

# Vizier: un sistema de recomendación genérico y multidimensional

## Vizier: a generic and multidimensional recommendation framework

Andrea Barraza-Urbina Est. & Angela Carrillo Ramos Ph. D.

Departamento de Ingeniería de Sistemas, Pontificia Universidad Javeriana, Bogotá, Colombia.

*abarraza@javeriana.edu.co, angela.carrillo@javeriana.edu.co*

Recibido para revisión 30 de agosto de 2011, aceptado 18 de octubre de 2011, versión final 14 de noviembre de 2011

**Resumen** — Los *Sistemas de Recomendación*, al ser herramientas que puedan ser utilizadas para identificar fácilmente productos de interés, han surgido para apoyar, aumentar y automatizar el proceso natural y diario de compartir recomendaciones, y de esta manera, reducir un espacio de alternativas. Este artículo tiene como objetivo presentar Vizier, un framework construido bajo un enfoque genérico, que proporciona servicios a aplicaciones de recuperación de información para que éstas a su vez, se encuentren en la capacidad de ofrecer recomendación de productos/servicios tomando en cuenta diversas dimensiones de *Adaptación/Personalización* (por ejemplo, la dimensión del usuario, el contexto, entre otros). Por un lado, con Vizier se espera beneficiar a aquellas entidades que actualmente desarrollan aplicaciones de recuperación de información y desean agregar servicios de recomendación (por ejemplo, aplicaciones *e-commerce*). Por otro lado, se busca ofrecer una solución que idealmente proporcione resultados adaptados/personalizados que consideren la multidimensionalidad de los usuarios, productos y el contexto. Con el fin de validar *Vizier* se creó *ZoundBeat*, prototipo funcional de un reproductor de música que es capaz de invocar los servicios del framework propuesto con el ánimo de ofrecer a sus usuarios recomendación de canciones.

**Palabras Clave** — Sistema Multi-Agente, Sistema de Recomendación, Genérico, Multidimensional, Personalización, Adaptación.

**Abstract** — Recommender Systems have emerged to help support, augment and systematize the everyday natural social process of creating and sharing recommendations by developing tools that can be used to quickly identify interesting products, and therefore, reduce a search space of alternatives. This paper aims to present Vizier, a framework constructed under a generic approach, which provides services to Information Retrieval applications so these may offer product recommendations that consider several Adaptation/Personalization dimensions (e.g., user dimension, context, among others). On the one hand, with Vizier it is expected to assist those entities that currently develop Information Retrieval

applications and wish to add recommendations to their services (e.g., E-Commerce applications). On the other hand, it is hoped to offer a solution that hopefully provides adapted/personalized results by considering the multidimensionality of users, items and context. In order to validate Vizier, ZoundBeat was created, a functional prototype of a music player that is capable of invoking the proposed framework to offer its users song recommendations.

**Keywords** — Multi-Agent System, Recommendation System, Generic, Multidimensional, Personalization, Adaptation.

### I. INTRODUCCIÓN

Hoy en día, gracias al surgimiento del Internet y el auge de las tecnologías de la comunicación, cualquier individuo puede tener acceso a grandes cantidades de información a través de casi cualquier dispositivo, prácticamente desde cualquier lugar del mundo. Los avances tecnológicos han superado los límites del espacio y del tiempo, revolucionando la forma de vida actual, permitiendo que individuos puedan comunicarse y compartir cualquier tipo de datos, sin esfuerzo y casi sin ningún costo. Como consecuencia, las exigencias de la Sociedad de la Información han cambiado, y hay una demanda creciente de información de manera inmediata y en grandes volúmenes. Así como la demanda de información ha aumentado, también lo ha hecho la oferta. Sin embargo, los mecanismos existentes creados con el fin de conectar dicha demanda y dicha oferta se han quedado cortos; además en la literatura revisada, técnicas eficaces para la recuperación de información, así como sistemas de apoyo eficientes, aún no se han creado. No obstante, las herramientas adecuadas deben ser desarrolladas para apoyar a usuarios en la tarea de encontrar la información que realmente necesitan, cuando más la necesitaban, tomando en cuenta las limitaciones impuestas por su dispositivo de acceso.

Una posible respuesta se puede encontrar en la creación de *Sistemas de Recomendación*. Los *Sistemas de Recomendación*, al ser herramientas que puedan ser utilizadas para identificar fácilmente “*cosas que valen la pena*” (*Konstan* [11], explica que *Cosas es una palabra técnica que no limita lo que se recomienda, pueden ser productos, servicios o personas... son cosas!*)[11], han surgido para apoyar, aumentar y automatizar el proceso natural y diario de compartir recomendaciones, y de esta manera, reducir un espacio de alternativas[4][20][18]. En esencia, estos sistemas son capaces de guiar la tarea de navegar (o *browse*) entre un mar de información al ofrecerle de manera proactiva sugerencias personalizadas a sus usuarios; y de esta manera, permitirles descubrir productos útiles e interesantes en menos tiempo. Además, los *Sistemas de Recomendación* se encuentran en la capacidad de responder a requisitos de búsqueda incompletos y a necesidades implícitas de un usuario, al sugerirle productos o servicios que éste ni siquiera sabía que existían, mucho menos sabía qué quería, permitiéndole al usuario descubrir algo completamente nuevo, aquellas *joyas ocultas*, incluso cuando el usuario tiene necesidades de información poco claras o generales (por ejemplo, “*yo quiero una galleta*”) [13].

Se han llevado a cabo numerosos trabajos en el área de *Sistemas de Recomendación*; sin embargo, ninguno es completamente exitoso y todavía necesitan considerables mejoras. Una debilidad es que las técnicas existentes de recomendación tienen una fuerte dependencia al tipo de producto o el tipo de dominio para el cual fueron diseñadas, y, en consecuencia, no son genéricas. Además, la mayoría de los *Sistemas de Recomendación* actuales operan en un espacio de dos dimensiones (considerando sólo información sobre el usuario y los productos que se ofrecen) y no se adaptan al contexto de un usuario [1][19]. Aunque la inclusión de características contextuales en técnicas de recomendación aún no ha demostrado en todos los casos proporcionar mejoras significativas sobre los *Sistemas de Recomendación* tradicionales, considerar el contexto potencialmente podría permitir generar recomendaciones más acertadas y tiene beneficios aún no descubiertos.

Este artículo tiene como objetivo presentar *Vizier*, un framework diseñado bajo un enfoque genérico, que proporciona servicios a aplicaciones orientadas a la recuperación de información para que éstas, a su vez, pueden ofrecer recomendaciones de productos/servicios que toman en cuenta diversos aspectos de *Adaptación/Personalización* (por ejemplo, las dimensiones usuario, contexto). A fin de validar este framework, se desarrolló *ZoundBeat*, una aplicación destinada a recomendar listas de reproducción de canciones a sus usuarios, considerando características contextuales, por medio del uso de los servicios ofrecidos por *Vizier*.

El presente se encuentra organizado de la siguiente manera: el *modelo general de Vizier* se explica en la *sección 2*. A continuación, en la *sección 3*, se especifica por qué los Sistemas

Multi-Agente son herramientas adecuadas para la construcción de *Sistemas de Recomendación*. La *sección 4* presenta el diseño de *Vizier*. Con el fin de validar *Vizier*, se creó el reproductor de música denominado *ZoundBeat*, el cual es presentado en la *sección 5*. Una visión general de la implementación del prototipo funcional se muestra en la *sección 6* y la *sección 7* analiza los resultados obtenidos. Posteriormente, la *sección 8* compara trabajos relacionados y finalmente, la *sección 9* presenta conclusiones y trabajos futuros.

## II. MODELO GENERAL DE VIZIER

El primer paso hacia el diseño de *Vizier* fue identificar todos los componentes que un *Sistema de Recomendación* Multidimensional y Genérico debe incluir para luego organizarlos en el *Modelo General de Vizier*. Los componentes que se incluirán en el modelo se identificaron en la lista de decisiones clave que deben ser tomadas en cuenta para la construcción de cualquier *Sistema de Recomendación*, que además, puede considerar criterios de *Adaptación/Personalización*, especificada en [6]. La lista y el *Modelo General de Vizier* fueron diseñados para servir de fundamento para desarrolladores que deseen realizar cualquier tipo de *Sistema de Recomendación* que también podría ser *Multidimensional y Genérico*.

Se determinó que los componentes que todo *Sistema de Recomendación* debe tener son: información de usuario-producto (es decir, información que indique la utilidad que representa un producto para un usuario), un componente a cargo de la *recopilación de datos*, al menos una *técnica de recomendación* y un componente a cargo de la *entrega y presentación de los resultados de recomendación*. Además, se podría considerar un componente opcional que se encuentre a cargo de la evaluación del sistema, así como de la mejora continua de otros componentes, considerando la retroalimentación obtenida a partir de la evaluación. Además, dado que algunas técnicas de recomendación sólo pueden ser consideradas bajo ciertas condiciones, entonces el uso de múltiples técnicas de recomendación dentro del mismo motor de recomendación podría ser conveniente. Si varias técnicas de recomendación son necesarias, entonces un componente encargado de seleccionar la técnica de recomendación más útil para una situación particular es requerido. Además, un *Sistema de Recomendación* de carácter **Genérico** debe tener un componente que actúe como interfaz entre la información de cualquier tipo de perfil y las técnicas de recomendación. Para ir un paso más allá, el uso de perfiles genéricos podría ser útil, dado que la información básica (que puede ser compartida entre los diferentes servicios) disponible para cualquier situación sería claramente especificada. Por último, los componentes que un ***Sistema de Recomendación Multidimensional*** debe tener son: *perfiles de usuario, productos y contexto y técnicas de*

recomendación que consideren el contexto para la construcción de las recomendaciones.

Ahora, después de que todos los componentes han sido identificados, el segundo paso es organizarlos en el Modelo

General de Vizier. Este modelo sirvió de base para el diseño inicial del *Framework Vizier*, que se presenta en la sección 4. La arquitectura del Modelo General de Vizier se puede observar en Figura 1.

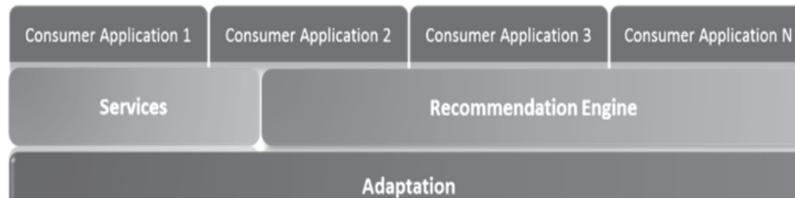


Figura 1. Modelo General de Vizier

El modelo se divide en dos capas: la capa de *Adaptación* y la capa del *Motor de Recomendación*. Por un lado, la capa de *Adaptación* es la encargada de recopilar datos de las fuentes de datos disponibles, y utilizar estos datos, para actualizar y mantener los perfiles. Por otro lado, la capa del *Motor de Recomendación* utiliza la información proporcionada por la capa de *Adaptación*, con el fin de proporcionar servicios de recomendación a las diferentes *aplicaciones consumidoras*. Un componente adicional denominado *Servicios* se puede observar en el modelo. Este componente no es más que una representación de los demás componentes que se comunican con la capa de *Adaptación*, pero que no proporcionan servicios de recomendación. Las siguientes secciones ofrecen una visión detallada tanto de la capa de *Adaptación* como de la capa de *Motor de Recomendación*.

### 2.1 Capa de Adaptación

La Figura 2 muestra una vista detallada de la capa de *Adaptación*. Esta capa es la encargada de recopilar los datos de las fuentes de datos disponibles, y utilizar estos datos para actualizar y mantener los perfiles que se encuentran en el componente *Profile Set*. Además, a través del componente *Interface*, la información de los perfiles se puede acceder fácilmente desde las diversas técnicas de recomendación que se encuentran en la capa *Recommendation Engine*. Por otro lado, el componente *Data Collector* se encuentra a cargo de la recuperación de datos de manera implícita o explícita de las fuentes de datos, y de comunicarle dichos datos, al componente *Profile Updater*. El *Profile Updater* se encarga de tomar los datos en bruto y si es necesario, pre-procesarlos para convertirlos en información útil que luego se adjunta al *Profile Set*. Si se ha elegido una técnica para crear un perfil inicial, los componentes *Data Collector* y *Profile Updater* pueden cooperar para construir este perfil inicial. Finalmente, el componente *Profile Maintainer* es el encargado del mantenimiento de los perfiles mediante la administración de tareas relacionadas con: funciones de olvido, la construcción y el uso de modelos predictivos, y modificar los perfil dadas solicitudes de los usuarios (con el fin de llevar a cabo esta última tarea, el componente *Profile Maintainer* debe cooperar con el componente *Data Collector*).

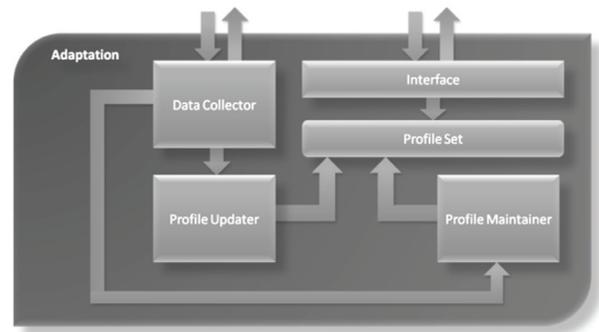


Figura 2. Visión detallada de la capa de Adaptación

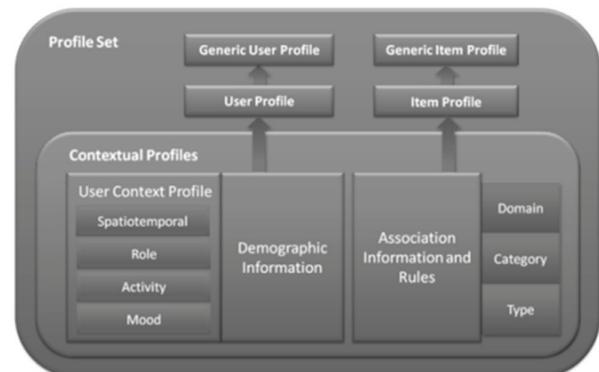


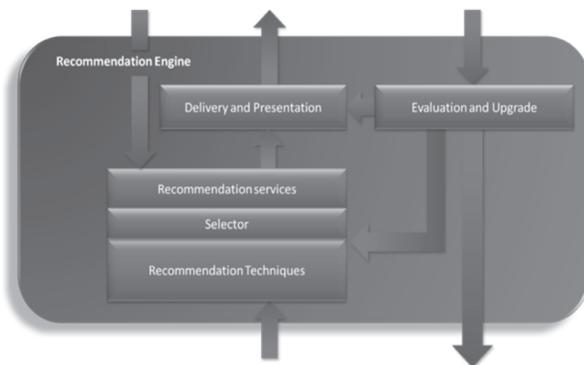
Figura 3. Visión detallada del componente Profile Set

La Figura 3 propone una organización para el componente *Profile Set*. No obstante, debido a que el *Modelo General de Vizier* debe ser lo suficientemente genérico y flexible para soportar diversas maneras de representar los datos, el componente *Profile Set* se puede organizar de diferentes maneras, siempre y cuando el componente *Interface* se adapte a dicha organización. La organización propuesta para el *Profile Set* cuenta con tres tipos de perfiles: *User Profile*, *Item Profile* y *Contextual Profiles*. Los perfiles *User Profile* e *Item Profile* pueden extender de un *Perfil Genérico* (*Perfil Genérico de Usuario* y *Perfil Genérico de Producto*) que determina la información básica que estaría disponible para cualquier tipo de servicio. Por un lado, el *User Profile* contiene información relevante para los usuarios, sus gustos, preferencias, intereses,

necesidades, entre otros. Por otro lado, el *Item Profile* contiene información que describe un producto. El tercer tipo de perfil es el *Contextual Profile*, que puede ser independiente o dependiente con respecto a los otros perfiles. Un *Contextual Profile* dependiente se encuentra estrictamente relacionado a un *User Profile* o a un *Item Profile*. Un *Contextual Profile* independiente (por ejemplo, *información demográfica*), no se encuentra directamente relacionado a un *User Profile* o a un *Item Profile*. Sin embargo, los *Perfiles de Usuario* y *Perfiles de Producto* pueden asociarse con *Perfiles Contextuales* independientes. Se propone que los *Perfiles Contextuales* que pueden ser asociados con un *User Profile* son los siguientes: (i) *Perfil Contextual de Usuario*: contiene información relativa al contexto de un usuario, tales como las actividades que él/ella lleva a cabo dado un rol de usuario particular, información espacio-temporal, los diferentes estados de ánimo que el usuario experimenta dadas ciertas situaciones, entre otros. (ii) *Información Demográfica*: este perfil contiene información demográfica que se puede obtener a través de fuentes de datos externas (por ejemplo, Internet), o deducir a partir de la información sobre los usuarios que se encuentra dentro de los *Perfiles de Usuario*. Se propone que los *Perfiles Contextuales* que pueden ser asociados con un *Item Profile* son los que contienen información contextual acerca de los productos dadas tres perspectivas: el *dominio*, la *categoría* y el *tipo*. Además, estos perfiles (*perfil de Dominio*, *Categoría* y *Tipo*) tienen asociaciones entre sí que se puede definir en el componente *Información de Asociación y Reglas*. Cada uno de los perfiles propuestos puede ser extendido y más *Perfiles Contextuales* podrían ser agregados al modelo.

## 2.2 Recommendation Engine

Finalmente, la *Figura 4* muestra una vista detallada de la capa *Recommendation Engine*.



**Figura 4.** Visión detallada de la capa *Recommendation Engine*  
Esta capa se compone de los siguientes componentes:

- *Delivery and Presentation*: este componente es el encargado de desplegar los resultados de recomendación (obtenidos del componente de *Recommendation Services*) a los usuarios, con una cierta presentación y a través de un canal de distribución.

- *Evaluation and Upgrade*: evalúa los resultados de recomendación y actualiza los distintos componentes dentro del *Modelo General de Vizier* buscando aumentar el desempeño final. Este es un componente opcional dado que los servicios de recomendación aún se pueden proporcionar si no se encuentra incluido. Sin embargo, el componente representa una adición importante a fin de garantizar la mejora continua de los servicios de recomendación.
- *Recommendation Services*: las *aplicaciones consumidoras* pueden invocar un servicio de recomendación proporcionado por el componente *Recommendation Services* el cual coopera con el componente *Selector*.
- *Selector*: selecciona la técnica de recomendación más apta dadas las características de un determinado *Recommendation Services*. Si sólo hay una técnica de recomendación disponible, entonces la creación de este componente es innecesaria.
- *Recommendation Techniques*: gestiona todas las técnicas disponibles de recomendación (al menos uno).

Este documento se basa en la premisa de que *los Sistemas Multi-agente* son una de las mejores herramientas para la construcción de *Sistemas de Recomendación*. Esta premisa es analizada en la siguiente sección.

## III. AGENTES PARA SISTEMAS DE RECOMEDACIÓN

Los *Sistemas Multi-agente (SMA)* son herramientas adecuadas para desarrollar aplicaciones que requieren tanto de distribución como de inteligencia, como lo son los *Sistemas de Recomendación* [6]. Por un lado, los *Sistemas de Recomendación* tienden a encontrarse en entornos distribuidos (por ejemplo, *WWW*) donde los datos pueden ser obtenidos de diversas fuentes. No sólo son los *Sistemas de Recomendación* generalmente distribuidos; también tienden a ser grandes, descentralizados y dinámicos. Por otro lado, las cualidades especiales que los agentes poseen pueden ser aprovechadas de manera diferente y creativa para diseñar múltiples soluciones de recomendación que requieren de inteligencia. Por ejemplo:

- Un agente puede ser utilizado en *Sistemas de Recomendación* para recolectar información del usuario, analizar sus comportamientos pasados y así modelar esta información en perfiles de forma dinámica que se adapten a las preferencias cambiantes de los usuarios. De esta manera, un agente puede representar al usuario en el sistema, y a través de técnicas de cooperación y negociación, recomendar los mejores productos al usuario.
- Agentes especializados en ciertos dominios de recomendación podrían ser diseñados, cada uno con conocimiento experto que les permita recomendar productos sólo para un tipo específico de dominio (por ejemplo, música, películas,

documentos de investigación, entre otros) y apoyándose en técnicas de recomendación diferentes (por ejemplo, algunos podrían hacer recomendaciones con un enfoque basado en contenidos y otros con un enfoque de filtrado colaborativo).

- Un ecosistema de agentes especializados basados en algoritmos genéticos pueden competir para ofrecer las mejores recomendaciones a los usuarios [12]. Por lo tanto, como el ecosistema evoluciona también lo hace la calidad de las recomendaciones ofrecidas [12].

En resumen, los SMA podrían llegar a ser instrumentos eficaces que se pueden utilizar para diseñar varias soluciones

para el problema de recomendación.

Con la guía que ofrece el *Modelo General de Vizier* y el uso de un enfoque orientado a SMA, se obtuvo el diseño inicial para el *Framework*. La siguiente sección describe este diseño.

#### IV. VIZIER

En esta sección se explica en detalle el diseño inicial del *Framework Vizier* (ver *Figura 5*). En concreto, *Vizier* es:

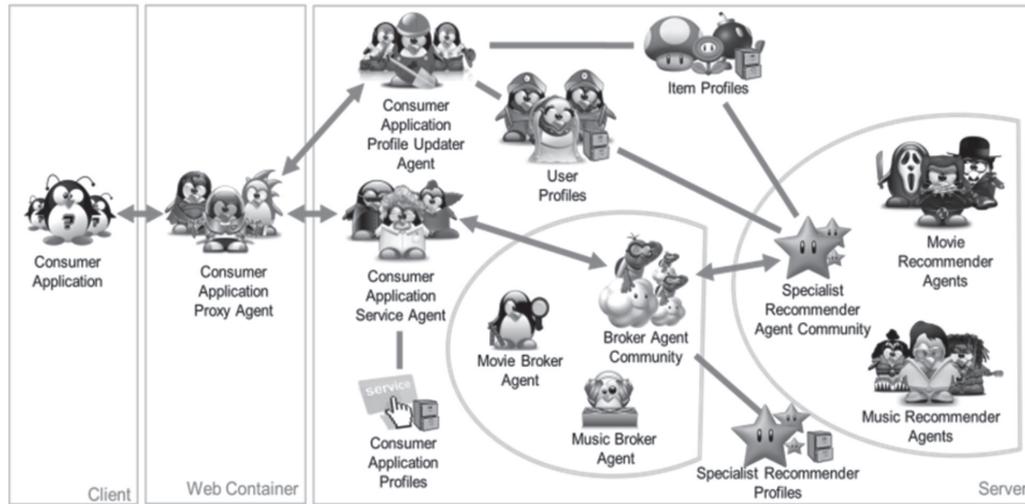


Figura 5. Multi-Agent Vizier Recommendation Framework

- *Multidimensional*: lo que significa que el framework puede considerar múltiples dimensiones de *Adaptación/Personalización* (por ejemplo, la dimensión usuario, contexto, entre otros) para construir una recomendación. Incluir información contextual puede ser crucial para ofrecer recomendaciones en algunas aplicaciones, y tomando en cuenta dimensiones contextuales tales como el tiempo (por ejemplo, la temporada del año, el mes, el día), compañía (es decir, la persona con la que el producto se consume) y el lugar; puede mejorar significativamente la calidad de las recomendaciones ofrecidas [1].
- *Genérico*: significa que el framework se puede ajustar para ofrecer servicios a diversas *aplicaciones consumidoras* (es decir, la aplicación orientada a la recuperación de información que es capaz de invocar a *Vizier* para proporcionar servicios de recomendación) las cuales pueden ofrecer productos pertenecientes a diversos dominios (por ejemplo, multimedia), categorías (por ejemplo, música) y/o tipos (por ejemplo, salsa). El hecho que las técnicas de recomendación existentes se encuentran en su mayoría estrictamente relacionadas con el tipo de producto para el cuál el sistema fue diseñado, crea la necesidad de diseñar un framework que ofrezca recomendaciones de manera genérica

y de esta forma ofrecerle a desarrolladores un mejor servicio de apoyo [16][17].

El diseño de *Vizier* se divide en tres ambientes: cliente, contenedor *web* y servidor. Los agentes de plataforma de *Vizier* residen en el servidor. El cliente representa el *Consumer Application*, la cual solicita servicios de recomendación y de actualización de perfiles a través de su agente *Consumer Application Proxy Agent* (alojado en un contenedor *web*). En las siguientes secciones se explican los diferentes componentes que se encuentran en *Vizier*, y cómo el framework responde a las solicitudes de servicio de las *aplicaciones consumidoras*.

##### 4.1 Consumer Application Proxy Agent

El *Consumer Application Proxy Agent*, alojado en un contenedor *web*, actúa como un intermediario entre el *Consumer Application* y los agentes de plataforma de *Vizier*. Un *Consumer Application Proxy Agent* es creado y adaptado a cada *Consumer Application* registrado en *Vizier* (un *Consumer Application* debe registrarse primero para que en *Vizer* se pueda crear un ambiente adecuado para manejar las diversas solicitudes de servicio), y responder a las solicitudes de servicio (incluso si requieren de diferentes servicios de recomendación) generadas a partir de cualquier usuario del *Consumer Application*. De esta manera, el *Consumer Application Proxy Agent* debe

garantizar una comunicación transparente entre el *Consumer Application* y *Vizier* dadas las propiedades de conexión únicas (por ejemplo, el punto de entrada del dispositivo, tipo de conexión, tipo de red, protocolos de comunicación) que pudiesen surgir de su interacción. Además, el *Consumer Application Proxy Agent* es móvil y puede desplazarse entre diversos contenedores *web* si es necesario o puede clonarse en caso de sobrecarga.

#### 4.2 Comunidad de *Specialist Agent*

*Vizier* alberga una comunidad de agentes *Specialist Recommender Agents*, cada uno especializado en las diversas técnicas de recomendación disponibles en el framework. Cada técnica requiere de ciertas entradas de datos y ofrece diferentes servicios de recomendación. Por otra parte, el *Specialist Recommender Agent* puede acceder a la información de los perfiles de usuario y de productos con el fin de construir una recomendación. Además, dado el hecho de que no todas las técnicas de recomendación son apropiadas para todas las situaciones, un agente *Specialist Recommender Agent* se dice que es experto en ofrecer recomendaciones en el contexto de ciertos dominios, categorías y/o tipos. La información que describe las características del *Specialist Recommender Agent* y los requisitos que necesita para ofrecer adecuadamente recomendaciones, se encuentra contenida en un perfil denominado *Specialist Recommender Profile*.

#### 4.3 Comunidad de *Broker Agent*

Dentro de los agentes de plataforma de *Vizier*, reside una comunidad de agentes *Broker Agent*. En la comunidad, existe un agente, el *Broker Agent* para cada dominio, categoría y/o tipo que tiene un *Specialist Recommender Agent* asociado (por ejemplo, *Music Broker Agent*, *Movie Broker Agent* – ver la sección 4.2). Los *Broker Agents* pueden cooperar con los *Consumer Application Service Agents* (ver sección 4.3.2.2) con el fin de ofrecer servicios de recomendación. Un *Consumer Application Service Agent*, al recibir una solicitud de servicio de recomendación, solicita los servicios del *Broker Agent* apropiado que a su vez responde con un resultado de la recomendación. A fin de responder, el *Broker Agent* (que actúa como un componente *Selector*) debe identificar al analizar la información dentro de los *Perfiles de Specialist Recommender*, el *Specialist Recommender Agent* apropiado que pueda responder a la solicitud de servicio de recomendación. El *Broker Agent* le asigna al *Specialist Recommender Agent* la tarea de responder a la solicitud de servicio de recomendación. Cuando se obtienen resultados de recomendación del *Specialist Recommender Agent*, el *Broker Agent* le responde al *Consumer Application Service Agent*. Trabajos futuros deben estudiar con más detalle el proceso de elegir el *Specialist Recommender Agent* apropiado, y posiblemente, definir un sistema de confianza y reputación en que el *Broker Agent* puede ordenar *Specialist Recommender Agents* de acuerdo a resultados de evaluación del servicio de recomendación.

#### 4.3.1 Profile Set

El conjunto inicial de perfiles en *Vizier* contiene cuatro perfiles: *User Profile*, *Item Profile*, *Perfil de la Consumer Application* y *Perfil de Specialist Recommender*. En esta fase inicial del diseño estos perfiles no se encuentran definidos en detalle. No obstante, algunos datos que estos perfiles deben incluir son los siguientes:

- *Perfil de la Consumer Application*: este perfil debe contener información sobre el *Consumer Application* asociada, el tipo de servicio de recomendación que necesita, en qué formato deben retornarse los resultados, qué tipo de información está disponible con el fin de construir una recomendación, y el *Broker Agent* asociado.
- *Perfil de Specialist Recommender*: este perfil debe contener información acerca del *Specialist Recommender Agent* asociado, sobre las entradas de datos que necesita el agente, el tipo de recomendación que proporciona el agente, el formato en que se retornan los resultados, y los dominios, categorías y/o tipos en el cuál el agente es un experto.
- *Perfiles de Usuario y Producto*: por un lado, los *perfiles de usuario* deben contener información relevante de los usuarios, sus gustos, preferencias, intereses, necesidades, entre otros. Por otro lado, un *Item Profile* contiene información que describe un producto. Tanto los perfiles de usuario como los de producto pueden asociarse a *perfiles contextuales*.

Trabajos futuros deben llevarse a cabo para definir de manera detallada la información que deben contener los diferentes perfiles, y cómo esta información será representada en el framework.

#### 4.3.2 Solicitud de Servicio

El *Consumer Application* puede solicitar dos tipos de servicios a través del *Consumer Application Proxy Agent*: solicitud para la actualización de un perfil y solicitud de servicio de recomendación. El *Consumer Application Proxy Agent* en consecuencia envía estas solicitudes de servicio al *Consumer Application Profile Updater Agent* (que es el componente *Actualizador de Perfil*) o al *Consumer Application Service Agent*. Estos agentes ambos deben responder al *Consumer Application Proxy Agent*, quién envía las respuestas del servicio al *Consumer Application*. En las siguientes secciones se explica de manera detallada cómo el *Consumer Application Profile Updater Agent* y el *Consumer Application Service Agent* responden a las solicitudes de servicio.

##### 4.3.2.1 Consumer Application Profile Updater Agent

El *Consumer Application* solicita un servicio de actualización de un perfil cuando desea actualizar su perfil dentro de *Vizier* con nueva información. Cuando una solicitud de actualización de perfil se envía el *Consumer Application Proxy Agent*, éste envía la información al *Consumer Application Profile Updater Agent*. Un *Consumer Application Profile Updater Agent* se ha creado y adaptado a cada aplicación registrada en *Vizier*. Este

agente se encarga de actualizar los perfiles de usuario y perfiles producto de acuerdo con la información de actualización recibida.

#### 4.3.2.2 Consumer Application Service Agent

Cuando una solicitud de servicio de recomendación es enviada por el *Consumer Application* al *Consumer Application Proxy Agent*, el *Consumer Application Proxy Agent* reenvía esta petición al *Consumer Application Service Agent* correspondiente. En el diseño inicial del *Vizier*, un *Consumer Application Service Agent* se ha creado y adaptado a cada servicio de recomendación que un *Consumer Application* puede solicitar. Por ejemplo, si un servicio de música solicita la recomendación de una lista de reproducción inteligente de canciones y en una segunda solicitud requiere de la recomendación de una canción para una lista ya existente, estas solicitudes son manejadas por servicios de recomendación diferentes. Cada servicio de recomendación debe estar asociado a un determinado dominio, categoría y/o tipo, y de esta manera, la respuesta del servicio apropiado puede ser proporcionada. El *Consumer Application Service Agent* puede tener acceso al *Perfil de la Aplicación Consumidor* asociada, donde el *Broker Agent* que debe ser utilizado se ha definido (este agente se especifica en función del dominio, categoría y/o tipo de servicio

de recomendación). El *Consumer Application Service Agent* utiliza los servicios proporcionados por el *Broker Agent* relacionado, con el fin de responder a un servicio de recomendación. Al solicitarle un servicio de recomendación al *Broker Agent*, el *Consumer Application Service Agent* deberá informarle al *Broker Agent* las características y exigencias especiales de los resultados esperados de recomendación (por ejemplo, si una recomendación debe considerar un determinado atributo contextual o no) que se pueden encontrar definidas en el *Perfil de la Aplicación Consumidor*. Cuando el *Broker Agent* le envía los resultados de recomendación al *Consumer Application Service Agent*, éste le envía los resultados al *Consumer Application Proxy Agent* que a su vez se los envía a el *Consumer Application*.

La siguiente sección describe a *ZoundBeat*, un reproductor de música que utiliza los servicios prestados por *Vizier*.

## V. ZOUNDBEAT

*ZoundBeat* (ver *Figura 6*) es un reproductor de música creado específicamente para validar el framework *Vizier*.



Figura 6. ZoundBeat

Con el fin de percibir el valor añadido de *Vizier*, *ZoundBeat* no puede ser tan sólo un reproductor de música común y debe cumplir con un cierto número de características adicionales para que *Vizier* sea capaz de considerar características contextuales del usuario en la construcción de las recomendaciones. De esta manera, *ZoundBeat* debe cumplir con los requisitos típicos de un reproductor de música, y, además, debe ofrecer los medios para capturar información contextual y proporcionar

servicios de recomendación con la ayuda de *Vizier*. Por esta razón, *ZoundBeat* captura información contextual ingresada por el usuario de manera explícita de los estados de ánimo y las actividades que el usuario asocia a la sesión actual, las listas de reproducción y las canciones. Al compartir esta información con *Vizier*, *Vizier* y *ZoundBeat* colaboran para ofrecerles a los usuarios recomendaciones de listas de reproducción.

Se decidió después de un análisis cuidadoso que *ZoundBeat* recomienda una lista de reproducción de canciones basada en la información contextual del usuario y un límite de tiempo especificado por el usuario al solicitar la recomendación. También se decidió que *ZoundBeat* sólo capturara datos explícitos por parte del usuario en el primer prototipo. Entonces, el usuario tiene la posibilidad de modificar la información de las canciones e información contextual. La información contextual que se toma en cuenta incluye las actividades y estados de ánimo (ver *Figura 7*) que el usuario puede asociar a la sesión actual, listas de reproducción y canciones. Los estados de ánimo que el usuario puede seleccionar son: muy feliz, feliz, serio, triste y muy triste. Las actividades que el usuario puede seleccionar son: de rumba, relajándose, durmiendo, trabajando y ejercitándose.

Dentro de *ZoundBeat*, el usuario puede introducir información del estado de ánimo y de la actividad que asocia a cada canción, así como seleccionar una calificación representada por una escala de uno a cinco (siendo cinco la puntuación más alta) para cada canción. Por último, el usuario puede seleccionar el estado de ánimo y las actividades con las cuales él/ella se siente más identificado para la sesión actual. Información contextual de todas las canciones y de la sesión del usuario se enviará a *Vizier* que utiliza esta información para actualizar los perfiles y genera recomendaciones sensibles al contexto de listas de reproducción. Cuando *ZoundBeat* recibe una recomendación de *Vizier*, la aplicación le despliega la lista de reproducción al usuario.

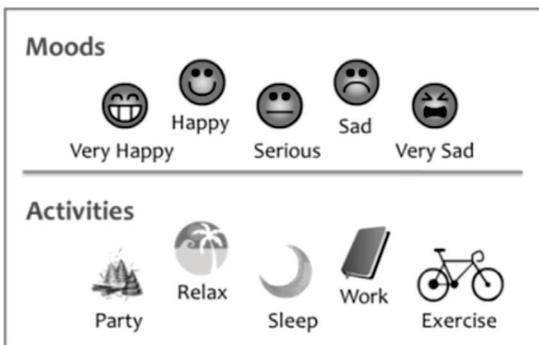


Figura 7. Iconos que representan los estados de ánimo y actividades en *ZoundBeat*

VI. RESUMEN DE IMPLEMENTACIÓN

La implementación inicial de *Vizier* sólo se centra en un subconjunto de los agentes, como se muestra en la *Figura 8*. Cada agente del diseño Multi-agente de la *Figura 8* se puede mapear a un agente del diseño inicial de *Vizier* de la siguiente manera: el *ZoundBeat Proxy Agent* es un *Consumer Application Proxy Agent*, el *ZoundBeat Profile Updater Agent* es un *Consumer Application Profile Updater Agent*, el *ZoundBeat Service Agent* es un *Consumer Application Service Agent*, y el *Music Recommender Agent* es un *Specialist Recommender*

*Agent*. Dentro del prototipo inicial, el conjunto de perfiles se compone de un *User Profile*, un *Contextual Profiles de usuario-canción* y *perfiles de canción*. El *User Profile* se asocia al *Contextual Profiles de usuario-canción* que incluirá toda la información contextual obtenida de las entradas del usuario por *ZoundBeat* para las canciones. Habrá un *perfil de canción* para cada canción que existe en la biblioteca de música en *ZoundBeat*. Cada *perfil de canción* contendrá información de las canciones, tales como: nombre de la canción, artista, nombre del álbum, entre otros.

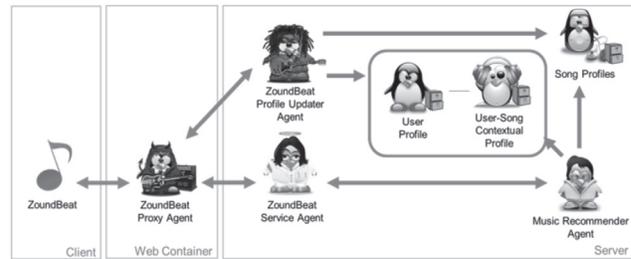


Figura 8. Diseño inicial de *Vizier* para el prototipo

*ZoundBeat* y *Vizier* se llevaron a cabo utilizando las herramientas de desarrollo y entornos de desarrollo integrado que se muestran en la *Figura 9*.

	ZoundBeat	Multi-Agent Vizier Recommendation Framework
Development Tools	Adobe Air + Adobe Flex SDK + ActionScript	JAVA JDK + BESA + Hibernate JPI
Integrated Development Environment	Flash Builder	Netbeans

Figura 9. Herramientas de Desarrollo y Entornos de desarrollo para *ZoundBeat* y *Vizier*

Por un lado, *ZoundBeat* se ha desarrollado utilizando *Adobe Flex* [8], que es un framework de código abierto que permite la creación de aplicaciones Rich Internet Applications (RIA) basadas en *Adobe Flash* y por lo tanto, el lenguaje de programación *ActionScript*. Además, para que la aplicación pudiese ejecutarse como una aplicación cliente independiente, se utilizó el entorno de ejecución *Adobe Air* que permite que *Adobe Flash* funcione como una aplicación cliente de escritorio en cualquier ordenador personal con un sistema operativo de Windows, Linux o Mac que contenga *Adobe Air* instalado [8] (las pruebas sólo se llevaron a cabo en Windows). Al contrario de una aplicación RIA típica, las aplicaciones de *Adobe AIR* requieren instalación y por lo tanto, tienen menos restricciones de seguridad y pueden acceder y modificar los archivos locales. Esto es una ventaja importante porque *ZoundBeat* guarda información de la biblioteca de música e información del usuario dentro de

**un archivo local. No obstante, la migración de ZoundBeat a una aplicación web** puede ser una característica deseable en el futuro, lo que trae a la luz una importante ventaja de la utilización de *Adobe Flex*: **estas aplicaciones se pueden migrar a la web.**

Por otro lado, *Vizier* se ha desarrollado utilizando el lenguaje de programación *Java* y el framework para desarrollo de agentes *BESA* [9]. Los agentes de plataforma de *Vizier* residen en un servidor, mientras que el *Consumer Application Proxy Agent* reside en un contenedor web *Apache Tomcat*, ya sea en el mismo servidor o en otro distinto. Dentro del contenedor web mencionado, se coloca el framework *OpenAMF*. Este framework permite la comunicación remota entre *Adobe Flash* y *Java*.

Finalmente, los agentes de plataforma de *Vizier* guardan la información de los perfiles en una base de datos *PostgreSQL*. Con el fin de acceder y modificar la base de datos, se utiliza el framework de código abierto *Hibernate JPA*. La siguiente sección concluirá al mostrar los resultados obtenidos del proceso de implementación.

## VII. PRUEBAS Y RESULTADOS

Las pruebas aplicadas a *ZoundBeat* y *Vizer* en su desarrollo inicial fueron un éxito: ambos fueron probados por un grupo de usuarios que aportaron diversa información contextual y se obtuvieron resultados diferentes y adaptados. Los usuarios también probaron *ZoundBeat* y *Vizer* sin la opción de considerar aspectos contextuales (actividades, estado de ánimo) y no quedaron tan satisfechos con los resultados obtenidos.

La característica *Multidimensional* de *Vizier* fue validada a través de la implementación de un *Music Recommender Agent* especialmente diseñado para considerar las entradas de datos que *ZoundBeat* puede ofrecer, y de esta manera, *Vizier* es capaz de ofrecer junto con *ZoundBeat* recomendaciones de canciones sensibles al contexto. Es importante señalar que la técnica actual de recomendación utilizada por el agente *Music Recommender Agent* no está siendo probada ni validada de ninguna manera. Lo que fue importante demostrar es que *Vizier* podría ofrecer apoyo a una técnica de recomendación que considerará múltiples dimensiones de *Adaptación/Personalización*. En trabajos futuros, se debe llevar a cabo una investigación más profunda en el tema de recomendación de música para elaborar y aplicar adecuadamente una técnica de recomendación para *ZoundBeat*. Adicionalmente, es necesario contar con un conjunto de datos de música que contenga información contextual. Es importante recordar que la falta de un conjunto de datos apropiados es un gran obstáculo para cualquier proyecto en el tema de los *Sistemas de Recomendación* y son necesarios ciertos trabajos futuros para encontrar o construir dicho conjunto de datos.

La característica *genérica* de *Vizier* no pudo ser validada

aún. Trabajos futuros se deben llevar a cabo para crear o encontrar otras *aplicaciones consumidoras* que puedan utilizar *Vizier* para ofrecer recomendaciones a los usuarios. Además, el prototipo inicial de *Vizier* debe ser ampliado para incluir *Broker Agents*, más *Specialized Recommender Agents*, entre otras características necesarias para apoyar a *aplicaciones consumidoras* adicionales.

## VIII. TRABAJOS RELACIONADOS

La *Tabla 1* presenta una comparación clara entre los trabajos relacionados que fueron estudiados. Dentro de la tabla, el símbolo ‘+’ indica que los criterios se cumplen, el símbolo ‘-’ indica que los criterios no se cumplen, y el símbolo ‘?’ indica que no es claro si los criterios se cumplen. Desde una perspectiva general de las obras que trabajan en *Sistemas de Recomendación* genéricos surgió una observación particular: la cualidad genérica para *Sistemas de Recomendación* se puede lograr en dos niveles. Estos niveles se identifican de la siguiente manera: (a) *independencia del dominio*: el framework de recomendación puede ser ajustado para ofrecer servicios en diferentes dominios (por ejemplo, en entretenimiento, multimedia) (b) *independencia de la categoría*: el framework de recomendación puede ser ajustado para ofrecer servicios en diferentes categorías que pertenecen al mismo dominio (por ejemplo, un framework de recomendación que está especializado en el dominio de multimedia y puede ofrecer recomendaciones en las categorías de música, videos, televisión, entre otros). Estos atributos se toman en cuenta como criterios de comparación en la *Tabla 1*.

**Tabla 1.** Comparación de Trabajos Relacionados con Vizier (V).

	[10]	[16]	[14]	[15]	[7]	[2]	[5]	[3]	V
INDEPENDENCIA DEL DOMINIO	-	+	+	-	+	+	?	+	+
INDEPENDENCIA DE LA CATEGORÍA	+	?	?	+	?	?	?	?	+
MULTI-ALGORITMO	-	+	-	+	+	?	-	?	+
MULTIDIMENSIONAL	+	+	-	-	-	+	+	+	+

De la *Tabla 1*, se pueden obtener las siguientes conclusiones:

- A pesar de que la independencia al dominio y la independencia a la categoría ambas son características deseadas, la mayoría de trabajos se centran sólo en el desarrollo de sistemas que ofrecen independencia al dominio. Estos trabajos tienden a concentrarse en los atributos del dominio, y a olvidarse de los atributos relacionados con la categoría. Del mismo modo, si un trabajo en particular se centra en proporcionar independencia a la categoría, entonces se supone que estas categorías pertenecen al mismo dominio, y de esta manera, la información del dominio se ignora.
- Los trabajos que utilizan múltiples algoritmos encontraron que esta estrategia podría ser útil al ofrecer servicios de

recomendación para diferentes tipos de escenarios.

- Muchos de los trabajos relacionados han reconocido el potencial de ofrecer servicios sensibles al contexto y han propuesto diferentes mecanismos para realizar esta tarea. Sin embargo, se debe desarrollar más trabajo en esta área con el fin de obtener mejoras significativas en las técnicas de recomendación.
- A través de *Vizier* todos los criterios se pueden cumplir. Sin embargo, una validación más completa del framework propuesto debe llevarse a cabo.

## IX. CONCLUSIONES Y TRABAJOS FUTUROS

Este artículo presenta a *Vizier*, un framework genérico basado en agentes que considera información contextual para ofrecerle a diversas aplicaciones orientadas a la recuperación de información servicios de recomendación para que a su vez, puedan ofrecerles recomendaciones a sus usuarios. Tomando en cuenta que el framework se desarrolló bajo un enfoque genérico, sus servicios están orientados a beneficiar a cualquier entidad que desee ofrecer servicios de recomendación a sus usuarios (por ejemplo, aplicaciones de comercio electrónico). Además, tomando en cuenta múltiples dimensiones de *adaptación/personalización*, es de esperar que la solución propuesta ofrezca recomendaciones que son mucho más interesantes y útiles para el usuario que las proporcionadas por las soluciones actuales. De manera adicional, fue creado el reproductor de música *ZoundBeat* para actuar como una *Consumer Application* para *Vizier*; y de esta manera, ayudar a validar el framework. *ZoundBeat* es capaz de invocar los servicios de recomendación del framework para ofrecer a sus usuarios servicios de recomendación de canciones que son sensibles al contexto.

Dadas las restricciones de tiempo, se obtuvieron un gran número de logros; sin embargo algunos trabajos futuros son necesarios para sobrellevar las restricciones actuales:

- *Actualización constante del marco teórico y el estado del arte*: una investigación más extensa debe llevarse a cabo con el fin de profundizar en los diferentes conceptos relacionados con los *Sistemas de Recomendación* encontrados en la investigación actual. Estudios futuros deben explorar en profundidad las técnicas de recomendación para los diferentes servicios de recomendación que podría proporcionar *Vizier*. Además, la relación entre los diferentes dominios, categorías y tipos, también deben ser estudiados con el fin de realmente ofrecer servicios genéricos. Como paso inmediato, una investigación debe llevarse a cabo en el tema de recomendación de música y de esta manera añadir un *Specialist Recommender Agent* a *Vizier*.
- *Creación de un dataset que tome en cuenta diversos dominios*:

inicialmente el dataset que se está utilizando pertenece a la categoría de música. Sin embargo, con el fin de validar que el framework es realmente genérico, éste debe ofrecer servicios de recomendación a *aplicaciones consumidoras* que pertenezcan a distintos dominios, categorías y/o tipos. Esto significa que el framework no sólo debe contar con los agentes *Specialist Recommender Agent*; también necesita un dataset que tome en cuenta diversos dominios o varios dataset que pertenezcan a distintos dominios, categorías y/o tipos. Dado que los dataset existentes no incluyen información del contexto, el cumplimiento de los requisitos de dataset implica la creación de un nuevo(s) dataset(s).

- *Incorporar más de un Consumer Application*: dado que *Vizier* ha sido diseñado bajo una aproximación genérica, pruebas al prototipo deben ser llevadas a cabo utilizando diferentes aplicaciones consumidoras de diferentes dominios, categorías y/o tipos.
- *Perfiles Genéricos*: que deben ser agregados a *Vizier* para realmente ofrecer servicios genéricos.
- *Enfrentar los retos actuales de los Sistemas de Recomendación*: *Vizier* también enfrenta los retos actuales de los Sistemas de Recomendación (por ejemplo, escalabilidad) y trabajos futuros necesitan ser llevados a cabo para determinar cómo estos retos serán enfrentados.

## AGRADECIMIENTOS

El presente trabajo fue desarrollado bajo el marco del proyecto: “*Asma: Modelo de Adaptación Contextual en Sistemas MultiAgentes*”, apoyado por la Pontificia Universidad Javeriana, Bogotá, Colombia.

## REFERENCIAS

- [1] Adomavicius, G. Tuzhilin, A. 2005. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. En: IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, Junio 2005.
- [2] Adomavicius, G., Sankaranarayanan, R., Sen, S., Tuzhilin, A. 2005. Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach. En: ACM Transactions on Information Systems TOIS, vol. 23, no. 1, Enero 2005. ACM, NY, USA, 2005, pp. 103–145.
- [3] Anand, S., Mobasher, B. 2006. Contextual Recommendation. En: Berendt, B., Hotho, A., Mladenic, D., Semeraro, G. eds: Proc. of Workshop on Web Mining, From Web to Social Web: Discovering and Deploying User and Content Profiles WebMine 2006. Berling, Alemania, Septiembre 18, 2006. LNCS 4737, Springer-Verlag, Berling, Heidelberg, 2007, pp. 142-160
- [4] Ansari, A., Essegai, S., Kohli, R. 1999. Internet Recommendation Systems. Columbia University, New York, USA, 1999, 44 páginas.
- [5] Baltrunas, L. 2008. Exploiting Contextual Information in

- Recommender Systems. En: Pu, P., Bridge, D., Mobasher, B., Ricci, F. eds.: Proc. of the 2008 ACM Conference on Recommender Systems RecSys 2008. Lausanne, Switzerland, Octubre 23-25, 2008. ACM, NY, USA, 2008, pp. 295-298.
- [6] Barraza, A. 2011. Design of a Generic Multidimensional System that Provides Personalized Recommendation Services. Pontificia Universidad Javeriana, Bogota, Colombia, 2011, 80 páginas. <http://pegasus.javeriana.edu.co/~CIS1010IS01/>
- [7] Corizo, J., Carrero, F., Monsalve, B. 2010. An Architecture for a General Purpose Multi-Algorithm Recommender System. eds: Proc. of the Workshop on the Practical Use of Recommender Systems, Algorithms and Technologies, PRSAT 2010. Barcelona, España, Septiembre 30, 2010.
- [8] Deliver engaging applications beyond the browser across screens: <http://www.adobe.com/products/air/>. Último Acceso: Mayo 2011.
- [9] González, E., Bustacara, C. 2007. Desarrollo de aplicaciones basadas en sistemas Multi-Agentes. Pontificia Universidad Javeriana, Bogotá.
- [10] Kamimada, N., Tomohiro, T., Hoshino, M. 2009. Cross-category Recommendation for Multimedia Content. En: Furht, B. eds: Handbook for Digital Entertainment and Arts, vol. 1, Springer, US, 2009, pp. 27-57.
- [11] Konstan, J. 2008. Introduction to Recommender Systems. eds.: Proc. of the 2008 ACM SIGMOD International Conference on Management of Data, Tutorial Session, SIGMOD 2008. Vancouver, Canada, Junio 9-12, 2008. ACM, NY, USA.
- [12] Montaner, M., López, B., Lluís de la Rosa, J. 2003. A Taxonomy of Recommender Agents on the Internet. En: Artificial Intelligence Review, vol. 19, no. 4, 2003. Kluwer Academic Publishers, Netherlands, 2003, pp. 285-330.
- [13] Mukherjee, R., Sajja, N., Sen, S. 2003. A Movie Recommendation System-An Application of Voting Theory in User Modeling. En: User Modeling and User-Adapted Interaction, vol. 13, no. 1-2, Febrero 2003. Kluwer Academic Publishers, Hingham, MA, USA, pp. 3-33.
- [14] Patel, A., Balakrishnan, A. 2009. Generic framework for recommendation system using collective intelligence. eds: Proc. of the 4<sup>th</sup> Int. Conf. for Internet Technology and Secured Transactions, ICITST 2009. London, UK, Noviembre 9-12, 2009. IEEE, Washington, USA, pp. 1-6.
- [15] Qiu, Z., Mingrui, C. and Jun, H. 2010. Design of Multi-mode E-commerce Recommendation System. In: eds: Proc. of the 3<sup>rd</sup> Int. Symposium on Intelligent Information Technology and Security Informatics IITSI 2010. Jinjiangshan, China, Abril 2-4, 2010. IEEE, Washington, USA, 2010, pp. 530-533.
- [16] Räck, C., Arbanowski, S., Steglich, S. 2007. A Generic Multipurpose Recommender System for Contextual Recommendations. eds.: Proc. of the 8<sup>th</sup> International Symposium on Autonomous Decentralized Systems, ISADS 2007. Sedona, Arizona, USA, Marzo 21-23, 2007, IEEE Computer Society, Washington, USA, 2007, pp. 445-450.
- [17] Räck, C., Arbanowski, S., Steglich, S. 2006. Context-aware, Ontology-based Recommendations. eds.: Proc. of the International Symposium on Applications and the Internet Workshops SAINT Workshops 2006. Phoenix, Arizona, USA, Enero 23-27, 2006, IEEE Computer Society, Washington, DC, USA, 2006, pp. 98 – 104.
- [18] Resnick, P., Varian, H. Recommender Systems. 1997. En: Communications of the ACM, vol. 40, no. 3, Marzo 1997. ACM, New York, NY, USA, pp. 56-58.
- [19] Schafer, J., Konstan, J., Riedl, J. 2001. E-Commerce Recommendation Applications. En: Data Mining and Knowledge Discovery, vol. 5, no. 1-2, Enero-Abril 2001. Kluwer Academic Publishers Hingham, MA, USA, pp. 115-153.
- [20] Terveen, L., Hill, W. 2001. Beyond Recommender Systems: Helping People Help Each Other. En: Carroll, J. eds.: HCI in the New Millennium, no. 1, 2001. Addison-Wesley, pp. 487–509.



Cisco | Networking Academy®  
Mind Wide Open™

Cisco Networking Academy es un programa ampliamente conocido de e-doing que enseña a los estudiantes las habilidades tecnológicas de Internet en una economía global. El programa proporciona contenido basado en la Web, pruebas en línea, seguimiento del desempeño de los estudiantes, laboratorios con equipos reales y con simuladores, soporte y entrenamiento por parte de los instructores, así como preparación para las certificaciones estándares de la industria.



#### Oferta de cursos

- ✓ Mantenimiento de PC: IT Essentials
- ✓ Redes básicas: Cisco Certified Network Associate
- ✓ Redes avanzadas: Cisco Certified Network Professional
- ✓ Seguridad en routers: CCNA Security
- ✓ Voz sobre IP
- ✓ Asterisk básico

#### Programación 2011

Ciclo 48: Inicia 17 de enero. Finaliza 12 de marzo  
 Ciclo 49: Inicia 22 de marzo. Finaliza 23 de mayo  
 Ciclo 50: Inicia 30 de mayo. Finaliza 29 de julio  
 Ciclo 51: Inicia 8 de agosto. Finaliza 3 de octubre  
 Ciclo 52: Inicia 10 de octubre. Finaliza 12 de diciembre



Consulte los horarios de cada nivel a través de nuestros canales informativos al pie de página



#### Además...

- ✓ Alquiler de laboratorios virtuales para auto-estudio o cursos empresariales
- ✓ Presentación de exámenes de certificación para múltiples áreas bajo el Centro Pearson VUE
- ✓ Cursos exclusivos para su empresa
- ✓ Pregunte por nuestros descuentos

#### CATC - Academia Regional - Academia Local

Universidad Nacional de Colombia sede Medellín  
 Calle 65 78-28 Bloque M1 Oficina 101. Facultad de Minas

Teléfono: +57 4 4255268 Fax: +57 4 2341002 E-mail: [catc@unal.edu.co](mailto:catc@unal.edu.co) Web: <http://cnap.unalmed.edu.co>

Facebook: [fb.me/catcunal](https://www.facebook.com/catcunal) Twitter: [@catcunal](https://twitter.com/catcunal) Buzz: [google.com/profiles/catcunal](https://www.google.com/profiles/catcunal)

Medellín, Colombia

