 100 y el número máximo de usuarios del dataset (1246), con un intervalo de 100 entre cada uno:

4.5. Fases del enfoque propuesto

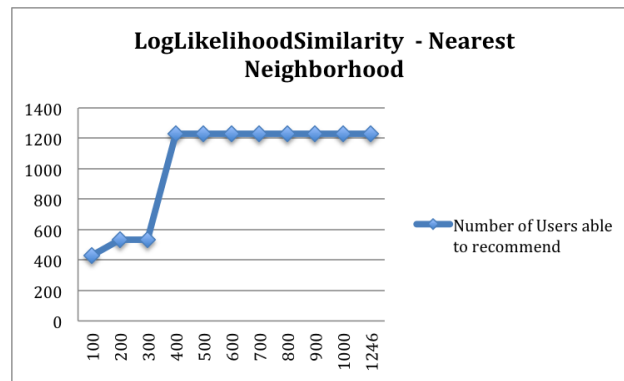


Figura 4.16: Evaluación del número de usuarios para los que el recomendador es capaz de recomendar - Similitud Log Likelihood basada en Nearest Neighborhood.

Como se observa en la anterior figura el número de usuarios para los que el recomendador es capaz de recomendar alcanza su máximo valor (1230) a partir del valor de vecindad 400.

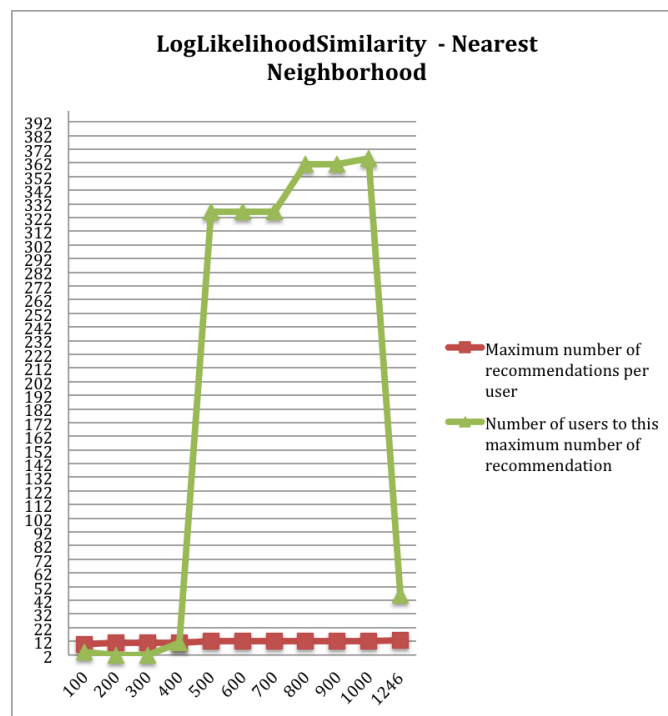


Figura 4.17: Evaluación del número máximo de recomendaciones por usuario y número de usuarios para cada número máximo de recomendaciones - Similitud Log Likelihood basada en Nearest Neighborhood.

La anterior figura muestra que el máximo número de recomendaciones por usuario es 13 y se presenta en el valor de vecindad igual al total de usuarios del dataset (1246) para 46 usuarios. Así mismo se presenta un caso especial en el que el número de recomendaciones por usuario 12 en el valor de vecindad 1000 para 365 usuarios.

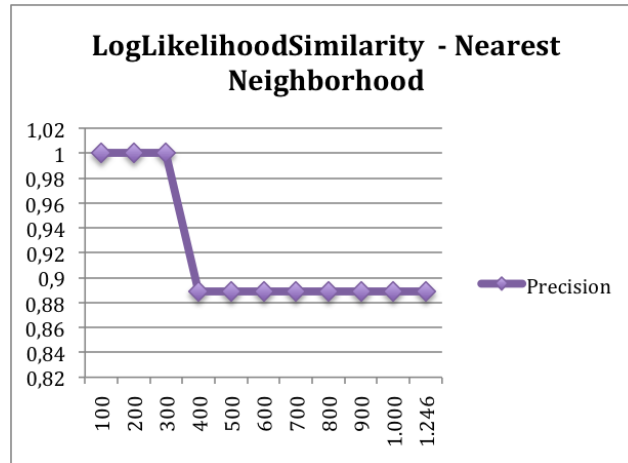


Figura 4.18: Evaluación de Precision - Similitud Log Likelihood basada en Nearest Neighborhood.

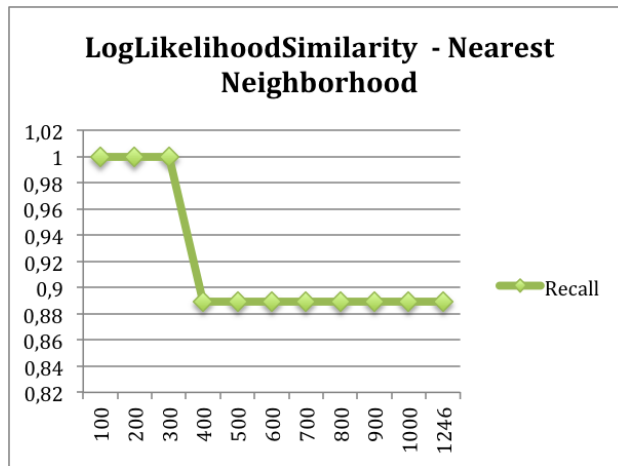


Figura 4.19: Evaluación de Recall - Similitud Log Likelihood basada en Nearest Neighborhood.

Según las figuras 4.18 y 4.19 los valores de Precision y Recall alcanzan un valor de 1 para los valores de vecindad entre 100 y 300.

Los gráficos de la Figura 4.20, Figura 4.21, Figura 4.22 y la Figura 4.23 informan de la evaluación para la Similitud Log Likelihood basada en la técnica de vecindad Neighborhood Threshold variando el umbral entre los valores 0,2 a 1.0:

4.5. Fases del enfoque propuesto

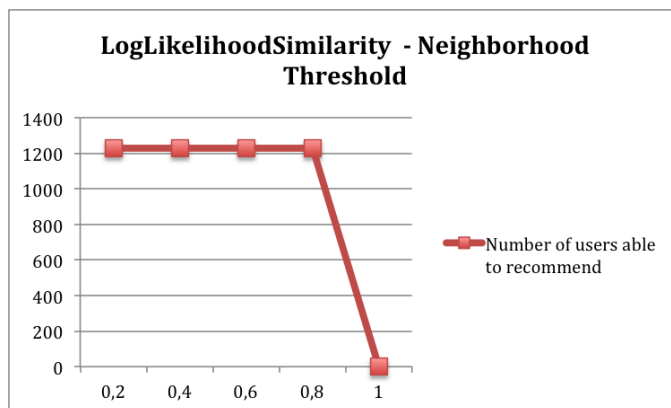


Figura 4.20: Evaluación del número de usuarios para los que el recomendador es capaz de recomendar - Similitud Log Likelihood basada en Neighborhood Threshold.

Como se observa en la anterior figura el número de usuarios para los que el recomendador es capaz de recomendar alcanza su máximo valor (1230) en valores de umbral entre 0,2 y 0,8.

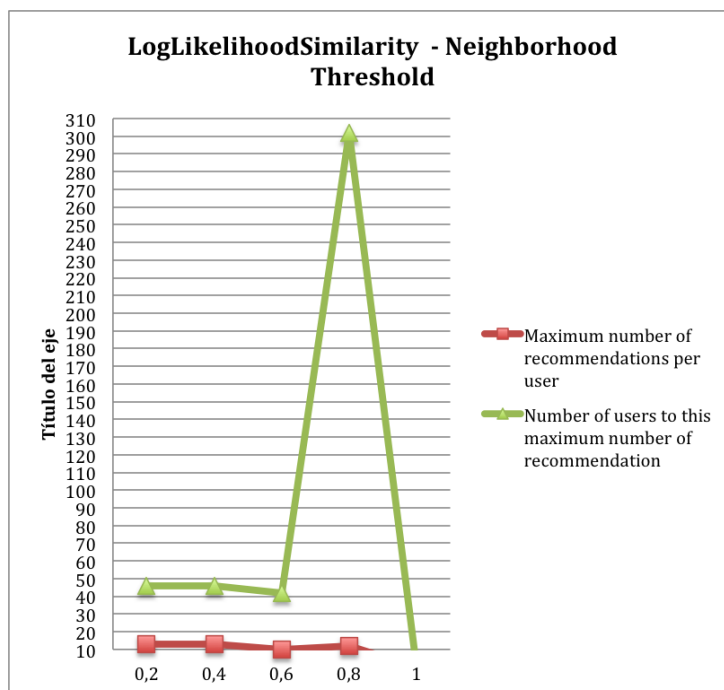


Figura 4.21: Evaluación del número máximo de recomendaciones por usuario y número de usuarios para cada número máximo de recomendaciones - Similitud Log Likelihood basada en Neighborhood Threshold.

Como se observa en la anterior figura el máximo número de recomendaciones por usuario es 13 en valores de umbral entre 0,2 y 0,4 para 46 usuarios. Así mismo se presenta un caso especial en el que el número de recomendaciones por usuario 12 en valor de umbral de 0,8 para para 302 usuarios.

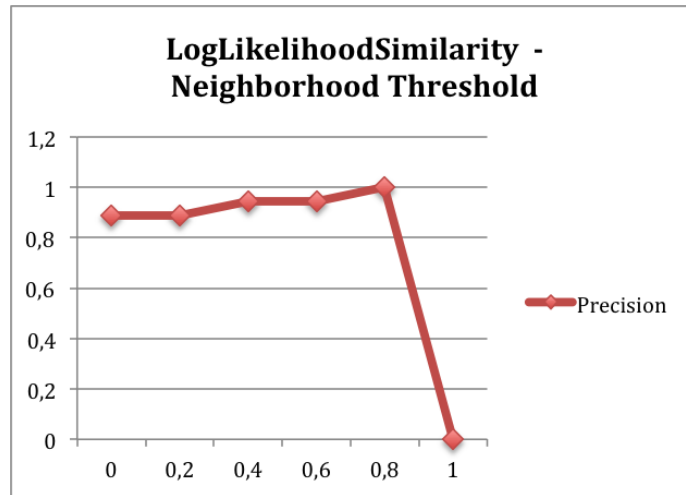


Figura 4.22: Evaluación de Precision - Similitud Log Likelihood basada en Neighborhood Threshold.

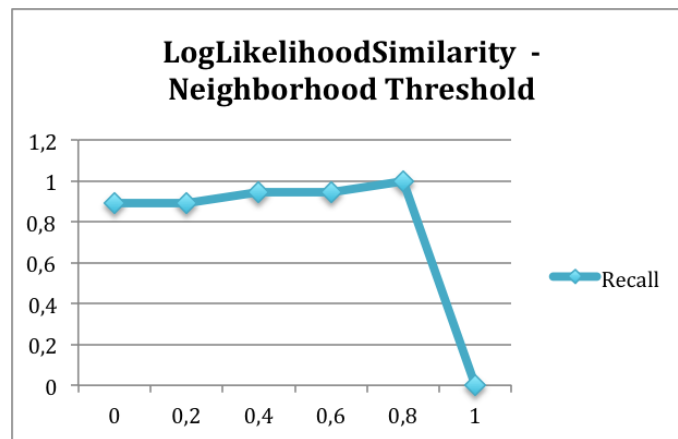


Figura 4.23: Evaluación de Recall - Similitud Log Likelihood basada en Neighborhood Threshold.

Según las figuras 4.22 y 4.23 los valores de Precision y Recall igual a 1 en valor de umbral de 0,8 para Log Likelihood y entre 0,4 y 0,6 para Tanimoto.

Los gráficos de la Figura 4.24, Figura 4.25, Figura 4.26 y la Figura 4.27, informan de la evaluación para el Coeficiente Tanimoto basado en la técnica de vecindad Nearest Neighborhood, variando el número de vecinos más cercanos tomando valores entre 100 y el número máximo de usuarios del dataset (1246), con un intervalo de 100 entre cada uno:

4.5. Fases del enfoque propuesto

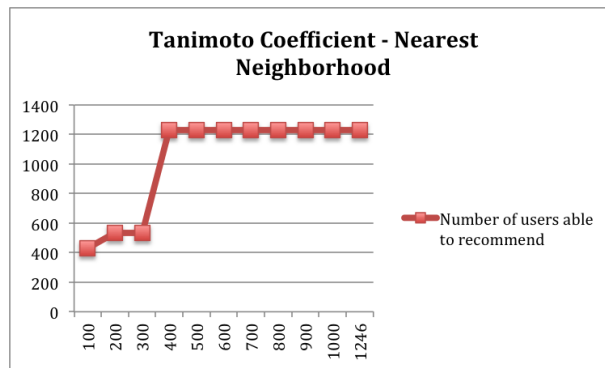


Figura 4.24: Evaluación del número de usuarios para los que el recomendador es capaz de recomendar - Similitud Log Likelihood basada en Neighborhood Threshold.

Como se observa en la anterior figura el número de usuarios para los que el recomendador es capaz de recomendar alcanza su máximo valor (1230) a partir del valor de vecindad 400.

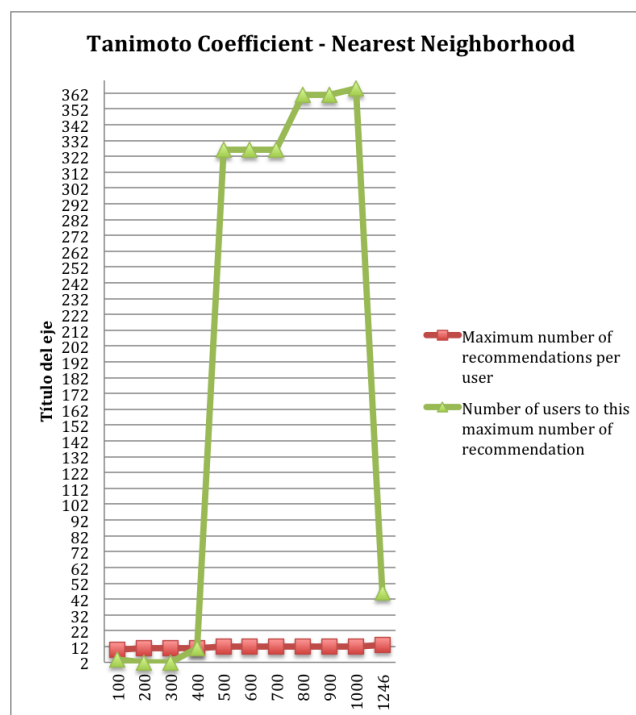


Figura 4.25: Evaluación del número máximo de recomendaciones por usuario y número de usuarios para cada número máximo de recomendaciones - Coeficiente Tanimoto basado en Nearest Neighborhood.

La anterior figura muestra que el máximo número de recomendaciones por usuario es 13 y se presenta en el valor de vecindad igual al total de usuarios del dataset (1246) para 46 usuarios. Así mismo se presenta un caso especial en el que el número de recomendaciones por usuario 12 en el

valor de vecindad 1000 para 365 usuarios.

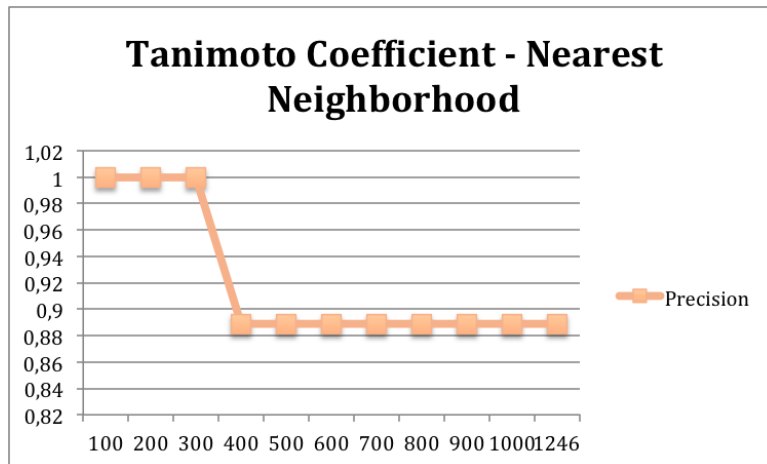


Figura 4.26: Evaluación de Precision - Coeficiente Tanimoto basado en Nearest Neighborhood.

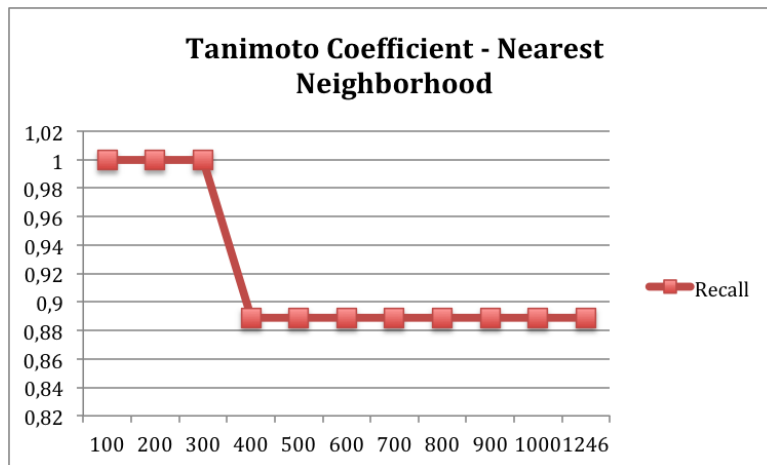


Figura 4.27: Evaluación de Recall - Coeficiente Tanimoto basado en Nearest Neighborhood.

Según las figuras 4.26 y 4.27 los valores de Precision y Recall alcanzan un valor de 1 para los valores de vecindad entre 100 y 300.

Los gráficos de las Figura 4.28, Figura 4.29, Figura 4.30 y la Figura 4.31 informan de la evaluación para el Coeficiente Tanimoto basado en la técnica de vecindad Neighborhood Threshold variando el umbral entre los valores 0,2 a 1,0:

4.5. Fases del enfoque propuesto

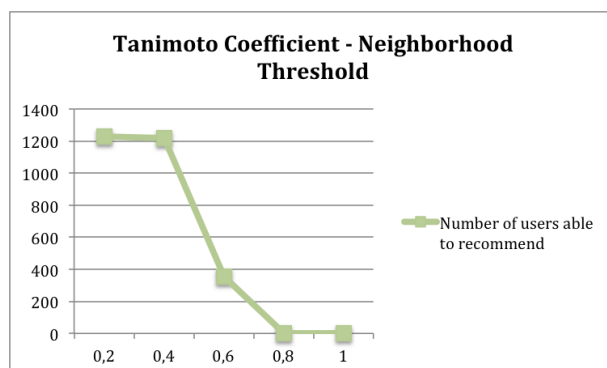


Figura 4.28: Evaluación del número de usuarios para los que el recomendador es capaz de recomendar - Coeficiente Tanimoto basado en Neighborhood Threshold.

Como se observa en la anterior figura el número de usuarios para los que el recomendador es capaz de recomendar alcanza su máximo valor (1230) en valores de umbral de 0,2.

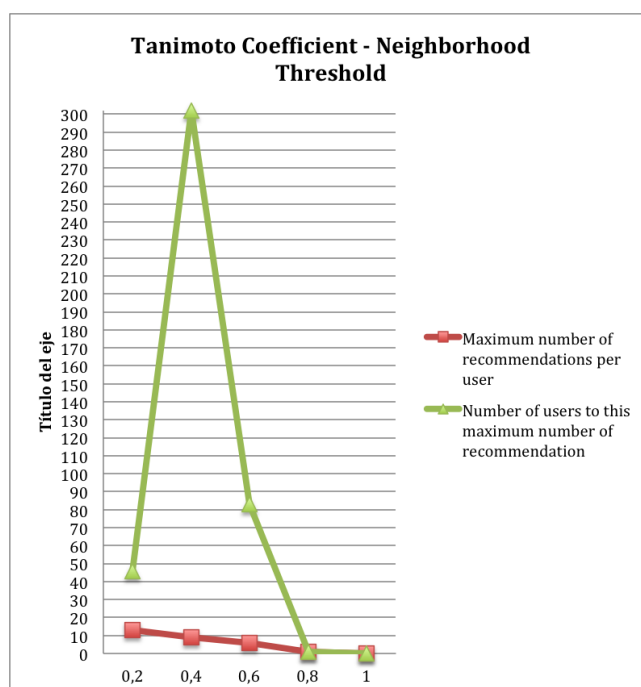


Figura 4.29: Evaluación del número máximo de recomendaciones por usuario y número de usuarios para cada número máximo de recomendaciones - Coeficiente Tanimoto basado en Neighborhood Threshold.

Como se observa en la anterior figura el máximo número de recomendaciones por usuario es 13 en valores de umbral de 0,2 para 46 usuarios. Así mismo se presenta un caso especial en el que el número de recomendaciones por usuario 12 en valor de umbral de 0,4 para 302 usuarios.

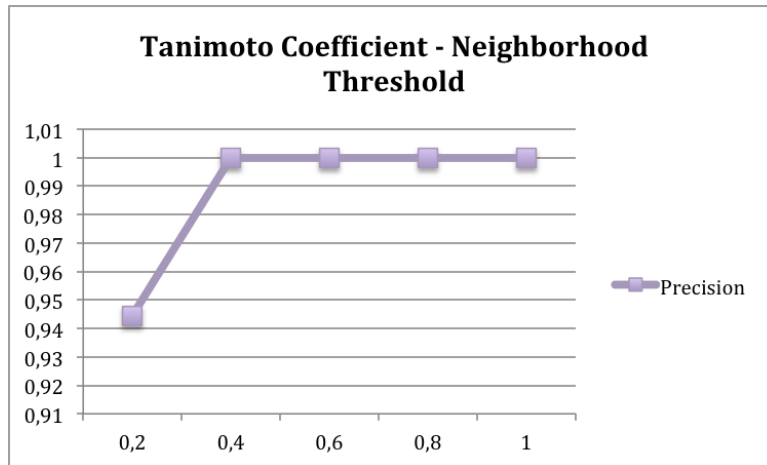


Figura 4.30: Evaluación de Precision - Coeficiente Tanimoto basado en Neighborhood Threshold.

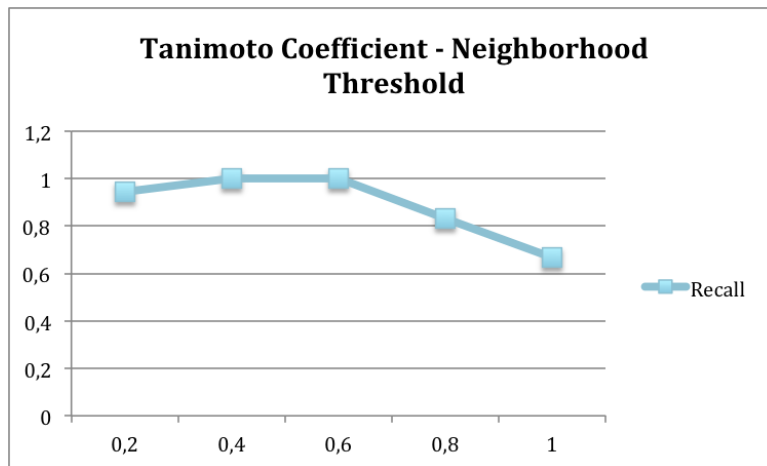


Figura 4.31: Evaluación de Recall - Coeficiente Tanimoto basado en Neighborhood Threshold.

Segun las figuras 4.30 y 4.31 los valores de Precision y Recall igual a 1 en valor de umbral entre 0,4 y 0,6.

Según los resultados presentados, para el caso de la técnica de vecindad basada en Nearest Neighborhood tanto para la Similitud Log Likelihood como para el Coeficiente Tanimoto el número de usuarios para los que el recomendador es capaz de recomendar, el número máximo de recomendaciones por usuario y el número de usuarios para cada número máximo de recomendaciones presentan el mismo comportamiento, es decir, el número de usuarios para los que el recomendador es capaz de recomendar alcanza su máximo valor (1230) a partir del valor de vecindad 400, el número máximo de recomendaciones por usuario es 13 y se presenta en el valor de vecindad igual al total de usuarios del dataset (1246) y el número de usuarios para este número máximo de recomendaciones es 46. Así mismo el máximo número de usuarios para un número de recomendaciones menor, el caso específico de 12 recomendaciones es 365 usuarios y se da en el valor de vecindad igual a 1000. Mientras que la

4.5. Fases del enfoque propuesto

evaluación de Precision y Recall demuestra que los mejores resultados se obtienen para los valores de vecindad entre 100 y 300 tanto para la Similitud Log Likelihood como para el Coeficiente Tanimoto.

La situación es un poco diferente para la técnica de vecindad basada en Neighborhood Threshold, para la Similitud Log Likelihood el número de usuarios para los que el recomendador es capaz de generar recomendaciones alcanza su máximo valor (1230) para los valores de Umbral comprendidos entre 0,2 y 0,8, mientras que para el Coeficiente Tanimoto el número de usuarios para los que el recomendador es capaz de recomendar alcanza su máximo valor (1230) para el valor de Umbral de 0,2. Por su parte, el número máximo de recomendaciones por usuario es 13 y el número de usuarios para éste es 46 y se presenta en los valores de Umbral comprendidos entre 0,2 y 0,4 para la Similitud Log Likelihood, mientras que para el Coeficiente de Tanimoto el número máximo de recomendaciones por usuario es 13 y el número de usuarios para éste es 46 y se presenta en el valor de Umbral igual 0,2. Así mismo el máximo número de usuarios para un número de recomendaciones menor, el caso específico de 12 recomendaciones es 302 usuarios y se da en el valor de Umbral igual a 0,8 para el caso de la Similitud Log Likelihood y 0,4 para el Coeficiente de Tanimoto. La diferencia también radica en la evaluación de Precision y Recall, ya que los valores máximos de Precision y Recall (igual a 1.0) se da en el valor de Umbral igual a 0,8 para la Similitud Log Likelihood y para el Coeficiente de Tanimoto se da en los valores de Umbral entre 0,4 y 0,6.

Debido a que a parte de la importancia de los valores de Precision y Recall, es importante garantizar las recomendaciones para la mayor cantidad de usuarios posibles y con un alto número de recomendaciones para cada uno, se elige el algoritmo basado en la métrica de Similitud Log Likelihood basada en Neighborhood Threshold con un valor de Umbral igual a 0,8.

4.5.4. Algoritmo de Maximización de consumo de recursos turísticos en un tiempo limitado

Esta fase toma el conjunto de recursos obtenidos de la fase de filtrado colaborativo para combinarlos y formar un conjunto de recursos que maximiza la cantidad de tiempo disponible por el usuario. Así mismo se tiene en cuenta el tiempo dedicado durante el desplazamiento.

Para el diseño de esta fase se tomó como base el problema de la “mochila”, el cual tiene como objetivo más general, seleccionar a un subconjunto del total del valor máximo, sujeto a la restricción de que su peso total no supere W .

En este sentido, la implementación del algoritmo de maximización empleó la solución al “Knapsack Problem”, en el cual se tienen algunos artículos con una ganancia p_j y un peso w_j que se empaquetan en una o varias mochilas de capacidad c . Vamos a suponer que todos los coeficientes p_j , w_j y c son números enteros positivos a pesar de los supuestos más débiles a veces pueden ser manejados en los problemas individuales.

El problema Knapsack es el problema de la elección de un subconjunto de los n elementos tales que la suma de los profits correspondiente se maximiza sin que la suma de los pesos exceda la capacidad c . Esto puede ser como formular el siguiente problema de maximización [212]:

$$\text{maximize } \sum_{j=1}^n p_j x_j \quad (6)$$

$$\text{subjecto } \sum_{j=1}^n w_j x_j \leq c, (7)$$

$$x_j \in \{0, 1\}, j = 1, \dots, n,$$

Donde x_j es una variable binaria igualada a 1 si el ítem j puede ser incluido en la mochila, y 0 si no.

Si se tiene una cantidad m_j de cada ítem tipo j , el Knapsack Problem acotado sería:

$$\text{maximize } \sum_{j=1}^n p_j x_j (8)$$

$$\text{subtecto } \sum_{j=1}^n w_j x_j \leq c, (9)$$

$$x_j \in \{0, 1, \dots, m_j\}, j = 1, \dots, n,$$

Este problema tiene numerosas aplicaciones en la teoría como en la práctica. Desde un punto de vista teórico, la estructura simple aboga por explotación de numerosas propiedades interesantes, que puede hacer que los problemas sean más fáciles de resolver. Los Knapsack Problems también surgen como subproblemas en varios algoritmos para más complejos problemas de optimización combinatoria, y estos algoritmos se beneficiarán de cualquier mejora en el campo de los Knapsack Problems.

Para este caso, este problema se empleó para maximizar el consumo de actividades en un determinado tiempo, así la variable peso w_j sería el tiempo de duración de cada actividad y c el tiempo disponible total con el que cuenta el usuario.

La solución a dicho problema fue implementada empleando el algoritmo de Programación dinámica, el cual es definido de la siguiente manera:

Supongamos que w_1, w_2, \dots, w_n, W son estrictamente enteros positivos. Se define $m[i, w]$ para ser el valor máximo que se puede alcanzar con el peso inferior o igual a w usando los elementos hasta i (primeros i artículos).

Podemos definir $m[i, w]$ recursivamente de la siguiente manera:

$$m[i, w] = m[i - 1, w] \text{ if } w_i > w$$

el nuevo ítem es mayor que el límite del peso actual

$$m[i, w] = \max(m[i - 1, w], m[i - 1, w - w_i] + v_i) \text{ if } w_i < w$$

La solución puede entonces ser encontrado mediante el cálculo de $m[n, W]$. Para hacer esto de manera eficiente, podemos utilizar una tabla para almacenar computaciones.

El siguiente es el algoritmo para la solución del problema por medio de la programación dinámica:

4.5. Fases del enfoque propuesto

```
//Values (stored in array v)
//Weights (stored in array w)
//Number of distinct items (n)
//Knapsack capacity (W)
for each j from 0 to W do
  m[0, j] := 0
end for
for each i from 1 to n do
  for j from 0 to W do
    if w[j] <= j then
      m[i, j] := m[i-1, j]
    end if
  end for
end for
```

Como resultado de esta fase, se obtienen las actividades que efectivamente si pueden ser realizadas en el tiempo disponible especificado por el usuario, teniendo en cuenta el tiempo que se requiere para realizar cada actividad y un tiempo de desplazamiento promedio entre cada una de ellas para garantizar maximizar el tiempo disponible con el que cuenta el turista.

4.5.5. Priorización por patrones de consumo

En esta fase, el usuario es el encargado de proporcionar los intereses los cuales corresponden a las categorías o patrones de consumo que le gustaría priorizar. Según el orden en el que sean escogidos se asignarán pesos de mayor a menor para emplearlos como postfiltro en el listado de recomendaciones y organizar la lista a ser desplegada al usuario.

En este sentido se relacionan los patrones de consumo con los tipos de actividades correspondientes, por ejemplo, Arte & Entretenimiento esta relacionado con actividades de tipo Arte, Aventura, Recreación, entre otros.

A continuación se describe el algoritmo utilizado para realizar el ordenamiento de las actividades a recomendar según los intereses o patrones de consumo indicados por el usuario:

```
for every interest i expressed by the user u
  get the type of activity ta concerned
for every type of activity ta
  for each recommended activity a
    type of activity ta compare to the type of each activity a
    if they match the types of activity added to the priority list
  return the priority list ranked by order of interest
```

Esta fase tiene como resultado organizar las actividades recomendadas a los turistas con base a sus intereses expresados, priorizándolas según el orden en los intereses marcados por el usuario en el sistema.

4.5.6. Optimización de Trayectoria

Esta última fase toma la salida de las actividades a recomendar priorizadas por los patrones de consumo elegidos por el usuario y emplea la solución al problema “Traveling Salesman Problem

(TSP).^o “Problema del Agente viajero”[213], el cual ha llamado mucho la atención de los matemáticos y científicos informáticos específicamente porque es muy fácil de describir y tan difícil de resolver. El problema sólo se puede establecer como: si un vendedor viajero desea visitar exactamente una vez cada uno de una lista de las ciudades m (donde el costo de viajar de la ciudad i a la ciudad j es c_{ij}) y luego regresar a la ciudad natal, ¿cuál es la ruta menos costosa que el viajante puede tomar?

El problema

Desde los días en que los vendedores viajaban de puerta en puerta vendiendo aspiradoras y enciclopedias, tenían que planificar sus rutas, de casa en casa o una ciudad a otra. Cuanto más corta la ruta, mejor. Encontrar la ruta más corta para un conjunto de lugares es un problema con una dificultad exponencial: encontrar el camino más corto para 20 ciudades tiene el doble de dificultad que para 10 ciudades.

Se garantizaría una búsqueda exhaustiva de todos los caminos posibles para encontrar la más corta, pero es computacionalmente intratable para todos pero los pequeños conjuntos de ubicaciones. Para los problemas más grandes, se necesitan técnicas de optimización para buscar inteligentemente el espacio de soluciones y encontrar soluciones casi óptimas.

Matemáticamente, el problema del agente viajero puede ser representado como un grafo, donde los lugares son los nodos y los pesos en los bordes representan las distancias entre los lugares. El objetivo es encontrar el camino más corto con la suma de los pesos. A continuación, un sencillo grafo de cuatro nodos y el ciclo más corto que visita todos los nodos como se observa en la figura 4.32:

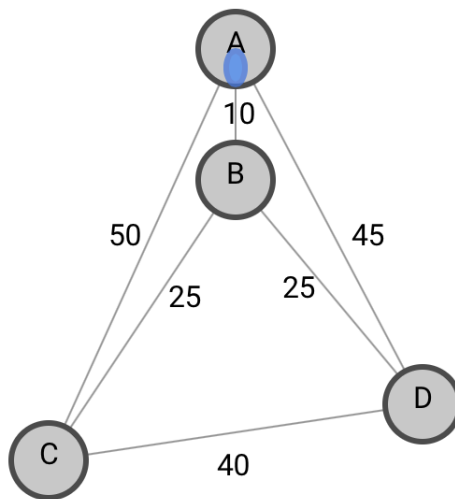


Figura 4.32: Grafo representativo del Problema del Agente Viajero.

El TSP tradicional es simétrico: la distancia del punto A al punto B es igual a la distancia desde el punto B al punto A. Gracias al API Directions de Google se puede resolver el problema TSP con ubicaciones del mundo real por medio de una Google Directions API key, proporcionando las ubicaciones en una URL y obteniendo la respuesta de vuelta como JSON.

4.5. Fases del enfoque propuesto

API Directions de Google

El API Directions de Google es un servicio que calcula direcciones entre ubicaciones mediante una petición http, por medio del cual se puede buscar direcciones para varios modos de transporte, incluyendo el tránsito, conducir, caminar o andar en bicicleta. Las direcciones pueden especificar orígenes, destinos y puntos sea como cadenas de texto (por ejemplo, “Chicago, IL.” “Darwin, NT, Australia”) o como coordenadas de latitud/longitud. El API Directions puede devolver direcciones de varias partes utilizando una serie de waypoints. Este servicio está diseñado en general para el cálculo de direcciones estáticas (conocidas con antelación) generalmente para la colocación de contenido de la aplicación en un mapa; y no está diseñado para responder en tiempo real a la entrada del usuario, por ejemplo.

Uso de Waypoints en rutas

En el cálculo de rutas utilizando el API Directions, también puede especificar Waypoints (puntos de referencia) para direcciones en el modo conducir, caminar o andar en bicicleta. Los Waypoints permiten calcular rutas por lugares adicionales, en cuyo caso la ruta devuelta pasa por los puntos dados.

Los puntos intermedios se especifican en el parámetro de waypoints y consisten en una o más direcciones o lugares separados por el carácter (“—”).

Por defecto, el servicio Directions calcula una ruta a través de los waypoints previstos en el orden dado. Opcionalmente, puede pasar el parámetro “optimize:true” como primer argumento dentro del parámetro de waypoints para permitir que el servicio Directions optimice la ruta proporcionada por la reordenación de los waypoints en un orden más eficiente. (Esta optimización es una aplicación del problema del Agente Viajero.)

Si se indica al servicio Directions para optimizar el orden de los waypoints, su orden será devuelto en el campo

```
waypoint_order
```

dentro del objeto de las rutas. El campo

```
waypoint_order
```

devuelve valores que son de base cero.

El siguiente ejemplo calcula una ruta de viaje por carretera desde Adelaide, Australia del Sur para cada una de las principales regiones vinícolas del sur de Australia utilizando la optimización de rutas.

```
https://maps.googleapis.com/maps/api/directions/json?
origin=Adelaide,SA&destination=Adelaide,SA&
waypoints=optimize:true|Barossa+Valley,SA|Clare,
SA|Connawarra,SA|McLaren+Vale,SA&key=API_KEY
```

La inspección de la ruta calculada indicará que se calcula la ruta con el siguiente waypoint orden:

```
waypoint_order: [ 1, 0, 2, 3 ]
```

En la presente propuesta, se ha empleado el API Directions de Google para optimizar la trayectoria a seguir por parte del turista en su visita a la ciudad, en ese sentido, ésta última fase retorna la ruta optimizada a seguir por parte del turista para ser desplegada en un mapa del aplicativo y de esta forma optimizar su tiempo dentro de su visita a la ciudad de destino.

4.6. Resumen

A partir del estudio realizado, se hace un planteamiento del enfoque de recomendación para motivar o incentivar el posible consumo de oferta turística por medio de la habilitación y maximización de posibles movimientos a realizar por parte de un turista optimizando su tiempo en un destino, presentando un esquema con las fases que lo componen, destacando que es un enfoque híbrido que combina a base de filtrado de conocimiento y filtrado colaborativo. Como primera fase se presentó la Inferencia de Perfiles Contextuales, la cual se encarga de predecir el perfil del turista por medio de técnicas de Aprendizaje Automático basadas en algunas variables contextuales del Modelo del Usuario. La segunda fase es el Sistema de Recomendaciones basado en Conocimiento, encargado de realizar el proceso de recomendación empleando un motor de inferencia, guiado por las reglas construidas que se razonan sobre la base de conocimiento (KB) representada por la ontología OWL y cuya salida de este proceso es un conjunto de recursos pertinentes, para este caso, esos recursos son actividades turísticas relacionados con un turista. La siguiente fase del enfoque propuesto es el uso de filtrado colaborativo para predecir las preferencias del usuario y generar recomendaciones usando el algoritmo basado en la métrica de Similitud Log Likelihood con la técnica de vecindad Neighborhood Threshold con un valor de Umbral igual a 0,8, la cual es una similitud basada en usuarios. La fase siguiente toma el conjunto de recursos obtenidos del filtrado colaborativo para combinarlos y formar un conjunto de recursos que maximiza la cantidad de tiempo disponible por el usuario tomando en cuenta el tiempo dedicado durante el desplazamiento, empleando un algoritmo de Programación dinámica que calcula de manera recursiva las actividades que efectivamente si pueden ser realizadas en el tiempo disponible especificado por el usuario. La fase a continuación toma el orden en el que sean escogidos los intereses (correspondientes a las categorías o patrones de consumo) para priorizar la lista de recomendaciones a ser desplegada al usuario. La última fase toma la salida de las actividades a recomendar priorizadas por los patrones de consumo elegidos por el usuario y emplea la solución al problema “Traveling Salesman Problem (TSP).” “Problema del Agente viajero”, para generar la ruta menos costosa en términos de tiempo que pase por todos los sitios donde se realizan las actividades recomendadas.

Capítulo 5

Implementación del Sistema Propuesto

Este capítulo presenta la arquitectura física y lógica del enfoque propuesto para el sistema de recomendaciones junto con los detalles de implementación de cada uno de los módulos que la integran, con lo cual se termina de dar cumplimiento al objetivo 2.

La arquitectura es un conjunto de principios de estructuración que permite que un sistema que se compone de un conjunto de sistemas más simples, cada uno con su propio contexto local que es independiente de, pero no es incompatible con el contexto del sistema más amplio en su conjunto. La arquitectura se creó para describir la estructura del sistema a construir y cómo esa estructura es compatible con los requerimientos del negocio y de nivel de servicio [162]. En esta sección se discutirán los componentes por los que está compuesto el sistema.

Antes de entrar en detalle de cada componente del sistema, se mostrará una visión general de la arquitectura física y lógica de la presente propuesta. La figura 5.1 muestra los principales componentes del sistema, mientras que la figura 5.2 muestra los principales elementos lógicos que componen el conjunto del sistema.

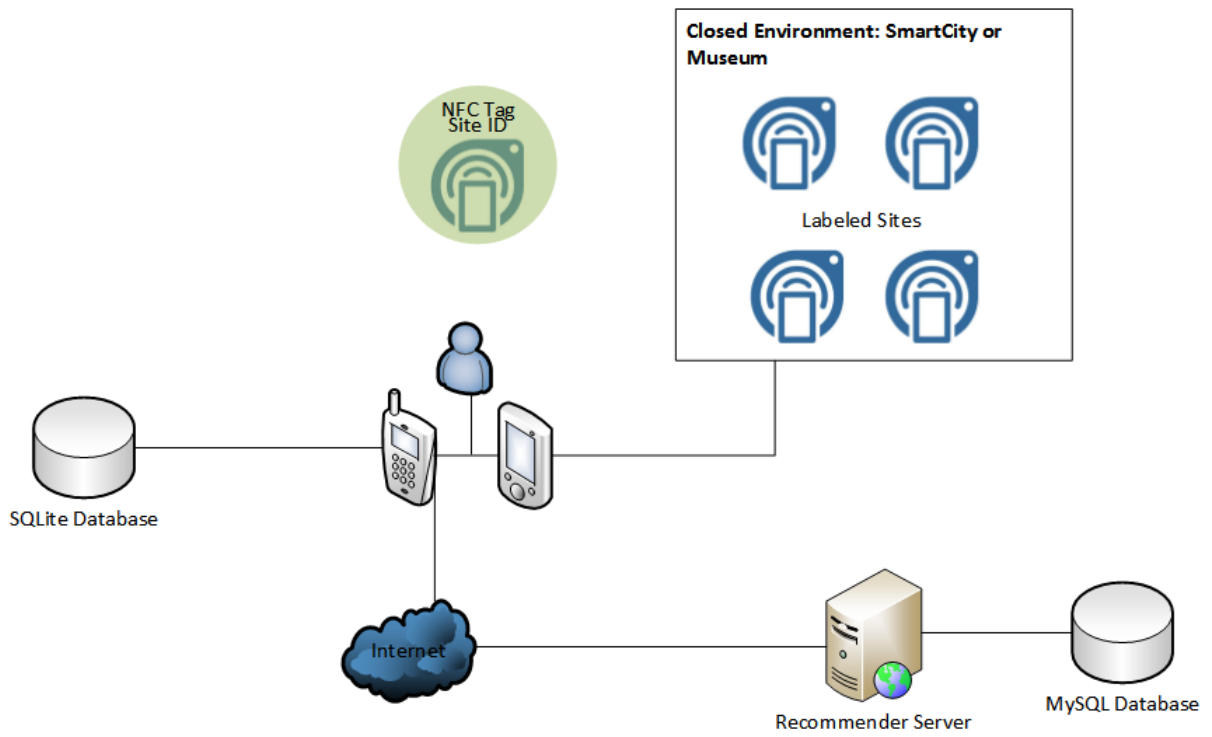


Figura 5.1: Arquitectura física del sistema.

La siguiente figura muestra la arquitectura lógica del sistema.

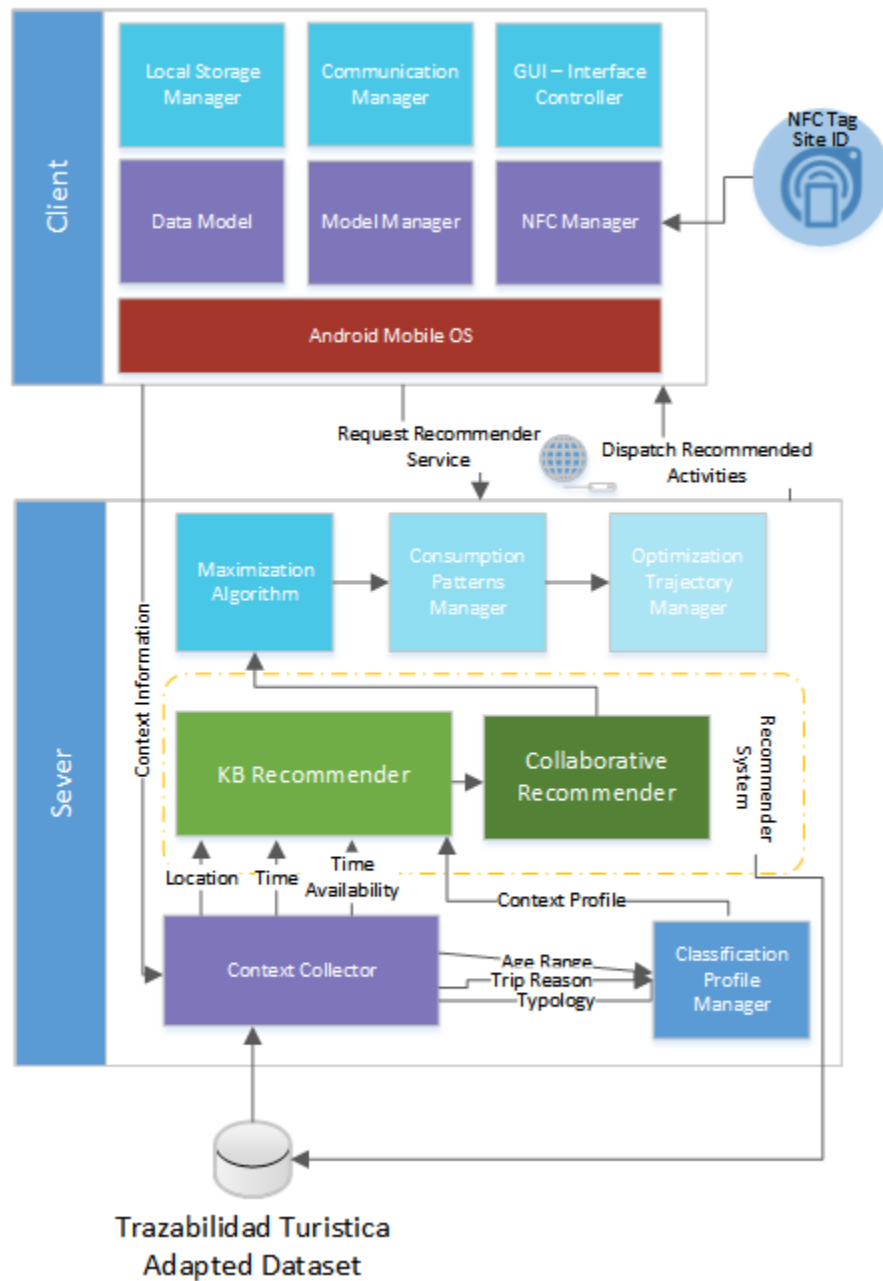


Figura 5.2: Arquitectura lógica del sistema.

Esta propuesta es una arquitectura distribuida que consta de tres módulos principales, tal como se representa en la Figura 5.2:

- **Smart Tags (Etiquetas Inteligentes):** es el objeto que representa una ubicación específica, una empresa, en este caso, un punto de interés turístico, o cualquier otro lugar en el escenario de aplicación, que está marcado con una etiqueta inteligente.
- **Cliente Móvil,** es una aplicación que se ejecuta en los dispositivos móviles de los usuarios. Gestiona la interacción con el usuario a través de una interfaz gráfica, mostrando las actividades recomendadas entregadas por el servidor, las cuales pueden ser visualizadas en una lista y en un mapa. También es el lector de etiquetas NFC, y la comunicación establecida se realiza en modo pasivo, porque el dispositivo móvil es el que tiene la fuente de alimentación y es el que genera la señal para el intercambio de datos. Otra funcionalidad es capturar la información contextual requerida por el módulo de inferencia de perfiles contextuales y el sistema de recomendaciones.
- **Servidor,** es el núcleo de la aplicación lógica del sistema. Incluye los servicios web de comunicación con el cliente móvil, así como también la implementación de cada una de las fases o módulos del enfoque propuesto.

5.1. Implementación de la solución

A continuación se describen los detalles de implementación de cada uno de los módulos que integran la arquitectura lógica presentada en la figura 5.2.

5.1.1. Smart Tags

El escenario de operación es un entorno abierto e inteligente gracias a las etiquetas inteligentes se propagan a través de él, donde hay objetos con información visual.

El usuario encontrará la etiqueta inteligente colocada en el punto de interés correspondiente, en la que, a primera vista, verá el nombre asignado a la misma (el lugar, la ubicación, punto de interés, etc.) y/o íconos, imágenes o cualquier otro texto o información visual del sitio turístico. El papel principal de las etiquetas inteligentes es ser un elemento habilitador del sistema propuesto, ya que permite localizar al usuario y que tenga acceso a las recomendaciones en un servicio de información turística, en relación con el lugar de interés donde se encuentra ubicada la etiqueta inteligente.

La Figura 5.3 muestra el número de formas en que se puede utilizar la tecnología NFC para diferentes propósitos. Hay básicamente tres categorías principales:

5.1. Implementación de la solución

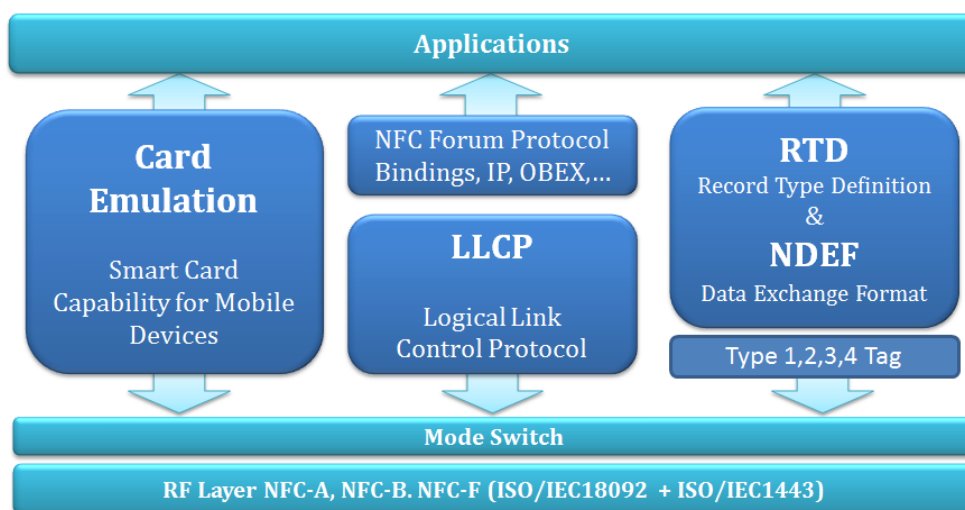


Figura 5.3: Usos de los dispositivos NFC. Adaptado de [214].

Para esta propuesta, cada punto de interés o ubicación tiene un identificador único (LUID) independientemente de la etiqueta inteligente a la que pertenece. Cada etiqueta inteligente (una etiqueta se relaciona con un punto de interés único) almacena un ID del Sitio etiquetado con el fin de permitir al usuario el acceso a información relacionada con:

- Nombre del punto de interés o ubicación.
- Información de contacto: como dirección y teléfono.

Así mismo, el sistema accede a las coordenadas físicas del punto de interés relacionado con el Tag, para realizar el proceso de recomendación.

5.1.2. Cliente Móvil

Este módulo es una aplicación móvil Android que implementa la interfaz gráfica de usuario que gestiona la interacción con los turistas y su implementación se basa en la plataforma Java Android. El diseño y desarrollo se basa en el paradigma Modelo-Vista-Controlador, este patrón arquitectónico tiene como objetivo aislar la lógica de negocio de la interfaz de usuario, por lo que es más fácil de modificar, ya sea la apariencia visual o las reglas de negocio subyacentes de una aplicación sin afectar al otro.

El modelo representa la lógica de negocio y es implementado por las siguientes clases: NFC Manager, Data Model, Model Manager, Communication Manager y de Local Storage Manager. El Data Model representa los datos manejados por la aplicación y contiene un subconjunto de entidades que cargan el modelo de información conceptual del sistema. El NFC Manager utiliza el API NFC de Android que permite las siguientes funciones: enviar y recibir datos NFC en forma de mensajes NDEF [215]. La lectura de datos NDEF de una etiqueta NFC se maneja con el sistema de etiquetas de envío, y se lleva a cabo con la creación de un Intent que encapsula e identifica la información, específicamente por medio del uso del $ACTION_TAG_DISCOVERED$, e interceptar el momento en el que es leída la etiqueta para llevar a cabo una acción específica.

La figura 5.4 muestra el formato de intercambio de datos de NFC, que es un formato de mensaje binario ligero y está diseñado para encapsular una o más cargas útiles de aplicación en el constructor del mensaje simple. Cada NDEF Message consta de uno o más NDEF Records. Cada NDEF Record puede transportar una carga útil de un tipo arbitrario y hasta $2^{31} - 1$ octetos de tamaño. Si la carga es mayor, los registros se pueden encadenar para apoyar datos más grandes.

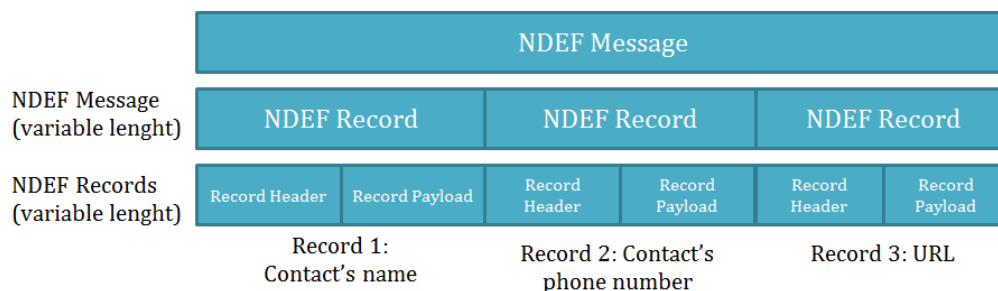


Figura 5.4: NDEF Message. Adaptado de [214].

El Model Manager gestiona la interacción entre Controller, la vista y los otros módulos que implementan el Modelo. Análogamente, cualquier invocación de los servicios proporcionados por el Communication Manager y el Local Storage Manager es gestionado por el Model Manager. El Communication Manager gestiona la comunicación basada en HTTP y servicios web REST-Ful entre el móvil y el servidor con el fin de enviar la información de la visita. El Local Storage Manager gestiona la base de datos de almacenamiento local a través del motor de base de datos SQLite.

La vista es responsable de desplegar las actividades recomendadas de forma gráfica (imagen del sitio en el que se realiza la actividad) al usuario por medio de una lista y un mapa con la ruta a seguir para su facilidad en la navegabilidad por la ciudad, por cada actividad recomendada se puede acceder a información mas detallada como una breve descripción e información de contacto. Así mismo es responsable de permitir la autenticación, registro y edición de información como: su tiempo disponible, rango para las recomendaciones, intereses, entre otros, las cuales son variables que toma el sistema de recomendaciones para operar.

El controlador gestiona las solicitudes del usuario, estas solicitudes pueden implicar cambios en el modelo de datos o solicitudes de comunicación con el servidor mediante el Communication Manager.

5.1.3. Servidor

El servidor contiene la aplicación lógica del enfoque del sistema de recomendaciones propuesto, como se mencionó anteriormente se compone de los siguientes módulos:

Context Collector

Es el componente responsable de recopilar toda la información de contexto necesaria. Esta información se captura a través del dispositivo móvil y su comunicación con el servidor por medio de servicios web, el cual se comunica con la base de datos del dataset Trazabilidad Turística adaptado.

5.1. Implementación de la solución

Classification Profile Manager

Este componente usa un algoritmo clasificador de Machine Learning para inferir contexto, en la presente propuesta se encarga de derivar automáticamente un perfil basado en el vector de contexto (perfiles contextuales). Como se mencionó en la sección 4.5.1, para entrenar a este algoritmo, se definen las etiquetas de clases (class labels) correspondientes a ese perfil, se definen un conjunto de reglas para etiquetar las clases y entrenar a los clasificadores para inferir perfiles contextuales, para lo cual, se hace una serie de supuestos relacionados con el comportamiento y estilo de vida de los turistas. Según los resultados obtenidos en la experimentación, en general se obtuvieron muy buenos resultados con la gran mayoría de algoritmos, sin embargo existieron dos que obtuvieron el mayor resultado de precisión y recall con las tres opciones de testing fue el algoritmo: WAODA de Bayes Classifier.

Este componente se implementa en java gracias al uso del API Weka usando el modelo WAODA aprendido y generado por Weka como predictor del Perfil Contextual de una nueva entrada del vector de contexto del vector usuario al que se le van a realizar las recomendaciones.

KB Recommender

Está formado por la base de conocimiento (KB) y el motor de Razonamiento. El primero es responsable de recibir la información y perfil de contexto del Context Collector y el Classification Profile Manager, almacenar las actividades a recomendar y crear la instancia de modelo de contexto. Antes de instanciar en la ontología las actividades a recomendar, el sistema realiza un filtro por ubicación y el rango en kilómetros especificado por el usuario, la cual tiene precargadas las reglas definidas en la sección 4.5.2. El segundo es el responsable de razonar sobre el lenguaje de Reglas de la Web Semántica (SWRL) para las reglas construidas utilizando un motor de inferencia, tales reglas guían el proceso de recomendación y la salida de este proceso es un conjunto de recursos pertinentes, en nuestro caso, esos recursos son actividades turísticas relacionadas con un usuario.

Para implementar este modulo se ha usado el API Jena [216] el cual se emplea como API de programación de ontologías y se encarga de cargar el modelo ontológico en memoria y acceder a un razonador externo para realizar inferencias acerca de las clases y las instancias en la ontología. La versión actual de Jena incluye un razonador OWL por defecto y dos pequeñas configuraciones rápidas. Cada una de las configuraciones está destinada a ser una adecuada ejecución de un subconjunto de semántica OWL pero ninguno de ellos es completa (en el sentido técnico). Para el razonamiento completo se utiliza un razonador OWL DL externo como Pellet, Racer o FaCT. El rendimiento (especialmente el uso de memoria) de la configuración del razonador más completa aún deja mucho que desear y el tema de los futuros trabajos - el tiempo lo permite. Hay dos maneras diferentes de utilizar Pellet en el API de Jena: o bien utilizando la interfaz directa de Pellet (muy recomendable); o el uso de la interfaz de Jena DIG (no recomendado). La interfaz directa Pellet es mucho más eficiente (por ejemplo, no tiene la sobrecarga de comunicación HTTP) y proporciona más inferencias (protocolo DIG tiene algunas limitaciones). Mediante la interfaz directa no es diferente que cualquier otro razonador Jena:

```
// ontology that will be used
String ont = "http://www.mindswap.org/2004/owl/mindswappers";
// create an empty ontology model using Pellet spec
OntModel model = ModelFactory.createOntologyModel(
    PelletReasonerFactory.THE_SPEC);
```

```
// read the file
model.read( ont );
// get the instances of a class
OntClass Person = model.getOntClass (
"http://xmlns.com/foaf/0.1/Person");
Iterator instances = Person.listInstances();
```

Las reglas se han incluido a la ontología por medio de la extensión SWRTabEditor, el cual es un subcomponente del SWRTab para Protege-OWL que soporta la edición de reglas SWRL. Se puede utilizar para crear reglas SWRL, editar reglas SWRL existentes, y leer y escribir reglas SWRL. El propio editor SWRL no realiza ninguna inferencia, para ello se utiliza el razonador Pellet, el cual se encarga de llevar a cabo la inferencia y hacer valer su nuevo conocimiento en el modelo OWL.

Collaborative Recommender

Este componente se encargará de predecir ratings usando el algoritmo basado en la métrica de Similitud Log Likelihood con la técnica de vecindad Neighborhood Threshold con un valor de Umbral igual a 0.8, la cual es una similitud basada en usuarios. Los items obtenidos en la anterior fase (consecuencia de aplicar un prefiltro) se emplean como entrada a este filtro colaborativo mediante Mahout. En términos de implementación, como el dataset no cuenta con ratings, se empleó la siguiente configuración:

Modelo: GenericBooleanPrefDataModel Recomendador: GenericBooleanPrefUserBasedRecommender Similitud basada en usuario: LoglikelihoodSimilarity Vecindad basada en Threshold: ThresholdUserNeighborhood con valor de 0,8

```
import java.util.ArrayList;
import java.util.List;
import java.util.logging.Logger;
import org.apache.mahout.cf.taste.common.TasteException;
import org.apache.mahout.cf.taste.impl.model.
GenericBooleanPrefDataModel;
import org.apache.mahout.cf.taste.impl.model.jdbc.
MySQLJDBCDataModel;
import org.apache.mahout.cf.taste.impl.neighborhood.
ThresholdUserNeighborhood;
import org.apache.mahout.cf.taste.impl.recommender.
GenericBooleanPrefUserBasedRecommender;
import org.apache.mahout.cf.taste.impl.similarity.
LogLikelihoodSimilarity;
import org.apache.mahout.cf.taste.model.JDBCDataModel;
import org.apache.mahout.cf.taste.neighborhood.UserNeighborhood;
import org.apache.mahout.cf.taste.recommender.RecommendedItem;
import org.apache.mahout.cf.taste.recommender.Recommender;
import org.apache.mahout.cf.taste.similarity.UserSimilarity;
import com.mysql.jdbc.jdbc2.optional.MysqlDataSource;

public class DBRecommender {
```

5.1. Implementación de la solución

```
public static Recommender recommender;
private static final Logger log = Logger
    .getLogger(DBRecommender.class.getName());

public static void initializeRecommender(){
try {

MysqlDataSource dataSource = new MysqlDataSource();
dataSource.setServerName("localhost");
dataSource.setPort(3306);
dataSource.setUser(Constants.dbUser);
dataSource.setDatabaseName(Constants.dbName);

JDBCDataModel model = new MySQLJDBCDataModel(
dataSource, "preferences", "id_user", "id_activity", "preference", "");
final GenericBooleanPrefDataModel boolModel =
new GenericBooleanPrefDataModel(
GenericBooleanPrefDataModel.toDataMap(model));
UserSimilarity similarity = new LogLikelihoodSimilarity(
    boolModel);
UserNeighborhood neighborhood = new ThresholdUserNeighborhood(
0.8, similarity, boolModel);

recommender = new GenericBooleanPrefUserBasedRecommender(
model, neighborhood, similarity);
} catch (TasteException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}

}

try {

List<RecommendedItem> recommendations = recommender.recommend(
userId,
20);

} catch (TasteException e) {
// TODO Auto-generated catch block
e.printStackTrace();
}

}
```

Como salida de este módulo, el sistema recomienda los ítems que los usuarios similares han evaluado de su preferencia pero todavía no han sido evaluados por este usuario, para este caso, se reco-

miendan las actividades turísticas basado en el dataset utilizado.

Maximization Algorithm y Consumption Patterns Manager

En éste módulo se implementan los algoritmos presentados en las secciones 4.5.4 y 4.5.5 por medio del lenguaje Java.

Optimization Trajectory Manager

Finalmente se gestiona la optimización de la trayectoria por medio del API Directions de Google, al cual se le pasan los waypoints o puntos intermedios por lo cuales debe pasar la ruta para obtener el orden en formato JSON, en el que se debería realizar para optimizar el tiempo del turista, teniendo en cuenta su desplazamiento.

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.List;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;

import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.protocol.BasicHttpContext;
import org.apache.http.protocol.HttpContext;
import org.server.actourec.entities.LatLng;
import org.w3c.dom.Document;
public String getMapsApiDirectionsUrl(List<LatLng> points,
LatLng location) {
    StringBuilder pointsStr = new StringBuilder();
    for (int i = 0; i < points.size(); i++) {
        pointsStr.append(points.get(i).getLatitude()+"",
            "+points.get(i).getLongitude());
        if (i<points.size()-1) {
            pointsStr.append("|");
        }
    }
    String waypoints = "waypoints=optimize:true|" + pointsStr.toString();
    String originDestination = "origin=" + location.getLatitude() +
        ", " + location.getLongitude() + "&destination="
        + location.getLatitude() + ", " + location.getLongitude();
```

5.1. Implementación de la solución

```
String sensor = "sensor=false";
String params = originDestination + "&"+waypoints + "&" + sensor;
String output = "json";
String url = "https://maps.googleapis.com/maps/api/directions/"
    + output + "?" + params;
return url;
}

public String getDirections(String mapsApiDirectionsUrl)
throws IOException {

String data = "";
InputStream iStream = null;
URLConnection urlConnection = null;
try {
    URL url = new URL(mapsApiDirectionsUrl);
    urlConnection = (URLConnection) url.openConnection();
    urlConnection.connect();
    iStream = urlConnection.getInputStream();
    BufferedReader br = new BufferedReader(
        new InputStreamReader(iStream));
    StringBuffer sb = new StringBuffer();
    String line = "";
    while ((line = br.readLine()) != null) {
        sb.append(line);
    }
    data = sb.toString();
    br.close();
} catch (Exception e) {
    System.out.println("Exception while reading url "
        +e.toString());
}
return data;
}
}
```

De forma adicional se encuentran los servicios web requeridos para gestionar la interacción con el usuario, en total son 6:

El primer servicio permite el Registro del usuario en el sistema, El segundo es el servicio de Autenticación de usuarios ya registrados, El tercero permite la edición de configuraciones del usuario, permitiendo editar sus intereses, tu tiempo disponible, rango de recomendaciones, entre otros, El cuarto servicio es el Registry Manager: gestiona el servicio de registro de las visitas a los puntos de interés en la base de datos, junto con la información de tiempo obtenida. El quinto servicio es el de las recomendaciones como tal, el cual ejecuta todo el enfoque propuesto para generar las actividades recomendadas al turista. El sexto y último servicio es el que le permite al turista enviar la calificación de la recomendación al sistema para motivos de evaluación de la presente propuesta.

De forma adicional, dentro de los pilotos iniciales se generó un Servicio de Generación Información a través de un mapa en una página web (Maps Generator Information Manager), el cual gestiona el servicio de despliegue de información de visitas a través de mapas en una página web. Para este propósito, se empleó el componente GMap de la librería Primefaces (un conjunto de componentes JSF) [217], que es construido con Google Maps API versión 3, integrado con el modelo de programación JSF y mejorado con capacidades Ajax. Esta API se utiliza para mostrar en un mapa la trazabilidad de la ruta seguida por un turista específico, así como también la ubicación de los sitios de interés, junto con el número de visitas que se han realizado en cada uno de los sitios.

El servidor ha sido diseñado y desarrollado como una aplicación J2EE (Java 2 Enterprise Edition). Así, la arquitectura del servidor consiste en una capa de presentación a través de mapas, una capa de lógica de aplicación de servicios y una capa de base de datos.

El intercambio de mensajes con las instancias de aplicaciones móviles se implementa de acuerdo a las especificaciones de servicios web, a través de la API de Java para servicios web XML-base (JAX-WS). Los servicios han sido implementados utilizando los servicios web RESTful JAX-RS. Los procedimientos se llevaron a cabo utilizando los servicios web de lenguaje Java con Jersey y compilados para la obtención de la aplicación de servidor.

La capa de base de datos gestiona el acceso a la información de base de datos. Los datos de la aplicación del sistema se conservan en una base de datos MySQL y se accede por medio del API Java Persistence.

Cada módulo ha sido diseñado y desarrollado con el fin de mantener lo más débilmente acoplados posible los componentes de la arquitectura. La interacción entre módulos se realiza a través de interfaces de servicios Web, reduciendo al mínimo la dependencia de las opciones tecnológicas y de ejecución.

5.2. Resumen

A lo largo de este capítulo se presenta una visión general de la arquitectura física y lógica de la propuesta, y finalmente se detallan los detalles de implementación de cada uno de los módulos que integran la arquitectura lógica presentada.

Capítulo 6

Experimentación y Resultados

Este capítulo presenta la evaluación del sistema de recomendaciones propuesto a través de dos aspectos: i) la calidad y utilidad de las recomendaciones y ii) el grado de aceptación del sistema por parte de un usuario final (turistas), ambas pruebas realizadas sobre un entorno de prueba y a través de un experimento on-line.

6.1. Metodología

El objetivo de esta sección es describir la metodología de la experimentación realizada en el presente proyecto de investigación, para lo cual se describirá la planeación del experimento, seguida por las métricas de evaluación empleadas y la ejecución de las pruebas y análisis de resultados.

6.1.1. Planeación del experimento

En esta sección se expone la planificación de la experimentación realizada, retomando el objetivo, definiendo la hipótesis, selección de sujetos y objetos experimentales.

Definición del Objetivo

El objetivo de la evaluación por medio de los experimentos realizados es “determinar la calidad” y nivel de aceptación por parte del turista del sistema presentado en esta propuesta por medio de la definición de cuatro escenarios simulados que buscan identificar a un turista en un perfil determinado para obtener recomendaciones por medio el enfoque propuesto.

Hipótesis

Esta experimentación busca la comprobación o negación de la Hipótesis relacionada a continuación: H1n (Hipótesis Nula) La utilización del sistema propuesto no incentiva el consumo de oferta turística por medio de la habilitación y maximización de posibles movimientos a realizar por parte de un turista optimizando su tiempo en un destino. H1a. (Hipótesis Alternativa) La utilización del sistema propuesto incentiva el consumo de oferta turística por medio de la habilitación y maximización de posibles movimientos a realizar por parte de un turista optimizando su tiempo en un destino. La comprobación o negación de la hipótesis enunciada anteriormente se realiza mediante la ejecución del método de evaluación Análisis de Características por Encuesta del grado de aceptación del sistema

por parte de un usuario final y la calidad y utilidad de las recomendaciones por medio de las métricas empleadas.

Selección de variables

Puesto que el objetivo del experimento es “determinar la calidad” y nivel de aceptación por parte del turista del sistema presentado, se utilizarán métricas de calidad descritas en la Sección 6.1.2 y el Análisis de Características por Encuesta. Así, las variables a utilizar son las siguientes:

Variables Independientes Perfil de Usuario, Intereses del Usuario, Hora del día, Día de la semana, Tiempo disponible del Usuario, Duración de las actividades recomendadas.

Variables Dependientes Actividades recomendadas por el sistema propuesto.

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 utilizar la aplicación en el dispositivo móvil con tecnología NFC de acuerdo a los instrucciones dadas, recibir las recomendaciones a medida que visitaran un sitio turístico (tocar etiqueta correspondiente al sitio), calificar dos o más recomendaciones de la lista y variar los parámetros o variables para observar el efecto del sistema. Así, se empleó el dataset ajustado a partir de los datos obtenidos del Proyecto de Trazabilidad Turística, cuyas características están descritas en la Sección 3.1.2. Para llevar a cabo la evaluación se han definido 4 escenarios simulados correspondientes a los 4 perfiles descritos en la sección 4.5.1:

Escenario de un turista Clásico: Suponga que es un turista que llega a la ciudad de Popayán de manera individual o en grupo, que tiene mas de 30 años y su motivo de viaje es principalmente el Turismo/Ocio.

Escenario de un turista que viaja en Familia: Suponga que es un turista que llega a la ciudad de Popayán acompañado de su familia o que su motivo de viaje principalmente es la Visita a Familiares.

Escenario de un turista intelectual: Suponga que es un turista que viaja a la ciudad de Popayán por motivo de Eventos o Trabajo/Negocios.

Escenario de un turista que prefiere la vida nocturna: Suponga que es un turista que llega a la ciudad de Popayán de manera individual o en grupo, que tiene entre 18 y 30 años y su motivo de viaje es principalmente el Turismo/Ocio.

A continuación se presenta la información general de los participantes de esta evaluación. La evaluación contó con 31 participantes entre 20 y 26 años de edad, de la ciudad de Popayán asistentes a la materia Fundamentos Tecnológicos para Empresas Turísticas del programa de Turismo de la Universidad del Cauca. Se citaron a los colaboradores en diferentes momentos a las instalaciones de la Universidad del Cauca a un recinto privado, ya que se conformaron 8 grupos de 4 personas (debido a que se tenían disponibles 4 teléfonos móviles con tecnología NFC), donde se realizó una exposición breve del proyecto, se explicó a dinámica del experimento y se realizó una pequeña demostración de

6.1. Metodología

funcionamiento de la aplicación y del procedimiento que debían realizar. Al final de cada sesión de experimentación (de 20 minutos aproximadamente cada una) las personas procedieron a rellenar la encuesta para medir el grado de aceptación del sistema.

Objetos Experimentales

Para cumplir con la finalidad de la experimentación se cuentan con las siguientes características: Número de usuarios: 31 Lugar de la muestra: Popayán Dimensiones contextuales analizadas: Ubicación, Perfil de Usuario, Hora del día, día de la semana y disponibilidad de tiempo del usuario. Edad de los participantes: 22 a 26 años

6.1.2. Métricas de Evaluación

La precisión de los sistemas de recomendaciones se ha evaluado en la literatura de investigación desde 1994 [218]. Muchas de las evaluaciones publicadas de sistemas de recomendación utilizan diferentes métricas, las cuales se clasifican ampliamente en tres clases: las métricas de precisión predictiva, métricas de precisión de clasificación, y métricas de precisión rango [219]. Para el contexto del presente proyecto es interesante la segunda clase de métricas, las cuales miden la frecuencia con la que un sistema de recomendación hace decisiones correctas o incorrectas acerca de si un artículo es bueno, estas métricas de clasificación son por lo tanto apropiadas para tareas tales como “encontrar buenos ítems”. Precision y Recall son las métricas más populares para la evaluación de los sistemas de recuperación de información. En 1968, Cleverdon les propone como las métricas clave [220], y se han celebrado desde entonces. Para la evaluación de los sistemas de recomendación, que han sido utilizados por [221], [222], [223] y [53].

Precision

La precisión es definida como la relación de los elementos relevantes seleccionados para número de elementos seleccionados. En otras palabras, la precisión es una medida de la eficiencia de la selección artículo pertinente definida a continuación:

$$Precision = \frac{Nrs}{Ns} \quad (10)$$

Donde Nrs es el conjunto de elementos seleccionados que son relevantes, y Ns el conjunto de elementos seleccionados o recomendados.

Precision representa la probabilidad de que un elemento seleccionado sea relevante. El número de elementos que se recomiendan en una lista puede ser muy alto, dependiendo del método de recomendación y el tamaño del conjunto de datos, y no es posible que un usuario pueda revisar y evaluarlos todos. Por esa razón, la métrica de evaluación tendrá en cuenta sólo los elementos principales, llamada recomendación Top-N [56], y por lo general se presenta en varios trabajos como Precision@n. Precision o Precision@n se utiliza para evaluar el sistema en el contexto de un solo usuario. Con el fin de obtener una única métrica que representa la precisión del método de recomendación sobre todo el conjunto de usuarios, se usa la métrica Mean Average Precision (MAP). MAP se obtiene calculando la media sobre la precisión promedio de la lista de recomendaciones de cada usuario, como:

$$MAP = \sum_{n=1}^N \frac{AveP(n)}{N} \quad (11)$$

En la ecuación, $AveP(n)$ es la precisión promedio para el usuario n , es decir, el promedio de los valores de precisión obtenida para el conjunto de top-N recomendaciones después de que se recupera cada recomendación relevante [224].

Recall

Recall se define como la relación de los elementos relevantes seleccionados al número total de elementos relevantes disponibles. En este sentido el Recall representa la probabilidad de que se selecciona un elemento relevante y se define a continuación:

$$Recall = \frac{Nrs}{Nr} \quad (12)$$

Donde Nrs es el conjunto de elementos seleccionados que son relevantes, y Nr el conjunto de elementos relevantes para el usuario.

Precision y Recall dependen de la separación de los artículos relevantes y no relevantes. La definición de “relevancia” y la forma correcta de calcular que ha sido una fuente importante de la argumentación en el campo de la recuperación de información. Además de los gustos de los usuarios que son diferentes, las escalas de calificación de usuario también pueden ser diferentes. Un usuario puede considerar una calificación de 3 en una escala de 5 puntos relevante, mientras que otro puede considerarlo irrelevante. Para el presente proyecto, se definió que un ítem es relevante cuando su calificación se encuentre entre 3 a 5. Aunque Precision y Recall son dos medidas diferentes, no pueden ser considerados separados una de otra. Se ha observado que Precision y Recall están inversamente relacionados. Generalmente, cuando el número de elementos recomendados aumenta, también lo hace el Recall, mientras que la precisión disminuye. Existen varios métodos que tienen en cuenta Precision y Recall y producen una sola métrica. Probablemente la más ampliamente utilizada es la medida-F, como se muestra en la siguiente ecuación, con $\alpha = 1$.

$$F = (1 + \alpha) \frac{Precision \times Recall}{\alpha \times Precision + Recall} \quad (13)$$

En el dominio de las recomendaciones, un score perfecto de Precision es 1,0 y significa que cada ítem que se recomienda en la lista es bueno (aunque no dice nada sobre si se sugirieron todas las buenas recomendaciones) mientras que un score perfecto de Recall es 1,0 y significa que todos los buenos ítems recomendados fueron sugeridos en la lista (aunque no dice nada de cómo muchas malas recomendaciones también estaban en la lista).

6.2. Plan de ejecución de pruebas

Utilidad

Si bien el ranking de referencia puntúa un ranking en su correlación con algún ranking “verdadero”, hay otros criterios para decidir sobre el orden de una lista de ítems. Una alternativa popular es asumir que la utilidad de una lista de recomendaciones es aditiva, dado por la suma de las utilidades de las recomendaciones individuales. La utilidad de cada recomendación es la utilidad del ítem recomendado descontada por un factor que depende de su posición en la lista de recomendaciones. Un ejemplo de una utilidad tal es la probabilidad de que un usuario observará una recomendación en la posición i en la lista. Se asume generalmente que los usuarios escanean las listas de recomendación desde el principio hasta el final, lo que hace que la utilidad de las recomendaciones se descuenta más fuertemente hacia el final de la lista. El descuento también puede interpretarse como la probabilidad de que un usuario podría observar una recomendación en una posición particular en la lista, con la utilidad de la recomendación ya que se observó que sólo depende de la opción recomendable. La métrica R-Score [225] supone que el valor de las recomendaciones declinan de manera exponencial por la lista clasificada para producir la puntuación siguiente para cada usuario u :

$$R_{a_i} = \sum_k \frac{\max(r_i(b_k) - d, 0)}{2^{\frac{k-1}{\alpha-1}}} \quad (14)$$

Donde (b_k) es el elemento en la posición k , $r_i(b_k)$ es el rating del usuario u para el ítem i , d es el valor predeterminado o el voto neutral y a es un número natural que denota el rango de un ítem de la lista que tiene un 50 % de probabilidad de ser visto, el cual controla la disminución exponencial del valor de las posiciones en la lista clasificada, generalmente se toma $\alpha = 2$. Las puntuaciones resultantes de cada usuario se agregan utilizando:

$$R_u = 100 \frac{\sum_{a_i} R_{a_i}}{\sum_{a_i} R_{a_i}^{max}} \quad (15)$$

Donde $R_{a_i}^{max}$ es la puntuación de la mejor clasificación posible para el usuario u (es decir, el sistema de recomendación clasificó los ítems exactamente como lo hizo el usuario).

6.2. Plan de ejecución de pruebas

Como se mencionó anteriormente, la evaluación busca comprobar la siguiente hipótesis, que es el objetivo del presente trabajo: H1a. (Hipótesis Alternativa) La utilización del sistema propuesto incentiva el consumo de oferta turística por medio de la habilitación y maximización de posibles movimientos a realizar por parte de un turista optimizando su tiempo en un destino. Para lo cual se realizó una experimentación de manera on-line, realizándose en primera instancia la evaluación de las métricas anteriormente definidas sobre los datos obtenidos en dicha experimentación, y seguidamente la conducción de la Prueba t sobre los datos recolectados por medio de la encuesta aplicada. En ambas evaluaciones se planteará una hipótesis que es la deseable a comprobar, de manera que si ambas hipótesis son validas entonces la hipótesis general H1a se comprueba y se cumple con el objetivo del presente trabajo.

6.2.1. Plan de Ejecución de Pruebas: Métricas

A través de esta evaluación se busca comprobar o refutar la siguiente hipótesis por medio del cálculo de las métricas anteriormente descritas: H01. (Hipótesis) La utilidad y Precisión promedio calculados de los datos obtenidos a través de la experimentación del sistema se encuentran por encima del 70 %.

Plan de Ejecución de Pruebas: Aceptación del Sistema

Por medio de esta evaluación se busca determinar el grado de aceptación del sistema por parte de usuarios finales, en este caso los turistas. A través de esta evaluación, el presente proyecto busca comprobar o refutar la siguiente hipótesis: H02: Los turistas demuestran una opinión positiva con respecto a la pertinencia de las actividades recomendadas de acuerdo al contexto del usuario por medio del sistema propuesto. Ya que el objetivo de la hipótesis es medir el grado de percepción sobre el objeto de evaluación, se emplea una encuesta con el fin de conocer aspectos relevantes sobre si el sistema es capaz de cumplir las expectativas de los turistas. De esta manera, la encuesta (ver Anexo B) está conformada por 15 preguntas que se componen de: Una descripción de la característica a evaluar, una estimación sobre su grado de importancia, una escala de evaluación y un espacio para observaciones; tal como se muestra en el siguiente Figura:

| | |
|--|---|
| I. Calidad de las recomendaciones | |
| Descripción: Esta característica mide la calidad percibida de la recomendación por parte del usuario. | |
| Pregunta 1: En general me gustan las actividades recomendadas por el sistema. | |
| Grado de importancia: Altamente deseable | |
| Escala de Evaluación: De acuerdo? | |
| Completamente | |
| En alto grado | |
| Parcialmente | |
| Muy poco | |
| Ninguno | X |
| Observación: | |

Figura 6.1: Ejemplo de pregunta de la Encuesta.

Esta encuesta se fundamenta en la evaluación expuesta por [226], cuyo enfoque está centrado en el usuario para la evaluación de los sistemas de recomendaciones. Esto quiere decir que se tienen en cuenta aspectos subjetivos relacionados con el comportamiento del usuario objetivo, las características personales y situacionales en la experiencia del usuario. Así mismo se observa que la percepción de la calidad y/o variedad de las recomendaciones son medidores importantes de la experiencia del usuario, teniendo en cuenta el esfuerzo o dificultad percibido, la eficacia percibida del sistema y la satisfacción de la elección. Para conducir la evaluación se emplea una Prueba t para una muestra, que permite verificar si en una muestra la variable de interés presenta una media determinada para evaluar la hipótesis nula de igualdad de la media con un valor determinado. De esta manera, la evaluación contó con 31 participantes teniendo en cuenta la teoría de pequeñas muestras, este número genera una distribución de muestreo adecuada, debido a que, cuando este número de muestras se aproximan a 31

6.3. Resultados de Evaluación

o tienden a número infinito, la dispersión correspondiente a la curva t disminuye aproximándose a la curva estándar, razón por la cual, se conoce como curva z a la curva t cuando N tiende a infinito. Es necesario especificar los conceptos de Alfa, p-Valor y Significancia Estadística.

Alfa: Representa el nivel de significancia, es decir, el nivel de alfa es la probabilidad de rechazar la hipótesis nula cuando la hipótesis nula es verdadera. Este concepto tiene un valor comprendido entre $[0,1]$. Además, el valor de alfa está asociado a un nivel de confianza de la prueba. A continuación se enumeran algunos niveles de confianza con sus valores correspondientes de alfa:

- Para obtener resultados con un nivel de confianza del 90 %, el valor de alfa es 0,1 ($1-0,90 = 0,10$).
- Para obtener resultados con un nivel de confianza del 95 %, el valor de alfa es $1-0,95 = 0,05$.
- Por lo general, el valor de alfa puede adoptar un rango diverso de valores. Por lo tanto, para resultados con un nivel de confianza C %, el valor de alfa es: $1 - C/100$.

p-Valor (p): El p-Valor es la probabilidad de obtener un resultado estadístico de la prueba, bajo el supuesto que la hipótesis nula es verdadera. A continuación, se expone la interpretación del p-Valor con base en un nivel de significancia del 10 %:

- $p < 0.01$: evidencia una muy fuerte suposición en contra de la hipótesis nula.
- $p = 0.01$: evidencia una fuerte suposición en contra de la hipótesis nula.
- $0.01 < p < 0.05$: evidencia una suposición moderada en contra de la hipótesis nula.
- $p > 0.1$: ninguna suposición en contra de la hipótesis nula.

Significancia Estadística: Para determinar si un resultado observado es estadísticamente significativo, se comparan los valores de alfa y p . Lo que conlleva a dos posibilidades:

- Si p es menor o igual a alfa: En este caso se rechaza la hipótesis nula. Cuando esto ocurre se dice que el resultado es estadísticamente significativo.
- Si p es mayor que alfa. En este caso, no se rechaza la hipótesis nula. Cuando esto ocurre se dice que el resultado no es estadísticamente significativo.

Considerando un nivel de significancia de 0.05 (correspondiente al de mayor aceptación en la academia y la industria), en la sección 6.3 se presentan los resultados conseguidos en cada evaluación descrita en la sección anterior.

6.3. Resultados de Evaluación

A continuación se presentan las pruebas que fueron realizadas utilizando el entorno de experimentación compuesto por las herramientas empleadas en el enfoque propuesto en el Capítulo 5. En los experimentos planteados fueron necesarias todos los componentes contenidos en la arquitectura planteada; sin embargo, en este capítulo se enfatiza en las pruebas para realizar la Evaluación de las Recomendaciones.

6.3.1. Resultados de Evaluación de Métricas

Se realizó la evaluación on-line por medio de los datos recolectados en el experimento midiendo cada una de las métricas presentadas en la sección anterior.

Resultados de Evaluación de Utilidad

Se realizó el cálculo de la utilidad para cada lista de recomendaciones presentada al usuario y luego se obtuvo la utilidad promedio de cada usuario. Finalmente para calcular la utilidad total del sistema se obtuvo la utilidad media para todos los usuarios. A continuación se presenta el cálculo de la utilidad para el usuario con id 111 a manera de ejemplo, donde $d = 2$ y $\alpha = 3$:

| k | b_k | $r_i(b_k)$ | $r_i(b_k) - d$ |
|-----|-------|------------|----------------|
| 1 | 55 | 4 | 2 |
| 2 | 56 | 0 | 0 |
| 3 | 58 | 3 | 1 |
| 4 | 57 | 0 | 0 |
| 5 | 109 | 3 | 1 |
| 6 | 61 | 0 | 0 |
| 7 | 115 | 0 | 0 |

Tabla 6.1: Cálculo de Utilidad R_{α_i}

| k | b_k | $r_i(b_k)$ | $r_i(b_k) - d$ |
|-----|-------|------------|----------------|
| 1 | 55 | 4 | 2 |
| 2 | 58 | 3 | 1 |
| 3 | 109 | 3 | 1 |
| 4 | 55 | 0 | 0 |
| 5 | 57 | 0 | 0 |
| 6 | 61 | 0 | 0 |
| 7 | 115 | 0 | 0 |

Tabla 6.2: Cálculo de Utilidad $R_{\alpha_i}^{max}$

El cálculo de R_{α_i} es igual 2,75 y $R_{\alpha_i}^{max} = 3,20$, por tanto el $R - score$ da como resultado 85,74706699, observando que para este usuario la lista de recomendaciones fue muy útil según las calificaciones asignadas a los ítems. El anterior procedimiento se llevó a cabo para todos los usuarios (en el Anexo B se presenta el cálculo de la utilidad para cada usuario de manera detallada) y a continuación se presentan los resultados resumidos de la utilidad calculada para cada usuario:

6.3. Resultados de Evaluación

| User ID | R_u |
|---------|-------|
| 111 | 78,22 |
| 106 | 70,71 |
| 321 | 61,40 |
| 113 | 100 |
| 107 | 86,88 |
| 323 | 90,67 |
| 130 | 81,53 |
| 152 | 100 |
| 109 | 100 |
| 331 | 91,66 |
| 151 | 100 |
| 306 | 90,95 |
| 112 | 88,58 |
| 335 | 69,17 |
| 192 | 83,76 |
| 330 | 83,71 |
| 116 | 100 |
| 339 | 88,36 |
| 327 | 100 |
| 337 | 75,22 |
| 376 | 85,35 |
| 342 | 81,11 |
| 364 | 83,08 |
| 120 | 95,01 |
| 512 | 94,14 |

Tabla 6.3: Cálculo de Utilidad para todos los usuarios.

Finalmente para el cálculo de la utilidad del sistema se obtiene la media para todos los usuarios, cuyo valor indica que las recomendaciones resultan muy útiles para los usuarios del sistema, puesto que está por encima del valor de referencia planteado (≥ 70):

$$R = 87,18$$

Resultados de Evaluación de Precision y Recall

Se realizó el cálculo de la precisión teniendo en cuenta los elementos Top-N de cada lista de recomendaciones presentada al usuario, ya que en el experimento se les solicitó a los usuarios que calificaran por lo menos dos ítems de la lista, se tomaron como Top-N los primeros 3 y como ítems relevantes los que fueron calificados de 3 a 5 y no relevantes calificados de 1 a 2. Por lo general se puede esperar una compensación entre Precision y Recall, pero a medida que las listas de recomendación son mas largas normalmente se suele mejorar el Recall, sin embargo también es probable que reduzca la precisión. Por lo tanto, se pueden calcular las curvas comparando Precision y Recall, o la verdadera

tasa de true positive y false positive. Las curvas del primer tipo son conocidos simplemente como curvas de Precision-Recall, mientras que las de este último tipo se conocen como curvas ROC. Mientras que ambas curvas miden la proporción de ítems preferidos que en realidad se recomiendan, las curvas Precision-Recall enfatizan la proporción de ítems recomendados que son preferidos mientras que las curvas ROC enfatizan la proporción de ítems que no son preferidos que terminan siendo recomendados. Se opta por usar la curva Precisión-Recall con base a las propiedades del dominio y el objetivo del sistema; para este caso, la medida de precisión describe qué proporción de las recomendaciones son realmente convenientes para el usuario, si las recomendaciones inapropiadas representan una fracción pequeña o grande de las actividades turísticas inadecuadas que podrían haber sido recomendadas (es decir, la tasa de falsos positivos) puede no ser tan relevante, por tanto a pesar que se calcula Precision y Recall, la medida que interesa evaluar para el contexto de la presente experimentación es Precision. A continuación se presenta el cálculo de Precision y Recall para el usuario con id 323 a manera de ejemplo, donde se toman los primeros 3 ítems de cada lista de recomendación y como ítems relevantes los que fueron calificados de 3 a 5 y no relevantes calificados de 1 a 2: Fueron presentadas dos listas de recomendaciones en dos instantes de tiempo diferentes definidas como el conjunto de elementos seleccionados o recomendados:

| Posición en la lista | Item ID | Rating |
|----------------------|---------|--------|
| 1 | 117 | 4 |
| 2 | 118 | 3 |
| 3 | 127 | 5 |
| 4 | 129 | 5 |
| 5 | 120 | 0 |
| 6 | 122 | 0 |
| 7 | 128 | 0 |

Tabla 6.4: Lista 1 (Hora: 9:43:14) de Recomendaciones para el usuario con ID 323

| Posición en la lista | Item ID | Rating |
|----------------------|---------|--------|
| 1 | 64 | 4 |
| 2 | 127 | 3 |
| 3 | 129 | 5 |
| 4 | 120 | 5 |
| 5 | 121 | 0 |
| 6 | 122 | 0 |
| 7 | 128 | 0 |

Tabla 6.5: Lista 2 (Hora: 9:48:42) de Recomendaciones para el usuario con ID 323

Como se puede observar, el conjunto de elementos relevantes para el usuario está definido como $Nr = 117, 118, 64, 127, 129$. Para el caso de ambas listas, el conjunto de elementos seleccionados que son relevantes para cada momento son $Nrs1 = 117, 118$ y $Nrs2 = 64, 127, 129$. Como se mencionó anteriormente se tomaron las 3 primeras recomendaciones para el cálculo de Precision@n, por lo que el conjunto de elementos recomendados Top-N está definido así: $Ns1 = 117, 118, 127$ y $Ns2 = 64, 127, 129$, para la Lista 1 y 2 respectivamente. A continuación el cálculo de Precision y Recall para cada una de las listas quedaría de la siguiente manera:

6.3. Resultados de Evaluación

$$Precision@n1 = \frac{Nrs1}{Ns1} = \frac{\{117, 118\}}{\{117, 118, 127\}} = \frac{2}{3} = 0,66$$

$$Precision@n2 = \frac{Nrs2}{Ns2} = \frac{\{64, 127, 129\}}{\{64, 127, 129\}} = \frac{3}{3} = 1$$

$$Recall1 = \frac{Nrs1}{Nr} = \frac{\{117, 118\}}{\{117, 118, 64, 127, 129\}} = \frac{2}{5} = 0,4$$

$$Recall2 = \frac{Nrs2}{Nr} = \frac{\{64, 127, 129\}}{\{117, 118, 64, 127, 129\}} = \frac{3}{5} = 0,6$$

Calculando la Precision@n promedio y el Recall Promedio para el usuario en consideración se obtendría:

$$AvePrecision@n_{323} = 0,833$$

$$AveRecall_{323} = 0,5$$

Para calcular Precision y Recall para múltiples usuarios se empleó el enfoque de calcular Precision y Recall (o tasa positiva verdadera y tasa de falsos positivos) en cada lista de recomendaciones de longitud N para cada usuario, y luego calcular el promedio de Precision y Recall en cada N [223]. Por lo tanto, este enfoque es útil en el escenario de recomendar algunos buenos ítems”, donde una decisión importante es la longitud de la lista de recomendación. El anterior procedimiento se llevo a cabo (en el Anexo B se presenta el cálculo de Precision y Recall para cada lista de recomendaciones presentada al usuario) y los resultados obtenidos para cada usuario se presentan a continuación:

| idUser | AvePrecision | AveRecall |
|--------|--------------|-----------|
| 111 | 0,44 | 0,33 |
| 106 | 0,66 | 1 |
| 321 | 0,55 | 0,33 |
| 113 | 0,5 | 0,5 |
| 107 | 0,66 | 0,2 |
| 323 | 0,833 | 0,5 |
| 130 | 0,8 | 0,5 |
| 152 | 0,833 | 0,5 |
| 109 | 0,833 | 0,5 |
| 331 | 0,77 | 0,38 |
| 151 | 0,833 | 0,5 |
| 306 | 0,66 | 0,5 |
| 112 | 0,833 | 0,5 |
| 335 | 0,833 | 0,5 |
| 192 | 0,55 | 0,5 |
| 330 | 0,5 | 0,5 |
| 116 | 1 | 1 |
| 339 | 0,65 | 0,33 |
| 327 | 0,5 | 0,5 |
| 337 | 0,833 | 0,5 |
| 376 | 0,833 | 0,5 |
| 342 | 0,66 | 0,2 |
| 364 | 0,6 | 0,2 |
| 120 | 0,75 | 0,25 |
| 512 | 0,77 | 0,09 |

Tabla 6.6: Cálculo de Precision y Recall Promedio para todos los usuarios.

Finalmente para el cálculo de Precision y Recall del sistema se obtiene la media para todos los usuarios:

$$MAP_{sistema} = \sum_{n=1}^N \frac{AvePrecision@n(n)}{N} = \frac{1}{25}(17,681) = 0,707$$

$$Recall_{sistema} = \sum_{n=1}^N \frac{AveRecall(n)}{N} = 0,745$$

Con los datos obtenidos se procedió a graficar la curva Precision-Recall:

6.3. Resultados de Evaluación

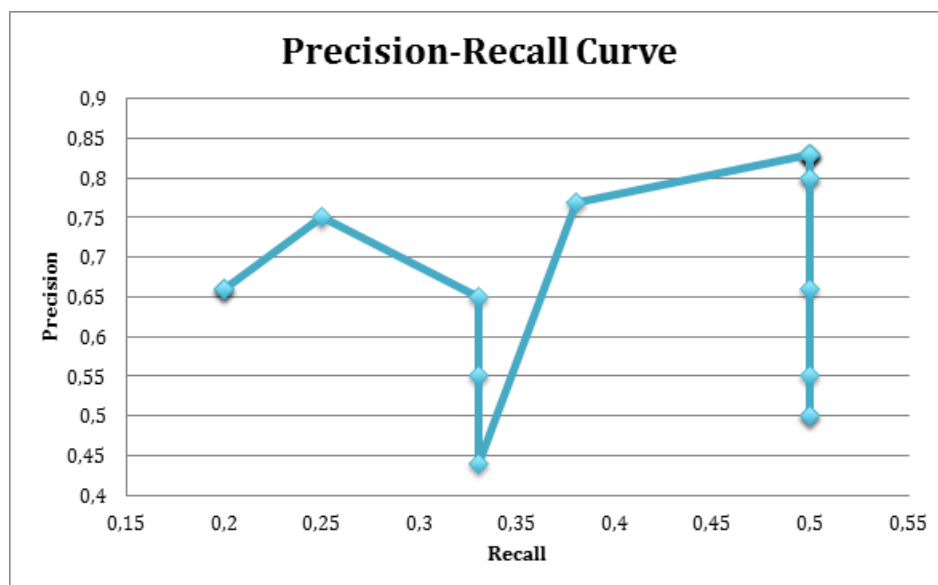


Figura 6.2: Ejemplo de pregunta de la Encuesta.

Por lo general, Precision y Recall están inversamente relacionados, es decir, a medida que aumenta Precision, Recall disminuye y viceversa. En la tarea de “recomendar algunos buenos ítems” es probable que prefiramos un sistema con una alta tasa de Precision, mientras que en la tarea de “recomendar todos los buenos ítems”, una tasa de Recall más alta es más importante que la de Precision. Para el caso del presente sistema, como se mencionó anteriormente, por el contexto turístico en el que se encuentra enmarcado es más importante la métrica de Precision que la de Recall, ya que es importante dar a conocer al usuario la oferta turística de la ciudad por medio del sistema de recomendaciones. Con relación a los resultados presentados, se observa que los valores obtenidos para las métricas de Utilidad (87,18 %) y Precision (70,7 %) fueron mayores a un 70 % lo que indica que se acepta la hipótesis objetivo(H01).

6.3.2. Resultados de Evaluación de aceptación del sistema

Se procede a transformar una escala de calificación de 1 a 5 a una propuesta con valores legibles para las personas, como se presenta en la tabla 6.7:

| Escala | Calificación |
|---------------|--------------|
| Completamente | 5 |
| En Alto Grado | 4 |
| Parcialmente | 3 |
| Muy Poco | 2 |
| Ninguno | 1 |

Tabla 6.7: Escala de Calificación transformada.

Se planteó dicha escala de tal forma que la escala “Parcialmente” exprese un grado de aceptación neutra (3), las escalas “Completamente” y “En alto Grado” reflejan un alto grado de aceptación (>3) y por último, las opciones “Muy poco” y “Ninguno” reflejan un grado de aceptación bajo (<3) sobre el sistema. Como la evaluación contó con 31 participantes, se realizó una revisión previa a la cuantificación de resultados, eliminando 3 de las encuestas porque se dedujo que no entendieron el objetivo del experimento. Así, a continuación se visualizan las frecuencias de los resultados obtenidos en cada pregunta de la encuesta:

| Escala | Frecuencia | Porcentaje (%) | Porcentaje Acumulado (%) |
|---------------|------------|----------------|--------------------------|
| Completamente | 9 | 32,14 | 32,14 |
| En Alto Grado | 4 | 14,28 | 46,42 |
| Parcialmente | 14 | 50 | 96,42 |
| Muy Poco | 1 | 3,57 | 100 |
| Ninguno | 0 | 0 | 100 |
| Total | 28 | 100.0 | |

Tabla 6.8: Tabla de Frecuencia Pregunta 1.

| Escala | Frecuencia | Porcentaje (%) | Porcentaje Acumulado (%) |
|---------------|------------|----------------|--------------------------|
| Completamente | 3 | 10,71 | 10,71 |
| En Alto Grado | 9 | 32,14 | 42,85 |
| Parcialmente | 14 | 50 | 92,85 |
| Muy Poco | 2 | 7,14 | 100 |
| Ninguno | 0 | 0 | 100 |
| Total | 28 | 100.0 | |

Tabla 6.9: Tabla de Frecuencia Pregunta 2.

6.3. Resultados de Evaluación

| Escala | Frecuencia | Porcentaje (%) | Porcentaje Acumulado (%) |
|---------------|------------|----------------|--------------------------|
| Completamente | 2 | 7,14 | 7,14 |
| En Alto Grado | 11 | 39,28 | 46,42 |
| Parcialmente | 14 | 50 | 96,42 |
| Muy Poco | 1 | 3,57 | 100 |
| Ninguno | 0 | 0 | 100 |
| Total | 28 | 100.0 | |

Tabla 6.10: Tabla de Frecuencia Pregunta 3.

| Escala | Frecuencia | Porcentaje (%) | Porcentaje Acumulado (%) |
|---------------|------------|----------------|--------------------------|
| Completamente | 7 | 25 | 25 |
| En Alto Grado | 10 | 35,71 | 60,71 |
| Parcialmente | 9 | 32,14 | 92,85 |
| Muy Poco | 2 | 7,14 | 100 |
| Ninguno | 0 | 0 | 100 |
| Total | 28 | 100.0 | |

Tabla 6.11: Tabla de Frecuencia Pregunta 4.

| Escala | Frecuencia | Porcentaje (%) | Porcentaje Acumulado (%) |
|---------------|------------|----------------|--------------------------|
| Completamente | 9 | 32,14 | 32,14 |
| En Alto Grado | 8 | 28,57 | 60,71 |
| Parcialmente | 9 | 32,14 | 92,85 |
| Muy Poco | 1 | 3,57 | 96,42 |
| Ninguno | 1 | 3,57 | 100 |
| Total | 28 | 100.0 | |

Tabla 6.12: Tabla de Frecuencia Pregunta 5.

| Escala | Frecuencia | Porcentaje (%) | Porcentaje Acumulado (%) |
|---------------|------------|----------------|--------------------------|
| Completamente | 15 | 53,57 | 53,57 |
| En Alto Grado | 10 | 35,71 | 89,28 |
| Parcialmente | 2 | 7,14 | 96,42 |
| Muy Poco | 1 | 3,57 | 100 |
| Ninguno | 0 | 0 | 100 |
| Total | 28 | 100.0 | |

Tabla 6.13: Tabla de Frecuencia Pregunta 6.

| Escala | Frecuencia | Porcentaje (%) | Porcentaje Acumulado (%) |
|---------------|------------|----------------|--------------------------|
| Completamente | 15 | 53,57 | 53,57 |
| En Alto Grado | 10 | 35,71 | 89,28 |
| Parcialmente | 2 | 7,14 | 96,42 |
| Muy Poco | 1 | 3,57 | 100 |
| Ninguno | 0 | 0 | 100 |
| Total | 28 | 100.0 | |

Tabla 6.14: Tabla de Frecuencia Pregunta 7.

| Escala | Frecuencia | Porcentaje (%) | Porcentaje Acumulado (%) |
|---------------|------------|----------------|--------------------------|
| Completamente | 9 | 32,14 | 32,14 |
| En Alto Grado | 16 | 57,14 | 89,28 |
| Parcialmente | 3 | 10,71 | 100 |
| Muy Poco | 0 | 0 | 100 |
| Ninguno | 0 | 0 | 100 |
| Total | 28 | 100.0 | |

Tabla 6.15: Tabla de Frecuencia Pregunta 8.

| Escala | Frecuencia | Porcentaje (%) | Porcentaje Acumulado (%) |
|---------------|------------|----------------|--------------------------|
| Completamente | 6 | 21,42 | 21,42 |
| En Alto Grado | 10 | 35,71 | 57,14 |
| Parcialmente | 10 | 35,71 | 92,85 |
| Muy Poco | 2 | 7,14 | 100 |
| Ninguno | 0 | 0 | 100 |
| Total | 28 | 100.0 | |

Tabla 6.16: Tabla de Frecuencia Pregunta 9.

| Escala | Frecuencia | Porcentaje (%) | Porcentaje Acumulado (%) |
|---------------|------------|----------------|--------------------------|
| Completamente | 7 | 25 | 25 |
| En Alto Grado | 12 | 42,65 | 67,85 |
| Parcialmente | 7 | 25 | 92,85 |
| Muy Poco | 2 | 7,14 | 100 |
| Ninguno | 0 | 0 | 100 |
| Total | 28 | 100.0 | |

Tabla 6.17: Tabla de Frecuencia Pregunta 10.

6.3. Resultados de Evaluación

| Escala | Frecuencia | Porcentaje (%) | Porcentaje Acumulado (%) |
|---------------|------------|----------------|--------------------------|
| Completamente | 15 | 53,57 | 53,57 |
| En Alto Grado | 9 | 32,14 | 85,71 |
| Parcialmente | 3 | 10,71 | 96,42 |
| Muy Poco | 1 | 3,57 | 100 |
| Ninguno | 0 | 0 | 100 |
| Total | 28 | 100.0 | |

Tabla 6.18: Tabla de Frecuencia Pregunta 11.

| Escala | Frecuencia | Porcentaje (%) | Porcentaje Acumulado (%) |
|---------------|------------|----------------|--------------------------|
| Completamente | 7 | 25 | 25 |
| En Alto Grado | 11 | 39,28 | 64,28 |
| Parcialmente | 10 | 35,71 | 100 |
| Muy Poco | 0 | 0 | 100 |
| Ninguno | 0 | 0 | 100 |
| Total | 28 | 100.0 | |

Tabla 6.19: Tabla de Frecuencia Pregunta 12.

| Escala | Frecuencia | Porcentaje (%) | Porcentaje Acumulado (%) |
|---------------|------------|----------------|--------------------------|
| Completamente | 12 | 42,85 | 42,85 |
| En Alto Grado | 8 | 28,57 | 71,42 |
| Parcialmente | 6 | 21,42 | 92,85 |
| Muy Poco | 2 | 7,14 | 100 |
| Ninguno | 0 | 0 | 100 |
| Total | 28 | 100.0 | |

Tabla 6.20: Tabla de Frecuencia Pregunta 13.

| Escala | Frecuencia | Porcentaje (%) | Porcentaje Acumulado (%) |
|---------------|------------|----------------|--------------------------|
| Completamente | 19 | 67,85 | 67,85 |
| En Alto Grado | 7 | 25 | 92,85 |
| Parcialmente | 2 | 7,14 | 100 |
| Muy Poco | 0 | 0 | 100 |
| Ninguno | 0 | 0 | 100 |
| Total | 28 | 100.0 | |

Tabla 6.21: Tabla de Frecuencia Pregunta 14.

Con el fin de describir apropiadamente las características relevantes de esta evaluación, se procede a realizar el análisis estadístico descriptivo a partir de la equivalencia numérica de las escalas de las frecuencias obtenidas en las calificaciones de cada pregunta a continuación:

| Escala | Frecuencia | Porcentaje (%) | Porcentaje Acumulado (%) |
|---------------|------------|----------------|--------------------------|
| Completamente | 18 | 64,28 | 64,28 |
| En Alto Grado | 9 | 32,14 | 96,42 |
| Parcialmente | 1 | 3,57 | 100 |
| Muy Poco | 0 | 0 | 100 |
| Ninguno | 0 | 0 | 100 |
| Total | 28 | 100.0 | |

Tabla 6.22: Tabla de Frecuencia Pregunta 15.

| Escala | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | P11 | P12 | P13 | P14 | P15 |
|-----------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| Completamente | 9 | 3 | 2 | 7 | 9 | 15 | 15 | 9 | 6 | 7 | 15 | 7 | 12 | 19 | 18 |
| En Alto Grado | 4 | 9 | 11 | 10 | 8 | 10 | 10 | 16 | 10 | 12 | 9 | 11 | 8 | 7 | 9 |
| Parcialmente | 14 | 14 | 14 | 9 | 9 | 2 | 2 | 3 | 10 | 7 | 3 | 10 | 6 | 2 | 1 |
| Muy Poco | 1 | 2 | 1 | 2 | 1 | 1 | 1 | 0 | 2 | 2 | 1 | 0 | 2 | 0 | 0 |
| Ninguno | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Media | 3,75 | 3,46 | 3,50 | 3,78 | 3,82 | 4,39 | 4,39 | 4,21 | 3,71 | 3,85 | 4,35 | 3,89 | 4,07 | 4,60 | 4,60 |
| Moda | 3 | 3 | 3 | 4 | 5 | 4 | 5 | 4 | 4 | 4 | 5 | 4 | 5 | 5 | 5 |
| Desviac. Típica | 0,96 | 0,79 | 0,69 | 0,91 | 1,05 | 0,78 | 0,78 | 0,62 | 0,89 | 0,89 | 0,82 | 0,78 | 0,97 | 0,62 | 0,56 |

Tabla 6.23: Tabla de Frecuencias.

De acuerdo al valor de la moda indica que la mayoría de usuarios tienen una opinión favorable “En Alto Grado”, es decir, consideran adecuado el sistema para incentivar el turismo. De esta manera, teniendo en cuenta que tres (3) corresponde al punto neutro, se formulan las siguientes hipótesis estadísticas:

- H03n (Hipótesis nula): El grado de aceptación promedio del sistema es igual a 3.
- H03a (Hipótesis alternativa): El grado de aceptación promedio del sistema es mayor a 3.

Para validar la hipótesis nula, se obtuvieron los resultados que se observan en la tabla 6.23 y considerando un nivel de significancia de 0.05, se consiguieron los resultados que se observan en la tabla 6.3.

Con relación a los resultados presentados en la anterior tabla, se observa que el nivel de significancia estadística es menor a 0,005, lo que indica que el grado de aceptación promedio sobre las preguntas evaluadas por turistas es estadísticamente significativo y se rechaza la hipótesis nula H3n, y por tanto se comprueba la hipótesis objetivo de esta evaluación H01. Así, para una perspectiva más general de los resultados la Figura 6.4 representa una distribución de las evaluaciones y la Figura 6.5 grafica el porcentaje de frecuencia de las calificaciones:

6.3. Resultados de Evaluación

| Pregunta | Valor de prueba = 3 | | | | | |
|----------|---------------------|----|----------|----------------------|--|----------|
| | t | gl | Sig. (p) | Diferencia de medias | 95% de intervalo de confianza de la diferencia | |
| | | | | | Inferior | Superior |
| P1 | 4,104 | 27 | 0,000300 | 0,75 | 0,38 | 1,12 |
| P2 | 3,099 | 27 | 0,004400 | 0,46 | 0,16 | 0,77 |
| P3 | 3,813 | 27 | 0,000700 | 0,50 | 0,23 | 0,77 |
| P4 | 4,533 | 27 | 0,000100 | 0,79 | 0,43 | 1,14 |
| P5 | 4,116 | 27 | 0,000300 | 0,82 | 0,41 | 1,23 |
| P6 | 9,378 | 27 | 0,000000 | 1,39 | 1,09 | 1,70 |
| P7 | 9,378 | 27 | 0,000000 | 1,39 | 1,09 | 1,70 |
| P8 | 10,200 | 27 | 0,000000 | 1,21 | 0,97 | 1,46 |
| P9 | 4,215 | 27 | 0,000250 | 0,71 | 0,37 | 1,06 |
| P10 | 5,091 | 27 | 0,000024 | 0,86 | 0,51 | 1,20 |
| P11 | 8,692 | 27 | 0,000000 | 1,36 | 1,04 | 1,68 |
| P12 | 6,011 | 27 | 0,000002 | 0,89 | 0,59 | 1,20 |
| P13 | 5,793 | 27 | 0,000004 | 1,07 | 0,69 | 1,45 |
| P14 | 13,523 | 27 | 0,000000 | 1,61 | 1,36 | 1,85 |
| P15 | 15,000 | 27 | 0,000000 | 1,61 | 1,39 | 1,83 |

T Valor de prueba T.
gl: Grado de libertad.

Figura 6.3: Prueba T - Evaluación del sistema.

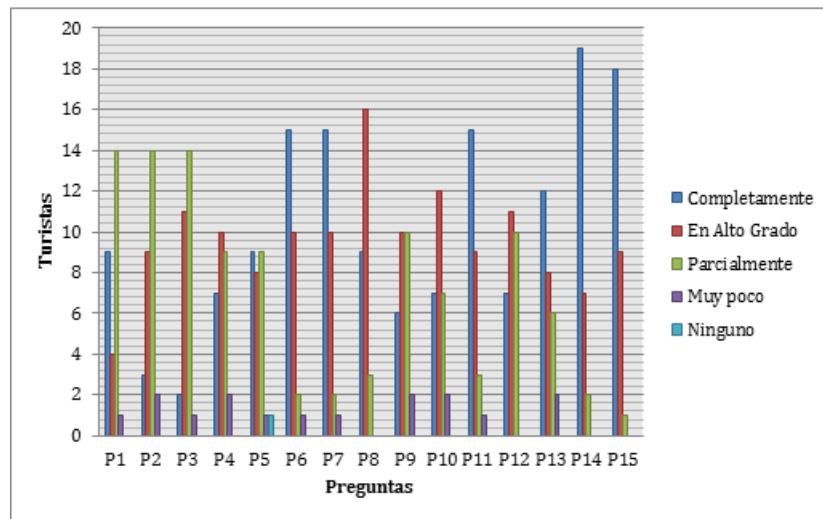


Figura 6.4: Distribución de las calificaciones.

A partir de la figura anterior, se obtienen las siguientes observaciones:

- Las preguntas con opinión parcialmente favorable, se presentaron en la en la P1,P2 y P3 indicando que la calidad de las recomendaciones es buena, ya que la tarea del sistema de recomendaciones era “recomendar algunos buenos ítems” y los turistas observaron que varios de ellos fueron relevantes para ellos.
- En términos generales, los resultados obtenidos en las preguntas P4, P5, P7,P8 y P9 fueron los deseables ya que la utilidad o eficacia del sistema percibida por el usuario tuvo gran nivel de aceptación comprobando que el sistema permite ampliar la perspectiva turística de los usuarios incentivando su consumo y optimizando su tiempo en el destino.
- Con respecto a las preguntas restantes, los resultados son igualmente deseables, indicando que el sistema es intuitivo, conveniente, algo variado y que su aplicabilidad a futuro e implementación en un entorno real podría ser exitoso.

Respecto a la totalidad de las calificaciones conseguidas, en la Figura 6.5: el 34 % corresponde a la opción “En Alto Grado”, el 37 % a “Completamente” y el 25 % a “Parcialmente”. Por lo tanto, los resultados positivos en la evaluación constituyen un 71 %, las opiniones neutras constituyen un 25 %, y para un 4 % de presentaron opiniones negativas. Teniendo en cuenta los resultados conseguidos, se puede considerar que el sistema tiene una alto grado de aceptación sobre la mayoría de usuarios.

6.4. Resumen

Este capítulo abordó una descripción detallada de la evaluación llevada a cabo para el sistema propuesto, realizándose dos tipos de evaluaciones por medio de un experimento de manera on-line: La primera consistió en obtener los resultados de las métricas de evaluación definidas sobre los datos obtenidos con el experimento; y la segunda evaluación consistió en una apreciación del sistema, que permitió determinar el grado de aceptación del mismo. Los resultados obtenidos en la experimentación

6.4. Resumen

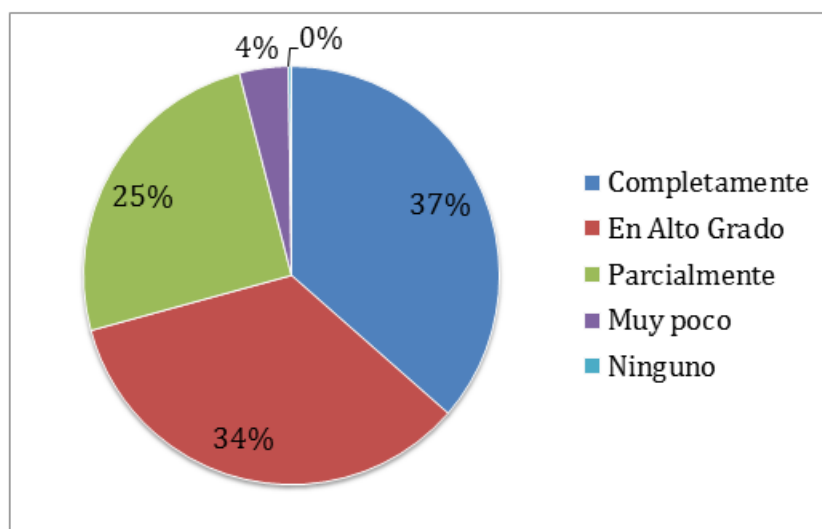


Figura 6.5: Porcentaje de frecuencia.

presentada, permiten deducir por un lado que el sistema propuesto alcanzó un alto nivel de utilidad y Precision por medio de la comprobación de la hipótesis H01 con los resultados de la evaluación de las métricas empleadas; y por otro lado los usuarios finales manifestaron una opinión positiva con respecto a la pertinencia de las actividades recomendadas de acuerdo al contexto del usuario por medio del sistema propuesto, como se pudo comprobar en la evaluación de la aceptación del sistema al validar la hipótesis H02. Finalmente gracias a la comprobación de lo anterior se puede afirmar que la utilización del sistema propuesto incentiva el consumo de oferta turística por medio de la habilitación y maximización de posibles movimientos a realizar por parte de un turista optimizando su tiempo en un destino definida como la hipótesis deseable a comprobar por medio de este capítulo (H1a), relacionada directamente con el cumplimiento del objetivo general del presente trabajo. A su vez, el contenido del presente capítulo van alineado con el cumplimiento del Objetivo Específico 3, Sección 1.3.

Capítulo 7

Conclusiones y Trabajo Futuro

Actualmente, el turismo es un dominio desafiante para el desarrollo de servicios basados en localización que proporcionan información personalizada y relevante (recomendaciones) a los turistas, sin embargo el usuario deseará priorizarla de acuerdo a patrones de consumo turístico y optimizar su tiempo de permanencia en el destino. 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 una conceptualización de un modelo de contexto, analizando los datos de contexto existentes y su posible modelado para su uso en el sistema, específicamente considerando la interacción del usuario con un entorno turístico ubicuo basado en la tecnología NFC, el enfoque de recomendación para motivar o incentivar el posible consumo de oferta turística por medio de la habilitación y maximización de posibles movimientos a realizar por parte de un turista optimizando su tiempo en un destino. Así, este capítulo describe inicialmente las principales conclusiones obtenidas, sus restricciones y finalmente se propone algunos trabajos futuros.

7.1. Conclusiones

- El diseño de un sistema de recomendación significa tomar decisiones que se pueden clasificar en los siguientes dominios: algoritmos (métodos de recomendación a utilizar), arquitectura (implementación del sistema), perfil de usuario (modelo de usuario), las cuales se ven limitadas en gran parte por el entorno del recomendador, el cual se puede estudiar a través de tres dimensiones: Usuarios (qué son los usuarios, cuáles son sus objetivos), Datos (cuáles son las características de los datos en que se basan las recomendaciones) y Aplicación (cuál es la aplicación general del recomendador en el sistema).
- Las aplicaciones context-awareness apoyan de manera inteligente las tareas del usuario, actuando autónomamente en nombre de los usuarios. Para esto es necesario definir un modelo de la información de contexto, el cual generalmente varía según el tipo de información que puede representar, el cual permita capturar diversas características de la información de contexto, incluyendo la variedad de tipos, dependencias, la calidad, y las historias de contexto. Además, se requieren métodos de procesamiento y gestión de contexto para apoyar el descubrimiento y la reutilización de la información contextual.
- Para el modelo contextual conceptualizado en la presente propuesta se analizaron y modelaron los datos de contexto existentes para su uso en el sistema, y gracias al enfoque centrado en el

turista, hace que se deba considerar la interacción del mismo con un entorno turístico ubicuo basado en la tecnología NFC, incluyendo información de ubicación detectada por medio de NFC, para el caso del modelo de usuario se consideró la información de tiempo, intereses del usuario, el perfil contextual, su tiempo disponible, tipología, el motivo de viaje y el rango de edad y características propias de las actividades turísticas a ser recomendadas: su tipo, horario de disponibilidad y duración de las mismas. Por el lado del modelo de datos se consideró el tipo, calidad/cantidad de los datos, ratings y sus dimensiones, entre otros.

- De acuerdo al dominio turístico es indispensable contar con el perfil de usuario para incorporarlo en el sistema de recomendaciones, y de acuerdo al conjunto de datos empleado se hizo necesario inferirlo a partir de ciertas variables con las que se contaba, empleando técnicas de Machine Learning para este proceso. Luego de su inferencia el perfil turístico es empleado como una de las entradas del sistema de recomendaciones propuesto. Así mismo por este motivo, es muy importante conocer las opiniones de otros turistas para generar las recomendaciones, y es por esto que se hizo uso de la técnica de Filtrado Colaborativo, la cual predice las preferencias del usuario y genera las recomendaciones empleando la métrica de similitud basada en usuarios.
- La información contextual en este trabajo se modela utilizando el enfoque basado en ontologías, ya que es muy adecuado para el intercambio de conocimiento. Dado que proporciona una especificación formal de la semántica de datos de contexto y esto es especialmente importante en entornos móviles y ubicuos donde diferentes entidades heterogéneas y distribuidas deben interactuar para el intercambio de información de contexto de los usuarios, es por esto que se usa un enfoque de recomendación basado en conocimiento, para lo cual se construyen reglas que se razonan sobre la base de conocimiento (KB) representada por la ontología.
- Para optimizar el tiempo del turista en el destino, se implementó un Algoritmo de Maximización de consumo de recursos turísticos en un tiempo limitado, el cual emplea un algoritmo de Programación dinámica que calcula de manera recursiva las actividades que efectivamente si pueden ser realizadas en el tiempo disponible especificado por el usuario. Adicionalmente se realizó una Optimización de Trayectoria, generando la ruta menos costosa en términos de tiempo que pase por todos los sitios donde se realizan las actividades recomendadas.
- Para priorizar las recomendaciones de acuerdo a los intereses proporcionados por el usuario que corresponden a las categorías o patrones de consumo en el orden en el que sean escogidos se asignaron unos pesos de mayor a menor para emplearlos como postfiltro en el listado de recomendaciones y organizar la lista a ser desplegada al usuario.
- De acuerdo al estudio realizado, el enfoque de recomendación presentado está compuesto 6 fases: el primera instancia la fase denominada “Inferencia de Perfiles Contextuales”, a continuación la fase llamada “Filtro basado en Conocimiento”, seguidamente la fase de “Filtrado colaborativo”, la fase siguiente “Algoritmo de Maximización de consumo de recursos turísticos en un tiempo limitado”, luego la fase “Priorización por patrones de consumo” una última fase denominada “Optimización de Trayectoria”, destacando que es un enfoque híbrido que combina a base de filtrado de conocimiento y filtrado colaborativo.
- Los resultados obtenidos en la experimentación realizada, permiten deducir por un lado que el sistema propuesto alcanzó un alto nivel de Utilidad y Precision por medio de la comprobación de la hipótesis “La utilidad y Precision promedio calculados de los datos obtenidos a través

7.2. Trabajos Futuros

de la experimentación del sistema se encuentran por encima del 70% con los resultados de la evaluación de las métricas empleadas; y por otro lado los usuarios finales manifestaron una opinión positiva con respecto a la pertinencia de las actividades recomendadas de acuerdo al contexto del usuario por medio del sistema propuesto, como se pudo comprobar en la evaluación de la aceptación del sistema al validar la hipótesis “Los turistas demuestran una opinión positiva con respecto a la pertinencia de las actividades recomendadas de acuerdo al contexto del usuario por medio del sistema propuesto”.

- Finalmente gracias a la comprobación de lo anterior se puede afirmar que la utilización del sistema propuesto incentiva el consumo de oferta turística por medio de la habilitación y maximización de posibles movimientos a realizar por parte de un turista optimizando su tiempo en un destino definida como la hipótesis deseable a comprobar “La utilización del sistema propuesto incentiva el consumo de oferta turística por medio de la habilitación y maximización de posibles movimientos a realizar por parte de un turista optimizando su tiempo en un destino”, relacionada directamente con el cumplimiento del objetivo general del presente trabajo.

7.2. Trabajos Futuros

- Mejorar el dataset con el fin de enriquecerlo con más ítems a recomendar y más ratings de usuario por actividad recomendada para evitar los problemas de dispersión de datos.
- Incorporar información de redes sociales en los mecanismos de recomendación propuestos para enriquecer aún más el sistema.
- Extender el soporte de la propuesta a otras plataformas móviles. Se plantea la necesidad de migrar la aplicación a otras plataformas móviles, como pueden ser iOS, Windows Phone, entre otros. Esto con el fin de observar los efectos que puede generar el tipo del mercado o plataforma, sobre el sistema propuesto.
- Evaluar la herramienta propuesta en un entorno real, etiquetando la ciudad con el fin de entrenar el sistema de recomendación a través de un conjunto considerable de usuarios finales.
- Investigar más algoritmos de Optimización de Trayectoria y Maximización de recursos en un tiempo limitado para explorar y evaluar sus características para su inclusión en el sistema.

Bibliografía

- [1] N. Leiper, *Tourism Management*. RMIT Press, 1995. 1, 3
- [2] U. C. Programme, “International tourism exceeds expectations with arrivals up by 52 million in 2013.” 1
- [3] L. Carrara and G. Orsi, “A new perspective in pervasive advertising,” tech. rep., Oxford University, 2011. 1
- [4] V. Coskun, K. Ok, and B. Ozdenizci, *Near Field Communication: From Theory to Practice*. Wiley Publishing, 1st ed., 2012. 1, 2
- [5] D. Remédios, L. Sousa, M. Barata, and A. L. Osório, “NFC technologies in mobile phones and emerging applications,” in *Information Technology For Balanced Manufacturing Systems - IFIP TC5, WG 5.5 Seventh International Conference on Information Technology for Balanced Automation Systems in Manufacturing and Services, Niagara Falls, Ontario, Canada, September 4-6, 2006*, pp. 425–434, 2006. 1, 2
- [6] T. K. Çapın, K. Pulli, and T. Akenine-Möller, “The state of the art in mobile graphics research,” *IEEE Computer Graphics and Applications*, vol. 28, no. 4, pp. 74–84, 2008. 1, 3
- [7] K. Verbert, N. Manouselis, X. Ochoa, M. Wolpers, H. Drachsler, I. Bosnic, and E. Duval, “Context-aware recommender systems for learning: A survey and future challenges,” *IEEE Trans. Learn. Technol.*, vol. 5, pp. 318–335, Jan. 2012. 1, 8
- [8] D. J. E. T. M. Z. A. Felfernig, S. Gordea, “A short survey of recommendation technologies in travel and tourism,” *ÖGAI Journal 25/2*, 2007. 1, 3
- [9] F. Tohid Ardahaey, “Economic impacts of tourism industry,” *International Journal of Business and Management*, vol. 6, Aug. 2011. 2, 3
- [10] A. Honkanen and P. Mustonen, “Tourism consumption revisited - an empirical study of finnish consumers,” *Research on Finnish Society Vol. 1 (2007)*, pp. 43–58, 2007. 2, 3
- [11] M. Zanker, M. Fuchs, W. Höpken, M. Tuta, and N. Müller, “Evaluating recommender systems in tourism - a case study from austria,” in *ENTER* (P. O’Connor, W. Höpken, and U. Gretzel, eds.), pp. 24–34, Springer, 2008. 2
- [12] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin, “Incorporating contextual information in recommender systems using a multidimensional approach,” *ACM Trans. Inf. Syst.*, vol. 23, pp. 103–145, Jan. 2005. 2

- [13] T. B. M. Plöbñig, “Designing recommender systems for tourism,” *Proc. ENTER*, 2004. 2, 28
- [14] K. Ducatel, M. Bogdanowicz, F. Scapolo, J. Leijten, and J.-C. Burgelman, “Scenarios for ambient intelligence in 2010,” IST Programme Advisory Group (ISTAG), 2001. 2
- [15] C. Ramos, J. C. Augusto, and D. Shapiro, “Ambient intelligence; the next step for artificial intelligence,” *IEEE Intelligent Systems*, vol. 23, no. 2, pp. 15–18, 2008. 2
- [16] F. Telefónica, “Smart cities: un primer paso hacia la internet de las cosas,” tech. rep., Telefónica, 2011. 2
- [17] S. Loke, *Context-Aware Pervasive Systems: Architectures for a New Breed of Applications*. Auerbach Publications, 1 ed., Dec. 2006. 2
- [18] H. Ailisto, L. Pohjanheimo, P. Väikkynen, E. Strömmer, T. Tuomisto, and I. Korhonen, “Bridging the physical and virtual worlds by local connectivity-based physical selection,” *Personal and Ubiquitous Computing*, vol. 10, no. 6, pp. 333–344, 2006. 2, 4
- [19] L. Desforges, “Tourism consumption and the imagination of money,” *Trans. Inst. Br. Geogr.*, vol. 266, no. 3, pp. 353–364, 2001. 3
- [20] G. Adomavicius and A. Tuzhilin, “Context-aware recommender systems,” in *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys ’08, (New York, NY, USA), pp. 335–336, ACM, 2008. 3, 7, 51
- [21] N. Davies, K. Cheverst, K. Mitchell, and A. Efrat, “Using and determining location in a context-sensitive tour guide,” *IEEE Computer*, vol. 34, no. 8, pp. 35–41, 2001. 4, 29, 30
- [22] “Ubicicero: A location-aware, multi-device museum guide,” *Interacting with Computers*, vol. 21, no. 4, pp. 288–303, 2009. 4, 30
- [23] A. K. Dey, G. D. Abowd, and D. Salber, “A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications,” *Hum.-Comput. Interact.*, vol. 16, pp. 97–166, Dec. 2001. 7, 9
- [24] B. Schilit, N. Adams, and R. Want, “Context-aware computing applications,” in *Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications*, WMCSA ’94, (Washington, DC, USA), pp. 85–90, IEEE Computer Society, 1994. 7
- [25] G. Chen and D. Kotz, “A survey of context-aware mobile computing research,” tech. rep., Hanover, NH, USA, 2000. 8
- [26] A. Schmidt, M. Beigl, and H.-W. Gellersen, “There is more to context than location,” *Computers and Graphics*, vol. 23, no. 6, pp. 893–901, 1999. 8
- [27] R. Beale and P. Lonsdale, *Mobile Context Aware Systems: The Intelligence to Support Tasks and Effectively Utilise Resources*. 2004. 8
- [28] L. Li, Y. Zheng, H. Ogata, and Y. Yano, “A framework of ubiquitous learning environment,” in *Computer and Information Technology, 2004. CIT ’04. The Fourth International Conference on*, pp. 345–350, Sept 2004. 8, 10

BIBLIOGRAFÍA

- [29] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*, HUC '99, (London, UK, UK), pp. 304–307, Springer-Verlag, 1999. 8
- [30] A. Kurti, "Context modeling to support the design of mobile learning," in *Proceedings of the 5th International Conference on Soft Computing As Transdisciplinary Science and Technology*, CSTST '08, (New York, NY, USA), pp. 536–541, ACM, 2008. 8
- [31] B. Hu and P. Moore, "Context modelling to support location based cooperative mobile learning." 8
- [32] A. Schmidt, "Potentials and challenges of context awareness for learning solutions," 2005. 8
- [33] W. Y. Lum and F. C. M. Lau, "A context-aware decision engine for content adaptation.," *IEEE Pervasive Computing*, vol. 1, no. 3, pp. 41–49, 2002. 8
- [34] M. Sharples, "The design of personal mobile technologies for lifelong learning," *Comput. Educ.*, vol. 34, pp. 177–193, Apr. 2000. 8
- [35] J. Berri, R. Benlamri, and Y. Atif, "Ontology-based framework for context-aware mobile learning," in *Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing*, IWCMC '06, (New York, NY, USA), pp. 1307–1310, ACM, 2006. 8
- [36] R. Benlamri and X. Zhang, "A global ontology space for mobile learning.," in *ICALT*, pp. 49–53, IEEE Computer Society, 2008. 9
- [37] W. Wahlster and A. Kobsa, "User models in dialog systems," in *User Models in Dialog Systems*, pp. 4–34, Springer-Verlag, 1989. 9
- [38] A. K. Dey and G. D. Abowd, "Towards a better understanding of context and context-awareness," in *In HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, pp. 304–307, Springer-Verlag, 1999. 10, 35
- [39] S. M. McNee, J. Riedl, and J. A. Konstan, "Being accurate is not enough: How accuracy metrics have hurt recommender systems," in *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, (New York, NY, USA), pp. 1097–1101, ACM, 2006. 10
- [40] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, pp. 30–37, Aug. 2009. 10
- [41] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, (New York, NY, USA), pp. 230–237, ACM, 1999. 10, 12, 28, 78
- [42] T. Kuroiwa and S. Bhalla, "Book recommendation system for utilisation of library services," *Int. J. Comput. Sci. Eng.*, vol. 5, pp. 207–213, Dec. 2010. 10
- [43] Ò. Celma, "Foafing the music: Bridging the semantic gap in music recommendation," in *Proceedings of 5th International Semantic Web Conference*, (Athens, GA, USA), pp. 927–934, 2006. 10

- [44] H. Berger, M. Denk, M. Dittenbach, A. Pesenhofer, and D. Merkl, “Photo-based user profiling for tourism recommender systems.” in *EC-Web* (G. Psaila and R. Wagner, eds.), vol. 4655 of *Lecture Notes in Computer Science*, pp. 46–55, Springer, 2007. 10
- [45] M. J. Pazzani, “A framework for collaborative, content-based and demographic filtering,” *Artif. Intell. Rev.*, vol. 13, pp. 393–408, Dec. 1999. 10, 29, 56
- [46] S. K. T. Lam, D. Frankowski, and J. Riedl, “Do you trust your recommendations? an exploration of security and privacy issues in recommender systems,” in *Proceedings of the 2006 International Conference on Emerging Trends in Information and Communication Security*, ETRICS’06, (Berlin, Heidelberg), pp. 14–29, Springer-Verlag, 2006. 11
- [47] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin, “Combining content-based and collaborative filters in an online newspaper,” 1999. 11, 13
- [48] R. Burke, “Hybrid recommender systems: Survey and experiments,” *User Modeling and User-Adapted Interaction*, vol. 12, pp. 331–370, Nov. 2002. 11, 13, 14, 54
- [49] M. Balabanović and Y. Shoham, “Fab: Content-based, collaborative recommendation,” *Commun. ACM*, vol. 40, pp. 66–72, Mar. 1997. 11, 67
- [50] N. Littlestone, “Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm,” *Mach. Learn.*, vol. 2, pp. 285–318, Apr. 1988. 11
- [51] J. B. Schafer, J. Konstan, and J. Riedl, “Recommender systems in e-commerce,” in *Proceedings of the 1st ACM Conference on Electronic Commerce*, EC ’99, (New York, NY, USA), pp. 158–166, ACM, 1999. 12
- [52] C. C. Aggarwal, J. L. Wolf, and P. S. Yu, “A new method for similarity indexing of market basket data,” in *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, SIGMOD ’99, (New York, NY, USA), pp. 407–418, ACM, 1999. 13
- [53] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl, “Application of dimensionality reduction in recommender system – a case study,” in *IN ACM WEBKDD WORKSHOP*, 2000. 13, 109
- [54] P. Yu, “Data mining and personalization technologies,” in *Database Systems for Advanced Applications, 1999. Proceedings., 6th International Conference on*, pp. 6–13, 1999. 13
- [55] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in *Proceedings of the 10th International Conference on World Wide Web*, WWW ’01, (New York, NY, USA), pp. 285–295, ACM, 2001. 13
- [56] M. Deshpande and G. Karypis, “Item-based top-n recommendation algorithms,” *ACM Trans. Inf. Syst.*, vol. 22, pp. 143–177, Jan. 2004. 13, 109
- [57] J. B. Schafer, J. A. Konstan, , J. Riedl, and J. Riedl, “E-commerce recommendation applications,” 2001. 13
- [58] R. Burke, “Knowledge-based recommender systems,” in *ENCYCLOPEDIA OF LIBRARY AND INFORMATION SYSTEMS*, p. 2000, Marcel Dekker, 2000. 13, 22, 31

BIBLIOGRAFÍA

- [59] M. Kerschberger, “Near field communication. a survey of safety and security measures,” tech. rep., Vienna, July 17, 2011. 14, 15, 16, 17
- [60] E. Vicente García, “Desarrollo de una aplicación de control de acceso y sistemas de identificación mediante la tecnología nfc,” tech. rep., Universidad Carlos III de Madrid, 2011. 14, 16
- [61] P. F. M. F. Carignano, “Tecnología inalámbrica near field communication y sus aplicaciones en sistemas embebidos,” in *Congreso argentino de sistemas embebidos (CASE)*, 2011. 15
- [62] A. d. G. G. M. V. Bueno Delgado, P. Pavón Mariño, “La tecnología nfc y sus aplicaciones en un entorno universitario,” tech. rep., Departamento de Tecnologías de la Información y las Comunicaciones, Universidad Politécnica de Cartagena, 2011. 15
- [63] D. A. Chavarría, “Tecnología de comunicación de campo cercano (nfc) y sus aplicaciones,” tech. rep., Universidad de Costa Rica, 2011. 15
- [64] D. F. Veloz, “Diseño e implementación de un prototipo para el control de acceso de personas aplicando la tecnología nfc por medio del uso de teléfonos celulares compatibles con esta tecnología,” tech. rep., Escuela Politécnica Nacional, 2010. 15, 16
- [65] A. C. Ruiz, “Desarrollo de una aplicación de pago a través de la tecnología nfc,” tech. rep., Universidad Carlos III de Madrid, 2011. 16
- [66] S. F. O. Aguirre, “Near field communication,” tech. rep., Departamento de Electrónica e Informática Universidad Católica “Nuestra Señora de la Asunción” Campus Santa Librada Asunción-Paraguay, 2014. 16
- [67] F. G. de la Sacristana, “Aplicación de inicio de sesión mediante autenticación con nfc,” tech. rep., Universidad Carlos III de Madrid, 2011. 17
- [68] T. Mitchell, “The discipline of machine learning,” Tech. Rep. CMU ML-06 108, 2006. 18
- [69] Merriam-Webster, “Definition of ‘learning’,” 2008. Consultado el 2 de noviembre de 2015. 18
- [70] P. Andreeva, M. Dimitrova, and P. Radeva, “Data mining learning models and algorithms for medical applications.” 18, 20, 62
- [71] E. Yndurain Gil, “Contribution to improve mobility uses through context-awareness,” 2012. 18, 19
- [72] N. Friedman and M. Goldszmidt, “Building classifiers using bayesian networks,” in *In Proceedings of the thirteenth national conference on artificial intelligence*, pp. 1277–1284, AAAI Press, 1996. 19, 20
- [73] D. Katsaros and Y. Manolopoulos, “A suffix tree based prediction scheme for pervasive computing environments,” in *Advances in Informatics, 10th Panhellenic Conference on Informatics, PCI 2005, Volos, Greece, November 11-13, 2005, Proceedings*, pp. 267–277, 2005. 19
- [74] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, “A practical guide to support vector classification,” tech. rep., Department of Computer Science, National Taiwan University, 2003. 19

- [75] L. Song, D. Kotz, R. Jain, and X. He, "Evaluating next-cell predictors with extensive wi-fi mobility data," in *IEEE Transactions on Mobile Computing*, pp. 1414–1424, 2004. 19
- [76] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, pp. 81–106, Mar. 1986. 20, 22
- [77] A. Cufoglu, M. Lohi, and K. Madani, "Classification accuracy performance of naive bayesian (nb), bayesian networks (bn), lazy learning of bayesian rules (lbr) and instance-based learner (ib1) - comparative study," in *Computer Engineering Systems, 2008. ICCES 2008. International Conference on*, pp. 210–215, Nov 2008. 20
- [78] K. Gopalratnam and D. J. Cook, "Active lezi: An incremental parsing algorithm for sequential prediction," in *In Sixteenth International Florida Artificial Intelligence Research Society Conference*, pp. 38–42, 2003. 20, 61
- [79] R. Begleiter, R. El-Yaniv, and G. Yona, "On prediction using variable order markov models.," *J. Artif. Intell. Res. (JAIR)*, vol. 22, pp. 385–421, 2004. 20
- [80] E. Kim, S. Helal, and D. Cook, "Human activity recognition and pattern discovery," *IEEE Pervasive Computing*, vol. 9, pp. 48–53, Jan. 2010. 20
- [81] S. Sigg, S. Haseloff, and K. David, "An alignment approach for context prediction tasks in ubicomp environments.," *IEEE Pervasive Computing*, vol. 9, no. 4, pp. 90–97, 2010. 20
- [82] A. Abraham, *Rule-Based Expert Systems*. John Wiley & Sons, Ltd, 2005. 20
- [83] L. Rokach and O. Maimon, "Top-down induction of decision trees classifiers - a survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 35, pp. 476–487, Nov 2005. 20
- [84] R. Kohavi, "The power of decision tables," in *Proceedings of the European Conference on Machine Learning*, pp. 174–189, Springer Verlag, 1995. 20
- [85] J. Cheng and R. Greiner, "Comparing bayesian network classifiers," in *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, UAI'99*, (San Francisco, CA, USA), pp. 101–108, Morgan Kaufmann Publishers Inc., 1999. 20
- [86] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Mach. Learn.*, vol. 6, pp. 37–66, Jan. 1991. 21
- [87] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods*. New York, NY, USA: Cambridge University Press, 2000. 21
- [88] L. Rokach and O. Maimon, *Data Mining with Decision Trees: Theory and Applications*. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2008. 21
- [89] T. Hancock, T. Jiang, M. Li, and J. Tromp, "Lower bounds on learning decision lists and trees," *Information and Computation*, vol. 126, no. 2, pp. 114 – 122, 1996. 22
- [90] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. 22

BIBLIOGRAFÍA

- [91] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Belmont, CA: Wadsworth International Group, 1984. 22
- [92] S. E. Middleton, D. C. De Roure, and N. R. Shadbolt, “Capturing knowledge of user preferences: Ontologies in recommender systems,” in *Proceedings of the 1st International Conference on Knowledge Capture, K-CAP '01*, (New York, NY, USA), pp. 100–107, ACM, 2001. 22
- [93] S. Shishehchi, S. Banihashem, and N. Zin, “A proposed semantic recommendation system for e-learning: A rule and ontology based e-learning recommendation system,” in *Information Technology (ITSim), 2010 International Symposium in*, vol. 1, pp. 1–5, June 2010. 22
- [94] N. Guarino, “Formal ontology and information systems,” in *Proceedings of Formal Ontology in Information Systems*, pp. 3–15, IOS Press, 1998. 22
- [95] e. a. Klein, M., *Ontologies and Schema Languages on the Web, in Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press, 2002. 22
- [96] A. Lozano Tello, *Mica de idoneidad de ontolog*. Tesis doctoral, 2002. 23
- [97] F.-L. M. a. G-Pz A, *Ontological Engineering*. Advanced Information and Knowledge Processing, Springer Verlag London, 2004. 23
- [98] R. L. Roberto, *EspecificaciL de una ontologara teleeducaci WEB semica*. Tesis doctoral, 2008. 23, 25
- [99] H. Knublauch, “Protege-owl api programmer’s guide,” tech. rep., 2010. Consultado el 2 de noviembre de 2015. 23
- [100] “The owl api,” 2012. Consultado el 2 de noviembre de 2015. 23
- [101] F. G. Shez, *Sistema basado en Tecnolog del Conocimiento para Entornos de Servicios Web Semicos*. PhD thesis, 2007. 24
- [102] J. M. Alban Gaignard, “Survey on semantic data stores and reasoning engines,” tech. rep., VIP Project, 2010. 24
- [103] K. Sirin, Parsia and Katz, “Pellet: A practical owl-dl reasoner,” *Web Semantics: science, services and agents on the World Wide Web*, vol. 5, no. 2, pp. 51–53, 2007. 24, 25
- [104] J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson, “Jena: Implementing the semantic web recommendations,” Tech. Rep. HPL-2003-146, Hewlett Packard Laboratories, 2003. 24
- [105] S. Bechhofer, R. Volz, and P. W. Lord, “Cooking the semantic web with the owl api,” in *The Semantic Web – ISWC 2003: Second International Semantic Web Conference, Sanibel Island, FL, USA*, pp. 659–675, Springer, 2006. 24
- [106] P. C. S. Bechhofer, R. Moller, “The dig description logic interface,” in *Proceedings of the 2003 International Workshop on Description Logics (DL2003)* (D. Calvanese, G. D. Giacomo, and E. Franconi, eds.), vol. 81 of *CEUR Workshop Proceedings*, (Rome, Italy), CEUR-WS.org, 2003. 24

- [107] E. R. . S. F. Luis Polo, Diego Berrueta, “Experimento semico para definir contextos y recursos: 1.0.,” *Proyecto MORFEO*, 2007. 25, 26
- [108] K. B. S. Bechhofer. Department of ComputerScience, “Fact++,” 2012. Consultado el 2 de noviembre de 2015. 25
- [109] S. I. de Investigaci la University of Innsbruck, “Racer renamed abox and concept expression reasoner,” tech. rep., 2012. Consultado el 2 de noviembre de 2015. 25
- [110] Y. C. Huang T., Li W., “Comparison of ontology reasoners: Racer, pellet, fact++,” in *American Geophysical Union, Fall Meeting*, vol. 1, p. 1068, 2008. 25
- [111] J. Lloyd, *Foundations of Logic Programming*. Berlin: Springer-Verlag, second ed., 1987. 25
- [112] C. M. R. A., W. C. Montand J. M. Martz, “Razonadores semicos: un estado del arte,” *Publicaci la Revista de la Facultad de Ingenier*, vol. A No. 21, 2010. 25
- [113] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosf, and M. Dean, “Swrl: A semantic web rule language combining owl and ruleml,” w3c member submission, World Wide Web Consortium, 2004. 26
- [114] J. J. S. Zapater, *Ontolog para servicios web semicos de informaci trco: descripcierramientas de explotaci*. PhD thesis, 2006. 26
- [115] Y. S. J. Z. W. Z. Y. Tian, “Managing and refining rule set for swrl,” *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*, pp. 1 – 5, 2008. 26
- [116] P. Patel-Schne, “Safe rules for owl 1.1,” tech. rep., Bell Labs Research, Alcatel-Lucent, 2008. 26
- [117] “Overview of owl-safe rules.” 2012. Consultado el 2 de noviembre de 2015. 26
- [118] “The rule markup initiative,” 2012. Consultado el 2 de noviembre de 2015. 26
- [119] R. L. and J. B., *Fundamentals of Speech Recognition*. Prentice Hall signal processing series, Englewood Cliffs, New Jersey: Prentice-Hall Signal Processing Series, 1993. 26
- [120] “Jena. A Semantic Web Framework for Java,” 2012. Consultado el 27 de enero de 2012. 26
- [121] “Jena 2 inference support, the transitive reasoner,” 2012. 26
- [122] Barrera., P., “Gesti la informaciltimedia en internet gestil conocimiento daml y ontolog consensuadas,” tech. rep., 2003. 26
- [123] “Context-awareness in mobile tourism guides – a comprehensive survey,” tech. rep., 2005. 27
- [124] M. V. Setten, S. Pokraev, J. Koolwaaij, and T. Instituut, “Context-aware recommendations in the mobile tourist application compass,” in *In Nejd, W. De Bra, P. (Eds.). AH 2004, LNCS 3137*, pp. 235–244, Springer-Verlag, 2004. 27, 28, 53

BIBLIOGRAFÍA

- [125] S. Poslad, H. Laamanen, R. Malaka, A. Nick, P. Buckle, and A. Zipf, “CRUMPET: creation of user-friendly mobile services personalised for tourism,” in *3G Mobile Communication Technologies, 2001. Second International Conference on (Conf. Publ. No. 477)*, pp. 28–32, 2001. 27
- [126] B. S. Belz, A. Nick, S. Poslad, and A. Zipf, “Personalized and Location-based Mobile Tourism Services,” in *Proc. Mobile-HCI, 2002*. 27
- [127] K. Cheverst, N. Davies, K. Mitchell, and A. Friday, “Experiences of developing and deploying a context-aware tourist guide: The guide project,” in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, MobiCom '00*, (New York, NY, USA), pp. 20–31, ACM, 2000. 27
- [128] K. Cheverst, N. Davies, K. Mitchell, A. Friday, and C. Efstratiou, “Developing a context-aware electronic tourist guide: Some issues and experiences,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '00*, (New York, NY, USA), pp. 17–24, ACM, 2000. 27
- [129] M. J. O’Grady and G. M. P. O’Hare, “Gulliver’s genie: agency, mobility, adaptivity.,” *Computers Graphics*, vol. 28, no. 5, pp. 677–689, 2004. 28
- [130] N. Hristova, G. M. P. O’hare, and T. Lowen, “T.: Agent-based ubiquitous systems: 9 lessons learnt,” in *In Workshop on System Support for Ubiquitous Computing (UbiSys'03), 5th International Conference on Ubiquitous Computing (UbiComp, 2003)*. 28
- [131] “Lol@: A umts location based service,” in *In: International Symposion on 3rd Generation Infrastructure and Services, Athen, 2001*. 28
- [132] “Mobidenk-mobile multimedia in monument conservation.,” *IEEE MultiMedia*, vol. 11, no. 2, pp. 72–77, 2004. 28
- [133] A. Kamar, “Mobile tourist guide (m-toguide),” *Deliv. 14 Proj. Final Rep. IST2001*, vol. 36004, 2003. 28
- [134] J. Roth, “Context-aware web applications using the pinpoint infrastructure,” in *Proceedings of the IADIS International Conference WWW/Internet*, pp. 3–10, November 2002. 28
- [135] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” *IEEE Trans. on Knowl. and Data Eng.*, vol. 17, pp. 734–749, June 2005. 28, 67
- [136] A. M. Team, “Apache mahout: Scalable machine-learning and data-mining library,” 2011. 28
- [137] M. D. Ekstrand, M. Ludwig, J. Kolb, and J. T. Riedl, “Lenskit: A modular recommender framework,” in *Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys '11*, (New York, NY, USA), pp. 349–350, ACM, 2011. 28
- [138] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou, “Mobile recommender systems in tourism,” *Journal of Network and Computer Applications*, vol. 39, pp. 319 – 333, 2014. 28

- [139] Y. Zheng and X. Xie, "Learning travel recommendations from user-generated gps traces," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 2:1–2:29, Jan. 2011. 29
- [140] P. D. Bitonto, F. D. Tria, M. Laterza, T. Roselli, V. Rossano, and F. Tangorra, "Automated generation of itineraries in recommender systems for tourism.," in *ICWE Workshops* (F. Daniel and F. M. Facca, eds.), vol. 6385 of *Lecture Notes in Computer Science*, pp. 498–508, Springer, 2010. 29
- [141] "moretourism: Mobile recommendations for tourism," in *Consumer Electronics (ICCE), 2011 IEEE International Conference on*, pp. 347–348, Jan 2011. 29
- [142] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin, "Incorporating contextual information in recommender systems using a multidimensional approach," *ACM Transactions on Information Systems*, vol. 23, pp. 103–145, 2005. 29, 53
- [143] M. Vozalis and K. G. Margaritis, "Collaborative filtering enhanced by demographic correlation," in *Proceedings of the AIAI Symposium on Professional Practice in AI, part of the 18th World Computer Congress*, pp. 293–402, 2004. 29
- [144] I. Satoh, "A location model for pervasive computing environments," in *Proceedings of IEEE 3rd International Conference on Pervasive Computing and Communications (PerCom'05)*, pp. 215–224, IEEE Computer Society, Press, 2005. 29
- [145] I. Smith, J. Tabert, T. Wild, A. Lamarca, A. Lamarca, Y. Chawathe, Y. Chawathe, S. Consolvo, S. Consolvo, J. Hightower, J. Hightower, J. Scott, J. Scott, T. Sohn, T. Sohn, J. Howard, J. Howard, J. Hughes, J. Hughes, F. Potter, F. Potter, P. Powledge, P. Powledge, G. Borriello, G. Borriello, B. Schilit, and B. Schilit, "Place lab: Device positioning using radio beacons in the wild," in *In Proceedings of the Third International Conference on Pervasive Computing*, pp. 116–133, Springer, 2005. 29
- [146] P. Bahl and V. N. Padmanabhan, "Radar: an in-building rf-based user location and tracking system," pp. 775–784, 2000. 29
- [147] y. E. J. M. F. Alizadeh-Shabdiz, O. IGNATYEV, "System and method for estimating positioning error within a wlan-based positioning system." Google Patents, 2009. 29
- [148] J. Chon and H. Cha, "Lifemap: A smartphone-based context provider for location-based services.," *IEEE Pervasive Computing*, vol. 10, no. 2, pp. 58–67, 2011. 29, 30
- [149] P. Jaring, V. Törmänen, E. Siira, and T. Matinmikko, "Improving mobile solution workflows and usability using near field communication technology," in *Ambient Intelligence* (B. Schiele, A. Dey, H. Gellersen, B. de Ruyter, M. Tscheligi, R. Wichert, E. Aarts, and A. Buchmann, eds.), vol. 4794 of *Lecture Notes in Computer Science*, pp. 358–373, Springer Berlin Heidelberg, 2007. 30
- [150] T. Halevi, D. Ma, N. Saxena, and T. Xiang, "Secure proximity detection for nfc devices based on ambient sensor data.," in *ESORICS* (S. Foresti, M. Yung, and F. Martinelli, eds.), vol. 7459 of *Lecture Notes in Computer Science*, pp. 379–396, Springer, 2012. 30

BIBLIOGRAFÍA

- [151] A. A. y R. Karlsen, “Experimenting with instant services using nfc technology,” in *SMART 2012, The First International Conference on Smart Systems, Devices and Technologies*, pp. 73–78, 2012. 30
- [152] F. Borrego-Jaraba, I. Luque Ruiz, and M. A. Gómez-Nieto, “A nfc-based pervasive solution for city touristic surfing,” *Personal Ubiquitous Comput.*, vol. 15, pp. 731–742, Oct. 2011. 30
- [153] J. K. K. da Silva, G. M. Machado, L. H. Thom, and L. K. Wives, “Business process modeling and instantiation in home care environments,” in *ICEIS 2014 - Proceedings of the 16th International Conference on Enterprise Information Systems, Volume 2, Lisbon, Portugal, 27-30 April, 2014*, pp. 513–525, 2014. 31
- [154] A. Bouzeghoub, K. N. Do, and L. K. Wives, “Situation-aware adaptive recommendation to assist mobile users in a campus environment,” in *The IEEE 23rd International Conference on Advanced Information Networking and Applications, AINA 2009, Bradford, United Kingdom, May 26-29, 2009*, pp. 503–509, 2009. 31
- [155] J. L. C. F. y R. C. I. G. G. Suárez, T. D. Fernández, “Context-aware recommender system based on ontologies,” 31
- [156] E. R. C. Simó, “Development of a tourism recommender system,” 2012. 31
- [157] M. Rey-Lopez, A. B. Barragans-Martinez, A. Peleteiro, F. A. Mikic-Fonte, and J. C. Burguillo, “moreTourism: Mobile recommendations for tourism,” in *Consumer Electronics (ICCE), 2011 IEEE International Conference on*, pp. 347–348, IEEE, Jan. 2011. 31
- [158] H. Park, S. Kwon, and H. Kwon, “Ontology-based approach to intelligent ubiquitous tourist information system,” in *Proceedings of the 4th International Conference on Ubiquitous Information Technologies & Applications (ICUT)*, (Fukuoka), 2009. 32
- [159] O. Ozdakis, F. Orhan, and F. Danismaz, “Ontology-based recommendation for points of interest retrieved from multiple data sources,” in *Proceedings of the International Workshop on Semantic Web Information Management, SWIM '11*, (New York, NY, USA), pp. 1:1–1:6, ACM, 2011. 32
- [160] Z. Yu, Y. Nakamura, S. Jang, S. Kajita, and K. Mase, “Ontology-based semantic recommendation for context-aware e-learning,” in *Ubiquitous Intelligence and Computing* (J. Indulska, J. Ma, L. Yang, T. Ungerer, and J. Cao, eds.), vol. 4611 of *Lecture Notes in Computer Science*, pp. 898–907, Springer Berlin Heidelberg, 2007. 32
- [161] Y. M. T. K. R. S. M. B. S. P. Sarunyoo Ngamsrithepparit, Thepchai Supnithi, “Rule management system for ontology-based recommendation system,” tech. rep., National Electronics and Computer Technology Center (NECTEC), 2012. 32
- [162] S. C. Kabore, *Design and implementation of a recommender system as a module for Liferay portal*. PhD thesis, 2012. 33, 40, 95
- [163] A. K. Dey, “Understanding and using context,” *Personal and Ubiquitous Computing*, vol. 5, pp. 4–7, 2001. 34

- [164] H. w. Gellersen, A. Schmidt, and M. Beigl, “Multi-sensor context-awareness in mobile devices and smart artefacts,” *Mob. Netw. Appl.*, vol. 7, pp. 341–351, 2002. 34
- [165] L. Barkhuus, L. Barkhuus, A. Dey, and A. Dey, “Is context-aware computing taking control away from the user? three levels of interactivity examined,” in *In Proceedings of Ubicomp 2003*, pp. 149–156, Springer, 2003. 34
- [166] O. Lassila and D. Khushraj, “Contextualizing applications via semantic middleware,” in *Proceedings of the The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, MOBIQUITOUS '05*, (Washington, DC, USA), pp. 183–191, IEEE Computer Society, 2005. 34
- [167] J. Coutaz, J. L. Crowley, S. Dobson, and D. Garlan, “Context is key,” *Commun. ACM*, vol. 48, pp. 49–53, Mar. 2005. 34
- [168] T. R. Gruber, “A translation approach to portable ontology specifications,” *KNOWLEDGE ACQUISITION*, vol. 5, pp. 199–220, 1993. 35
- [169] C. Bettini, J. Indulska, D. Nicklas, A. Ranganathan, D. Riboni, and et al., “A survey of context modelling and reasoning techniques,” 2008. 35
- [170] R. Zhou and K. Rechert, “Personalization for location-based e-learning,” in *Proceedings of the 2008 The Second International Conference on Next Generation Mobile Applications, Services, and Technologies, NGMAST '08*, (Washington, DC, USA), pp. 247–253, IEEE Computer Society, 2008. 36
- [171] G. O. A. R. Network, “Travel guide ontology,” 2003. Consultado el 26 de septiembre de 2014. 39, 60, 72
- [172] J. L. Herlocker and J. A. Konstan, “Content-independent task-focused recommendation,” *IEEE Internet Computing*, vol. 5, no. 6, pp. 40–47, 2001. 52
- [173] G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton, “Cyberguide: A mobile context-aware tour guide,” 1996. 53
- [174] F. Cena, L. Console, C. Gena, A. Goy, G. Levi, S. Modeo, and I. Torre, “Integrating heterogeneous adaptation techniques to build a flexible and usable mobile tourist guide,” *AI Commun.*, vol. 19, pp. 369–384, December 2006. 53
- [175] K. Oku, S. Nakajima, J. Miyazaki, and S. Uemura, “Context-aware svm for context-dependent information recommendation,” in *Proceedings of the 7th International Conference on Mobile Data Management, MDM '06*, (Washington, DC, USA), pp. 109–, IEEE Computer Society, 2006. 53
- [176] U. Panniello, A. Tuzhilin, M. Gorgoglione, C. Palmisano, and A. Pedone, “Experimental comparison of pre- vs. post-filtering approaches in context-aware recommender systems,” in *Proceedings of the Third ACM Conference on Recommender Systems, RecSys '09*, (New York, NY, USA), pp. 265–268, ACM, 2009. 53

BIBLIOGRAFÍA

- [177] Z. Yu, X. Zhou, D. Zhang, C.-Y. Chin, X. Wang, and J. Men, “Supporting context-aware media recommendations for smart phones,” *IEEE Pervasive Computing*, vol. 5, pp. 68–75, July 2006. 53
- [178] M. Raento, A. Oulasvirta, R. Petit, and H. Toivonen, “Contextphone: A prototyping platform for context-aware mobile applications,” *IEEE Pervasive Computing*, vol. 4, pp. 51–59, Apr. 2005. 59
- [179] A. Kivi, “Measuring mobile user behavior and service usage: Methods, measurement points, and future outlook,” in *Proceedings of the 6th Global Mobility Roundtable*, 2007. 60
- [180] C. Moisa, “Conceptual clarifications regarding youth travel,” *Revista Tinerilor Economisti (The Young Economists Journal)*, vol. 1, no. 14S, pp. 98–106, 2010. 60
- [181] T. I. Traveler, “Define clubbing tourism.” Consultado el 26 de noviembre de 2014. 60
- [182] Tourism and More, “The family vacation,” 2003. Consultado el 26 de noviembre de 2014. 60
- [183] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The weka data mining software: An update,” *SIGKDD Explor. Newsl.*, vol. 11, pp. 10–18, Nov. 2009. 62
- [184] G. Holmes, A. Donkin, and I. H. Witten, “Weka: a machine learning workbench,” pp. 357–361, August 1994. 62
- [185] M. Pazzani and D. Billsus, “Learning and revising user profiles: The identification of interesting web sites,” *Mach. Learn.*, vol. 27, pp. 313–331, June 1997. 67
- [186] T. R. Gruber, “A translation approach to portable ontology specifications,” *Knowl. Acquis.*, vol. 5, pp. 199–220, June 1993. 67
- [187] B. McBride, “Rdf vocabulary description language 1.0: Rdf schema,” 2004. 67
- [188] D. L. McGuinness and F. van Harmelen, “Owl web ontology language overview,” Tech. Rep. REC-owl-features-20040210, W3C, 2004. 67
- [189] P. P.-S. Chen, “The entity-relationship model—toward a unified view of data,” *ACM Trans. Database Syst.*, vol. 1, pp. 9–36, Mar. 1976. 67
- [190] I. Horrocks and P. F. Patel-Schneider, “A proposal for an owl rules language,” in *Proceedings of the 13th International Conference on World Wide Web, WWW '04*, (New York, NY, USA), pp. 723–731, ACM, 2004. 67, 68
- [191] M. Eirinaki and M. Vazirgiannis, “Web mining for web personalization,” *ACM Trans. Internet Technol.*, vol. 3, pp. 1–27, Feb. 2003. 68
- [192] G. Adomavicius and A. Tuzhilin, “Personalization technologies: A process-oriented perspective,” *Commun. ACM*, vol. 48, pp. 83–90, Oct. 2005. 68
- [193] K. Lakiotaki, P. Delias, V. Sakkalis, and N. F. Matsatsinis, “User profiling based on multi-criteria analysis: the role of utility functions,” *Operational Research*, vol. 9, no. 1, pp. 3–16, 2009. 68

- [194] J. Trajkova and S. Gauch, “Improving ontology-based user profiles.,” in *RIAO* (C. Fluhr, G. Greifenstette, and W. B. Croft, eds.), pp. 380–390, CID, 2004. 68
- [195] S. E. Middleton, N. R. Shadbolt, and D. C. De Roure, “Ontological user profiling in recommender systems,” *ACM Trans. Inf. Syst.*, vol. 22, pp. 54–88, Jan. 2004. 68
- [196] D. Godoy and A. Amandi, “User profiling for web page filtering,” *IEEE Internet Computing*, vol. 9, pp. 56–64, July 2005. 68
- [197] S. Gauch, J. Chaffee, and A. Pretschner, “Ontology-based personalized search and browsing,” *Web Intelli. and Agent Sys.*, vol. 1, pp. 219–234, Dec. 2003. 68
- [198] J. H. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubézy, H. Eriksson, N. F. Noy, and S. W. Tu *Int. J. Hum.-Comput. Stud.*, vol. 58, pp. 89–123, Jan. 2003. 69
- [199] D. Heckmann, *Ubiquitous user modeling*. PhD thesis, Saarland University, 2006. 69, 70, 71
- [200] “Etp-tourism.owl.” Consultado el 25 de septiembre de 2014. 72
- [201] P. J. Ludford, D. Cosley, D. Frankowski, and L. G. Terveen, “Think different: increasing online community participation using uniqueness and group dissimilarity.,” in *CHI* (E. Dykstra-Erickson and M. Tscheligi, eds.), pp. 631–638, ACM, 2004. 76
- [202] S. Owen, R. Anil, T. Dunning, and E. Friedman, *Mahout in Action*. Greenwich, CT, USA: Manning Publications Co., 2011. 76, 78, 79
- [203] U. Shardanand and P. Maes, “Social information filtering: Algorithms for automating “word of mouth”,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’95*, (New York, NY, USA), pp. 210–217, ACM Press/Addison-Wesley Publishing Co., 1995. 77
- [204] W. Hill, L. Stead, M. Rosenstein, and G. Furnas, “Recommending and evaluating choices in a virtual community of use,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’95*, (New York, NY, USA), pp. 194–201, ACM Press/Addison-Wesley Publishing Co., 1995. 77
- [205] P. Resnick and R. Sami, “The influence limiter: Provably manipulation-resistant recommender systems,” in *Proceedings of the 2007 ACM Conference on Recommender Systems, RecSys ’07*, (New York, NY, USA), pp. 25–32, ACM, 2007. 77
- [206] J. Herlocker, J. A. Konstan, and J. Riedl, “An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms,” *Inf. Retr.*, vol. 5, pp. 287–310, Oct. 2002. 78
- [207] H. Shimodaira, “Similarity and recommender systems,” 2015. 78
- [208] A. Tomcat, “The apache software foundation,” 2009. Consultado el 25 de septiembre de 2014. 79, 80
- [209] R. of the University of Minnesota, “Lenskit recommender toolkit,” 2010. Consultado el 25 de septiembre de 2014. 79

BIBLIOGRAFÍA

- [210] M. S. Gashler, “Lecture notes on machine learning,” 2013. Consultado el 25 de septiembre de 2014. 79
- [211] M. Hahsler, “Recommenderlab: A framework for developing and testing recommendation algorithms,” 2014. Consultado el 25 de septiembre de 2014. 79
- [212] D. Pisinger, “Algorithms for knapsack problems,” 1995. 89
- [213] M. P. Karla L. Hoffman and G. Rinaldi, “Traveling salesman problem.” 92
- [214] N. Forum, “Nfc forum,” 2015. Consultado el 2 de febrero de 2015. 99, 100
- [215] A. Developers, “Nfc basics,” 2015. Consultado el 20 de marzo de 2014. 99
- [216] T. A. S. Foundation, “Jena ontology api,” 2015. Consultado el 27 de septiembre de 2014. 101
- [217] “Primefaces,” 2015. Consultado el 20 de marzo de 2014. 106
- [218] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, “GroupLens: An open architecture for collaborative filtering of netnews,” pp. 175–186, ACM Press, 1994. 109
- [219] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems,” *ACM Trans. Inf. Syst.*, vol. 22, pp. 5–53, Jan. 2004. 109
- [220] C. Cleverdon and M. Kean, “Factors determining the performance of indexing systems.” Aslib Cranfield Research Project, Cranfield, England, 1968. 109
- [221] D. Billsus and M. J. Pazzani, “Learning collaborative information filters,” in *Proceedings of the Fifteenth International Conference on Machine Learning, ICML '98*, (San Francisco, CA, USA), pp. 46–54, Morgan Kaufmann Publishers Inc., 1998. 109
- [222] C. Basu, H. Hirsh, and W. Cohen, “Recommendation as classification: Using social and content-based information in recommendation,” in *In Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pp. 714–720, AAAI Press, 1998. 109
- [223] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Analysis of recommendation algorithms for e-commerce,” in *Proceedings of the 2nd ACM conference on Electronic commerce, EC '00*, (New York, NY, USA), pp. 158–167, ACM, 2000. 109, 117
- [224] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008. 110
- [225] J. S. Breese, D. Heckerman, and C. Kadie, “Empirical analysis of predictive algorithms for collaborative filtering,” in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, UAI'98*, (San Francisco, CA, USA), pp. 43–52, Morgan Kaufmann Publishers Inc., 1998. 111
- [226] P. Pu, L. Chen, and R. Hu, “Evaluating recommender systems from the user’s perspective: survey of the state of the art,” *User Model. User-Adapt. Interact.*, vol. 22, no. 4-5, pp. 317–355, 2012. 112

