



UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
La Universidad Católica de Loja

ÁREA TÉCNICA

**TITULACIÓN DE INGENIERO EN SISTEMAS INFORMÁTICOS Y
COMPUTACIÓN**

**Agente recomendador para dar soporte al estudiante en el desarrollo de
actividades de aprendizaje.**

Trabajo de fin de Titulación

AUTOR: Berrú Ramírez, Freddy Eduardo

DIRECTORA: Valdiviezo Díaz, Priscila Marisela, Ing.

LOJA-ECUADOR

2013

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE FIN DE TITULACIÓN

Mgs.

Priscila Marisela Valdiviezo Díaz

DOCENTE DE LA TITULACIÓN

De mi consideración:

El presente trabajo de fin de titulación: “Agente Recomendador para dar soporte al estudiante en el desarrollo de actividades de aprendizaje” realizado por Berrú Ramírez Freddy Eduardo ha sido orientado y revisado durante su ejecución, por se aprueba la presentación del mismo.

Loja, diciembre del 2013

f).....

DECLARACIÓN DE AUTORIA Y CESION DE DERECHOS

“Yo Berrú Ramírez Freddy Eduardo declaro ser autor del presente trabajo de fin de titulación: “Agente Recomendador para dar soporte al estudiante en el desarrollo de actividades de aprendizaje”, de la Titulación Ingeniero en sistemas informáticos y computación, siendo Priscila Marisela Valdiviezo Díaz directora del presente trabajo y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos y acciones legales. Además certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Adicionalmente declaro conocer y aceptar la disposición del Art. 67 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado que se realicen a través, o con el apoyo financiero, académico o institucional (operativo) de la Universidad”

f).....

Autor: Berrú Ramírez Freddy Eduardo

Cedula: 1104069040

DEDICATORIA

A mis padres por ser quienes me ayudan constantemente y han sido un ejemplo a seguir durante toda mi vida.

A mi querida esposa la cual desde que forma parte de mi vida me ha dado su apoyo incondicional.

A mí querido hijo Justin que es la razón de todos mis logros y deseos de superación

AGRADECIMIENTO

En primer lugar a Dios por guiar todos los pasos de mi vida

Un especial agradecimiento a la Ing. Priscila Valdiviezo, ya que con sus directrices se ha podido llegar al cumplimiento de este proyecto.

A la unidad de virtualización por la información facilitada durante todo el proyecto.

A la unidad de Inteligencia artificial de la UTPL y a todos los que la conforman por las facilidades prestadas para el desarrollo de este tema de tesis.

INDICE DE CONTENIDOS

Caratula.....	i
APROBACIÓN DEL DIRECTOR DEL TRABAJO DE FIN DE TITULACIÓN	ii
DECLARACIÓN DE AUTORIA Y CESION DE DERECHOS.....	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
ÍNDICE DE GRÁFICOS.....	viii
ÍNDICE DE TABLAS	ix
RESUMEN.....	1
ABSTARCT	2
INTRODUCCIÓN.....	3
CAPITULO 1: ESTADO DEL ARTE	6
1.1. Sistemas de gestión de aprendizaje (Learning Management System)	7
1.2. Sistemas tutores inteligentes.....	8
Módulo de Comunicación.....	9
Modelo del Estudiante.....	10
Modulo Tutor.....	10
Modulo del Dominio.....	10
1.3. Sistemas de recuperación de información.....	10
Componentes de un sistema de recuperación de información	12
Tipos de sistemas de recuperación de información.....	13
1.4. Sistemas recomendadores.....	13
1.4.1. Clasificación Sistemas Recomendadores según su fuente de conocimiento.....	15
Recomendadores Demográficos	15
Recomendadores Colaborativos:	16
Recomendadores Basados en Contenidos:	16
Recomendadores Basados en Conocimiento:.....	17
1.5. Agentes recomendadores en el campo educativo	17
Elementos del modelo de recomendaciones en un entorno educativo.	19
1.6. Técnicas de recomendación en entornos educativos.	22
1.6.1. Técnicas de Recomendación basadas en modelo	22
1.6.2. Técnicas de recomendación basadas en memoria.....	23
1.7. Anotaciones Semánticas.....	29
1.7.1. Frameworks para anotación semántica.....	30
1.7.2. Herramientas de anotación semántica.	31
1.8. Plataforma KIM	34
Arquitectura de la plataforma KIM	36
KIM API.....	37
1.9. Agentes inteligentes.....	39

1.10. Trabajos relacionados	40
CAPITULO 2: IDENTIFICACIÓN DE TAREAS PENDIENTES EN EL EVA.....	43
2.1. Entorno virtual de aprendizaje en la UTPL	44
2.2. Moodle (Modular Object Oriented Dynamic Learning Environment)	45
2.3. Características de EVA	47
Estructura básica del EVA.....	49
2.4. Base de datos del EVA	52
2.5. Escenario de prueba	53
2.6. Monitoreo de la base de datos mediante la herramienta TOAD	54
2.7. Proceso de identificación de tareas pendientes.....	57
2.8. Elaboración del Bloque de Tareas Pendientes.....	58
2.9. Recursos educativos abiertos	60
CAPITULO 3: DISEÑO Y DESARROLLO DEL AGENTE RECOMENDADOR EN EL EVA DE LA UTPL	63
3.1. Problemática	64
3.2. Diseño de la solución	65
3.3. Objetivos	66
3.4. Esquema del Agente.....	66
3.5. Elementos parte de Recomendación del Agente.....	67
3.5.1. Categoría.....	67
3.5.2. La técnica.....	68
3.5.3. Origen.....	68
3.5.5. Condiciones Aplicables.....	69
3.6. Arquitectura del Agente Recomendador de la UTPL	69
3.7. Desarrollo del Agente recomendador y sus componentes.....	70
3.7.3. Servicios web del agente y sus métodos principales.	74
3.7.4. Agente Anotador WSanotador:	75
3.7.5. appAgenteWS_recomendador.....	76
3.8. Integración del agente recomendador al EVA	77
3.9. Pruebas con el agente recomendador.....	79
3.9.1. Pruebas de Precisión.....	79
3.9.2. Pruebas de Usabilidad.....	80
3.10. Resultados de las pruebas	81
3.10.1. Resultados de la prueba de Precisión.....	81
3.10.2. Resultados de la pruebas de usabilidad.....	82
Conclusiones	84
Recomendaciones	86
Trabajos Futuros.....	87
BIBLIOGRAFIA.....	88
ANEXOS.....	89

ANEXO 1	89
Anexo 2	90
Anexo 3.....	94
Anexo 4	100
Anexo 5	104
Anexo 6	112
Anexo 6	130
Anexo 7	135
Anexo 8	148

ÍNDICE DE GRÁFICOS

Gráfico 1 Componentes de un STI (Álvarez, 2010).....	9
Gráfico 2 Sistema de Recuperación de Información IRIS (Itmazi, 2005).....	12
Gráfico 3 Tipos de agentes Recomendadores y sus fuentes de conocimiento (Burke, 2007).	15
Gráfico 4 Modelo de recomendaciones para escenarios de aprendizaje (Santos & Boticario, 2008)	19
Gráfico 5 Arquitectura de la plataforma KIM (Popov, Kiryakov, Kirilov, Manov, Ognyanoff, & Goranov, 2010).....	36
Gráfico 6 Diagrama de Flujo de Extracción de Información Semántica de KIM (Popov, Kiryakov, Kirilov, Manov, Ognyanoff, & Goranov, 2010)	39
Gráfico 7 Esquema del agente en conclusión con estado del arte	42
Gráfico 8 Arquitectura del entorno virtual de aprendizaje de la UTPL	47
Gráfico 9 Captura de pantalla curso en el EVA.....	48
Gráfico 10 bloque de actividades en Moodle (Moodle, Bloque de Actividades).....	50
Gráfico 11 Escenario de pruebas.....	53
Gráfico 12 registro de actividad en las tablas mdl_assignment_submissions y mdl_forum_discussions	58
Gráfico 13 Modelo entidad-relación del EVA.....	59
Gráfico 14 Bloque de tareas pendientes en el EVA	60
Gráfico 15 Modelo entidad - relación REA de la UTPL (Suarez, 2011)	62
Gráfico 16 Esquema Agente Recomendador UTPL.....	67
Gráfico 17 Arquitectura Agente Recomendador UTPL.....	69
Gráfico 18 tabla de recursos anotados	77
Gráfico 19 Captura del agente recomendador integrado al EVA.....	79
Gráfico 20 Porcentajes de la precisión en cada curso de prueba.....	82
Gráfico 21 Instalación bloque tareas pendientes en el EVA.....	91
Gráfico 22 Colocar el bloque en la página de los cursos.....	92

Gráfico 23 Lista de bloques para agregar a la página de los cursos	92
Gráfico 24 Bloque instalado en el EVA	93
Gráfico 25 Pantalla principal de acceso al EVA	96
Gráfico 26 Pantalla de administración del EVA.....	97
Gráfico 27 bloque de administración del EVA	97
Gráfico 28 Cambio clave para acceder como administrador	98
Gráfico 29 Cambio de fecha del periodo.....	98
Gráfico 30 Cambio de fecha en los cursos.....	99
Gráfico 31 Agregar variables de entorno.....	100
Gráfico 32 Interfaz de usuario de kim	102
Gráfico 33 Entidades en la interfaz de kim.....	103
Gráfico 34 Ontología de la plataforma kim	103

ÍNDICE DE TABLAS

Tabla 1 Filtrado colaborativo basado en usuario.....	25
Tabla 2 Filtrado colaborativo basado en ítems.....	25
Tabla 3 Filtrado colaborativo basado en estereotipos o demográfico.....	25
Tabla 4 Razonamiento basado en casos	26
Tabla 5 Técnica de recomendación basada en atributos	26
Tabla 6 Herramientas de anotación semántica, expectativas dentro del proyecto	33
Tabla 7 Tabla descriptiva del Servicio Anotador	75
Tabla 8 Tabla descriptiva del Servicio web Agente_recomendador	76
Tabla 9 resultados de las pruebas de precisión	81
Tabla 10 Precisión en cada curso de prueba	81

RESUMEN

El presente trabajo se lleva a cabo en el entorno virtual de aprendizaje (EVA) de la UTPL, en donde se ha visto la necesidad de dar soporte a los estudiantes en sus actividades de aprendizaje. Para este propósito se requiere del diseño y la construcción de un agente, que emita recomendaciones de recursos relacionados a las actividades que el estudiante tiene que realizar.

Se analiza el funcionamiento del EVA y el bloque de recursos educativos abiertos REA implementado en el mismo, se han identificado las actividades pendientes del estudiante, información que el agente utiliza y mediante el uso de la técnica de recomendación basada en atributos procesa esta información y da como resultado las recomendaciones al estudiante.

Para integrar esta técnica al agente recomendador se ha utilizado la herramienta web semántica kim, la cual ayuda en el proceso de anotaciones semánticas de los atributos y recuperación de los documentos que van a ser ofertados como recomendación.

En la parte final se ha acoplado el agente recomendador en el entorno mediante la utilización de servicios web y un cliente que consume estos servicios desde el EVA.

Palabras Clave: LMS; Moodle; KIM; IRIS;STI;SR

ABSTARCT

The present project has been developed in the virtual learning environment of UTPL, where it has seen the need have support that have the students in your activities.

For this purpose requires the design and construction of an agent, to issue recommendations about resources related to the activity that the student have to development.

We analyze the performance of EVA and the block REA of open educational resources deployed in this environment, we have identified the remaining activities of the student, the agent uses this information and by using the technique of attribute-based recommendation processes gives as a result the student recommendations.

To integrate this recommender agent technique was used kim semantic web tool, which helps in the process of semantic annotation and retrieval of the documents that will be offered as a recommendation.

In the final part is coupled recommender agent in the environment through the use of web services and a client that consumes these services from the EVA.

Keywords: LMS; Moodle; KIM; IRIS;STI;SR

INTRODUCCIÓN

En la UTPL se viene trabajando con un entorno virtual de aprendizaje EVA, el cual ha potenciado el proceso de aprendizaje de los estudiantes. Todos los cursos se llevan a cabo en el entorno y se desarrollan en cada periodo, una parte importante de estos cursos en línea son las actividades de aprendizaje ejecutadas en ellos, ya que estas actividades no solo contribuyen a la culminación y aprobación de los cursos sino también a otorgar conocimiento a los estudiantes, de ahí la importancia de dar soporte a estas actividades.

En la actualidad también se tiene implementado en la UTPL un módulo en el EVA, para el manejo de recursos educativos abiertos (REA), en el cual el profesor sube sus recursos para ayudar al estudiante.

Estas herramientas generan sobrecarga de información, por la gran cantidad de recursos que se suben en cada curso y en cada periodo universitario.

Esto hace que el estudiante no pueda elegir los recursos adecuados para resolver sus actividades en un determinado curso, ya sea porque a veces no tiene relación o se encuentran distintos recursos que pueden ayudarlo pero no distingue cual es el mejor.

Con el avance de las tecnologías del e-learning estos problemas aumentaran debido a que la información crecerá exponencialmente en estos entornos, por lo que se hace necesario la implementación de sistemas recomendadores los cuales según el trabajo de (Wang & Wu, 2011) aportan con asistencia personalizada y en tiempo real, al proporcionar recursos en base a los requerimientos del estudiante en un entorno educativo y así puede ejecutar sus actividades de aprendizaje.

Para el diseño de un sistema recomendador se tiene que enfocar en las metas que tiene dicho sistema (Setten, 2005), y según estas metas profundizar en lo que se quiere realizar, para el proyecto la meta principal es dar soporte al estudiante en sus actividades de aprendizaje.

Cuando se tiene establecido el objetivo para el cual es creado un sistema recomendador se tiene que pensar en que técnicas serán las adecuadas para su desarrollo, revisando la literatura de los sistemas recomendadores existen una gran variedad de técnicas descritas en: (Burke, 2007; Adomavicius, 2005; Itmazi, 2005).

Tomando en cuenta que el agente tiene que trabajar en un campo educativo, hay que escoger las técnicas que mejor se integren a este campo, ya que no es lo mismo integrar un sistema recomendador en un campo comercial que en uno educativo.

En el trabajo de (Santos & Boticario, 2008) se tiene presente que en un ambiente educativo es mejor la utilización de sistemas recomendadores basados en conocimiento, por lo cual se ha orientado el trabajo por esa técnica de recomendación.

La importancia que tiene este proyecto para la institución y para la sociedad en general es bastante significativa ya que es un paso inicial al uso de tecnologías de recomendación, a brindar una mejor adaptabilidad de los sistemas de gestión de aprendizaje (LMSs¹) a las necesidades que tiene el usuario, promover la investigación de sistemas de recomendación y tecnologías web semántica que en la actualidad están en auge y cada vez son más utilizadas por sus amplias capacidades.

Durante el desarrollo de esta investigación también han surgido problemas referentes a integrar el agente en el EVA, ya que las tecnologías que se han utilizado son muy diferentes, por una parte el agente ha sido construido enteramente en java y el entorno utiliza en cambio el lenguaje php, se ha tenido que encontrar la forma de comunicar estas dos tecnologías.

Otro problema que se ha presentado ha sido el gasto computacional que implica la utilización de herramientas web semánticas integradas a la parte semántica del agente, esto ha implicado utilizar un computador con mejores prestaciones, también se tiene el problema de poder incorporar nuevas ontologías que permitan ampliar el contexto sobre el cual trabaja el agente por cuestiones de licencia lo cual ha limitado el trabajo a utilizar ontologías propias de la plataforma KIM².

Se han sabido sobrellevar estos problemas y más bien se ha visto las oportunidades, ya que las herramientas utilizadas se pueden adaptar a nuevos proyectos y con diferentes enfoques. A continuación se da una breve descripción de los capítulos que comprende este proyecto de tesis:

Capítulo 1: este capítulo contempla todo el estado del arte de los sistemas recomendadores, las diferentes técnicas que existen, herramientas que pueden ayudar a la construcción del agente recomendador.

¹ LMSs: Sistemas de Gestión de Aprendizaje

² KIM: Knowledge Information Management, plataforma para el manejo de conocimiento.

Capítulo 2: en este capítulo se realiza un monitoreo de las actividades del estudiante en el EVA, para de aquí identificar aquellas actividades que el estudiante no ha realizado en el entorno.

Capítulo 3: en este último capítulo se da a conocer la problemática y su solución, posteriormente se realiza un esquema de los que sería el agente recomendador, el cual ayuda a diseñar una arquitectura del agente y al final se da los detalles de la construcción del agente junto a su integración al EVA.

CAPITULO 1: ESTADO DEL ARTE

1.1. Sistemas de gestión de aprendizaje (Learning Management System)

Un sistema de Gestión de aprendizaje es un software que permite la administración y creación de cursos a través del internet, existe una variedad de estos sistemas, son esenciales en el aprendizaje virtual en el trabajo de (Itmazi, 2005) se menciona que: “Los LMSs son el alma del aprendizaje electrónico ya que sus plataformas están dedicadas con este fin y pueden ser comerciales o de código abierto”.

Todos los sistemas de gestión de aprendizaje manejan las conexiones del usuario registrado, el catálogo de los cursos, además monitorizan las actividades del estudiante y los resultados de las mismas.

Algunas de las funcionalidades de los LMSs son las siguientes:

- Manejo de cursos y programas
- Administración de registros en el curso (identificación)
- Seguimiento del registro, el acceso y el progreso del estudiante
- Administración de roles
- Control de matriculación y pagos

En general los Sistemas de Gestión de Aprendizaje son utilizados en su mayoría por instituciones educativas de todo tipo, en la investigación de (Álvarez, 2010), nos comenta acerca de los LMSs más extendidos y utilizados.

- Moodle
- Schoolar360
- Blackboard

Además en esta investigación se describe algunas limitaciones de los Sistemas de Gestión de Aprendizaje como la pobre adaptación al usuario, ya que el entorno no se presenta de una forma personalizada a las necesidades del mismo.

Una de las grandes potencialidades de los sistemas de gestión de aprendizajes es que la mayoría son de código abierto, esto no solamente permite utilizarlos libremente sino también modificarlos y añadirles nuevas funcionalidades. Como se menciona anteriormente existen algunas plataformas que son generalmente las más utilizadas y populares en las instituciones educativas, una de estas es Moodle que viene formando parte en la UTPL desde varios años.

1.2. Sistemas tutores inteligentes

En el proceso de enseñanza aprendizaje se ha venido implementando herramientas tecnológicas desde sus inicios (Álvarez, 2010), tanto es así que los primeros proyectos de enseñanza asistida por computador fueron CBT (Computer Based Training) y CAI (Computer Aided Instruction).

Actualmente se ha evolucionado a pasos agigantados en el campo tecnológico, lo cual ha permitido diseñar mejores sistemas de enseñanza asistida por computadora creando infinidad de proyectos para instituciones educativas. Por lo que hasta la fecha existen alrededor de 200 LMSs de código abierto.

En los años 80 esta evolución da lugar a los STI (Sistemas Tutores Inteligentes), los cuales dan solución a la necesidad de enseñanza individualizada al usuario.

El objetivo de un STI según (Álvarez, 2010) es “Trasmitir al estudiante los conocimientos de un dominio, de manera individualizada, teniendo en cuenta un conjunto de rasgos propios del usuario, tratando de emular a un profesor humano en el proceso de aprendizaje”.

En la actualidad tenemos todas las herramientas para explotar de alguna manera las capacidades que tienen los STIs y por ende los sistemas recomendadores ya que forman parte de los sistemas tutores inteligentes y poder brindar una mejor asistencia al estudiante principalmente enfocados a este proyecto a nivel de actividades que realiza el mismo dentro del EVA de la UTPL

Es importante referirse a los sistemas tutores inteligentes y hacer una descripción de sus componentes en esta investigación debido a que con ellos se dio el origen de la enseñanza asistida por computadores, son sistemas que tratan de enseñar conocimiento individual con uso de técnicas de Inteligencia Artificial, tiene una estrecha relación con los agentes recomendadores por la arquitectura clásica que utilizan, donde se describen sus componentes, los cuales son utilizados por los agentes recomendadores como el modelo del estudiante, componente válido para la personalización de las recomendaciones. Los componentes de la arquitectura clásica de un STI surgen de la separación y sensibilidad a que se enseña, como se enseña y a quien se enseña (Nwana, 1990).

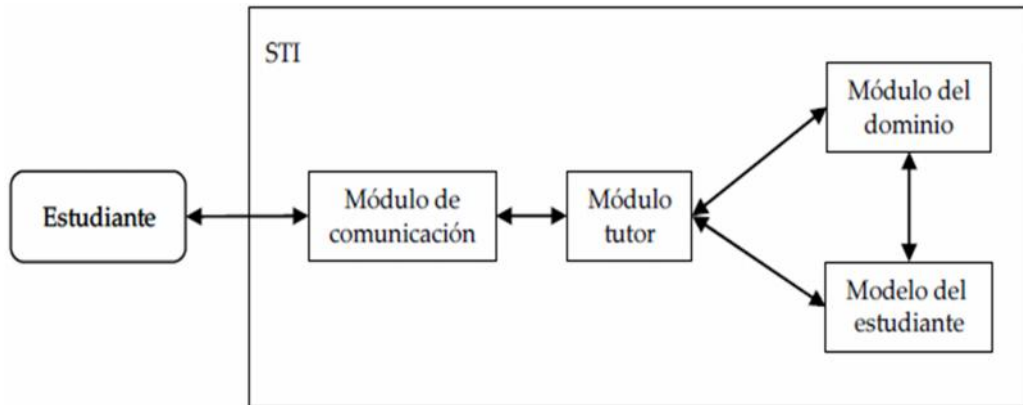


Gráfico 1 Componentes de un STI (Álvarez, 2010)

En el gráfico 1 que representa la arquitectura básica de un STI, se muestra cada uno de sus componentes, los cuales surgieron para individualizar la enseñanza.

Según el mismo autor “Los STIs ayudan al estudiante con una secuencia de actividades e interacciones que favorezcan su aprendizaje, pero basados en factores como conocimiento adquirido o errores cometidos anteriormente, la enseñanza se desarrolla de forma dinámica adaptada a cada ejecución de sesión del estudiante”, de ahí la importancia de los STI en los agentes recomendadores como establecimiento de un modelo especialmente del estudiante ya que es el principal actor en este proyecto.

Generalmente los LMSs aportan una gran flexibilidad a los estudiantes, ya que pueden realizar cambios a nivel de interfaz en colores, tipos de letra, presentación, pero lo que no ofrecen es una buena adaptabilidad a las necesidades del estudiante, lo que un sistema tutor inteligente si lo hace.

A continuación describimos cada componente de la arquitectura:

Módulo de Comunicación

Es el encargado de gestionar la comunicación entre el estudiante y el Sistema Tutor Inteligente.

Modelo del Estudiante

Este Modelo representa las acciones de estudiante dentro del sistema, su conocimiento y aplicación del mismo.

Según (Álvarez, 2010) “Idealmente tendría que representar aspectos sobre su comportamiento y conocimiento que repercuten en su aprendizaje”

La modelización del estudiante conlleva analizar y representar aspectos tales como conocimiento del dominio, perfil, histórico de sesiones que inciden en su aprendizaje.

Con el objetivo de adaptar a las características del estudiante los sistemas LMSs, hay que representar sus acciones dentro del mismo para incluir soluciones que le presten apoyo a su aprendizaje enfocarse en sus actividades.

Modulo Tutor.

Este módulo es el encargado de organizar las actividades pedagógicas que el estudiante tiene que realizar.

En (Álvarez, 2010), este modelo monitorea las actividades del estudiante adaptando sus acciones a sus respuestas, pero sin perder el control, además nos explica que el control puede ser total del estudiante o puede ser compartido con el sistema.

Modulo del Dominio.

El dominio es el campo de conocimiento en el cual se trabaja, y es muy importante ya que representa el contexto donde se puede utilizar el agente recomendador del dominio depende la técnica y tipo de agente a utilizar, en el caso de un campo educativo el dominio tiene que ver con los curso establecidos dentro del LMS y todos los temas generados o que se puedan generar dentro del mismo.

1.3. Sistemas de recuperación de información

Los sistemas de recuperación de información han venido siendo utilizados desde hace tiempo para suplir las necesidades de información de cada usuario, siendo así que la mayoría de estos sistemas se utilizan en los principales buscadores en internet como es el popular google.

Con el auge de diferentes tecnologías como las de internet es necesario que el usuario tenga accesibilidad a información útil a sus necesidades, esto es un problema muy grave ya que al momento no se cuenta con una solución determinante.

En el trabajo de (Setten, 2005), una persona tiene tres enfoques para encontrar información o ítems de interés: recuperación de información, filtrado de información y navegación de la información.

El primer enfoque tiene que ver con una consulta realizada por el usuario y encontrar la información adecuada a la consulta, la mayoría de motores de búsqueda utilizan este enfoque.

El segundo enfoque tiene que ver con sistemas de filtrado de información, es decir el sistema filtra u ordena un gran volumen de información y presenta solo la que le puede servir al usuario, en este enfoque la información es dinámica y se actualiza periódicamente. Además la información del usuario puede ser obtenida de forma explícita ofertada por el mismo usuario o el sistema puede hacerlo de forma implícita.

El tercer enfoque la formulación de la consulta se relaciona estrechamente con el resultado, es utilizado para solventar necesidades confusas, el usuario desconoce realmente lo que quiere y navega por información que podría satisfacer su requerimiento.

A través del tiempo se han venido mejorando los servicios de recuperación de información tradicional, este cambio ha permitido que en la actualidad se tengan recuperadores de información que basan su filtrado de información mediante el significado del término a buscar, esto hace que la búsqueda sea más eficiente.

El principal enfoque de un recuperador de información es buscar documentos adecuados a la consulta del usuario y devolverlos.

Según (Itmazi, 2005), hay tres conceptos a describir que están ligados a los sistemas recomendadores y tiene también relación con la arquitectura tradicional de un STI: Sistemas de recuperación de información, la personalización y el modelo de usuario.

El modelo de usuario se ha definido anteriormente, la personalización tiene relación a la adaptabilidad del LMSs con las características del usuario.

Un sistema de recuperación de información según (Itmazi, 2005) consiste en “un dispositivo interpuesto entre un usuario potencial de información y la colección de información en sí misma. Para un problema de información dado, el objetivo del sistema es capturar artículos deseados y eliminar mediante filtrado artículos no deseados”

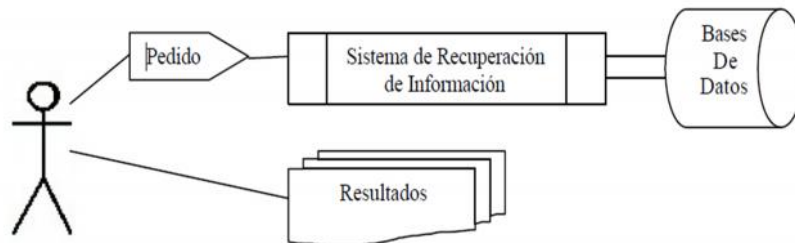


Gráfico 2 Sistema de Recuperación de Información IRIS (Itmazi, 2005)

Componentes de un sistema de recuperación de información

En la investigación de (Lidó, 2002) se identifican los siguientes componentes:

Una colección de Documentos: es la fuente de información donde se va a recuperar los documentos pueden estar estructurados o no

Un lenguaje de consulta: la consulta es realizada por el usuario que requiere de la información y esta puede ser un término, conjunto de palabras, cadena de caracteres, entre otros.

Aplicación: es la encargada de realizar todo el proceso de búsqueda y presentación de información a su vez tiene lo siguiente:

- Gestor de repositorio: encargado de administrar el conjunto de documentos, adicionalmente tiene un sistema de indexación.
- Proceso de emparejamiento: evaluador encargado de adecuar los documentos en el repositorio con las consultas del usuario.

- Proceso de relevancia: es el que organiza los documentos acorde a la relevancia de los mismos y los presenta en una lista a los usuarios.

Tipos de sistemas de recuperación de información

Según la misma investigación se ha establecido un modelo tratando de tipificar los sistemas de recuperación de información. Este modelo se basa en el funcionamiento del sistema, en un inicio se utilizó la búsqueda por emparejamiento booleano lo cual permite al usuario ingresar un término que coincida exactamente con el documento que se le devolverá, esto ha generado diferentes problemas de ambigüedad haciendo que el usuario obtenga información que no satisface su consulta o en otros casos obtener inmensidad de documentos haciendo imposible al usuario discernir cual utilizar. Por otra parte se ha considerado mejorar la búsqueda orientándola a una búsqueda estadística, es decir permitir al usuario establecer la importancia de los términos para la búsqueda, el sistema calcula la relevancia de los documentos presentados al usuario en función de los términos que aparecen en el documento. Si esta relevancia supera cierto umbral, el documento se presenta aunque ciertos términos no aparezcan en él.

Este modelo nos presenta dos tipos de sistemas de recuperación de información: sistemas de recuperación de información exactos y sistemas de recuperación de información aproximados.

1.4. Sistemas recomendadores

En (Adomavicius, 2005), se comenta que aunque las raíces de los sistemas recomendadores se remontan a una extensiva labor en ciencia cognitiva, teoría de aproximación, recuperación de información, manejo de ciencia y modelos de consumidores, se establece como un área de investigación individual a mediados de 1990, cuando los investigadores enfocan su atención al problema de recomendación que se basa en una estructura de calificaciones.

Según este mismo trabajo “En la formulación más común de un problema de recomendación, el problema es reducido a la estimación de calificaciones (ratings) para ítems que no han sido vistos por el usuario”.

En ocasiones estas calificaciones vienen dadas por el usuario a otros ítems. Una vez que se tenga estas calificaciones a un ítem, se puede recomendar los mejores calificados a un usuario.

Un problema de recomendación se puede definir de la siguiente manera de acuerdo al trabajo citado anteriormente:

Sea C el conjunto de usuarios y sea S el conjunto de todos los ítems que posiblemente se recomienden. El conjunto de ítems puede ser bastante amplio hasta en un rango de millones en algunas aplicaciones, como también el rango de usuarios es bastante grande.

Se designa u como una función de utilidad del ítem al usuario.

$$u = C \times S \longrightarrow R$$

La R representa un número entero no negativo y en un cierto rango.

En la investigación llevada a cabo en el presente artículo el conjunto C de usuarios está determinado por los estudiantes y el conjunto S son los ítems a recomendar es decir los recursos que dan soporte a sus actividades en el EVA.

La estimación de ratings es ejecutado de una manera diferente ya que el estudiante no tiene que puntuar (rated) los recursos, esta estimación se realiza de forma semántica y las puntuaciones se dan de acuerdo a la relación semántica del recurso con la actividad que el estudiante tiene que realizar en el EVA.

Los sistemas recomendadores forman parte de los sistemas tutores inteligentes y surgen por la necesidad de personalizar los sistemas de gestión de aprendizaje, brindando mejores opciones de información filtrada y adecuada a las características del usuario.

Existen varios tipos de sistemas recomendadores según el contexto en el que se lo vaya a utilizar en el caso de esta investigación es orientado al soporte de las actividades de aprendizaje de los estudiantes en el EVA.

Se trata de una técnica que posibilita filtrar información, de utilidad a un usuario en específico basado en su modelo, es decir en sus características personales y actividades de interacción con el sistema.

Independientemente de la técnica de recomendación que se utiliza todos los sistemas recomendadores poseen dos elementos integrados en su estructura: uno es el modelo de usuario y la otra es información de elementos que se puede recomendar al mismo.

En el trabajo de (Burke, 2007), el cual describe un sistema de recomendación híbrida, es decir un sistema que combina varias técnicas de recomendación da una clasificación de los diferentes técnica de recomendación.

Esta clasificación se basa en la fuente de conocimiento o el contexto para realizar la recomendación. En algunos sistemas el conocimiento puede ser las preferencias de usuario, en otros es ontológica o inferencia del dominio, agregada por la ingeniería del conocimiento de un humano.

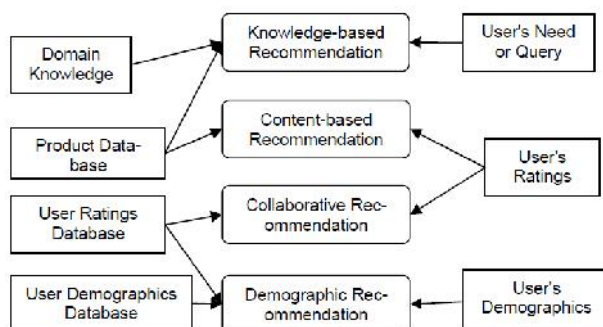


Gráfico 3 Tipos de agentes Recomendadores y sus fuentes de conocimiento (Burke, 2007).

Como se puede observar en la gráfica, las fuentes de datos pueden ser diversas como por ejemplo en un sistema donde se aplique dominio del conocimiento puede ser una técnica basada en conocimiento, en un sistema donde se trabaje con información del ítem o producto puede ser basado en conocimiento y basado en contenido, al utilizar las calificaciones del usuario puede ser colaborativo o basado en demografía y por ultimo al utilizar información demográfica del usuario se puede utilizar una técnica basada en demografía.

1.4.1. Clasificación Sistemas Recomendadores según su fuente de conocimiento.

Se describe cada tipo de sistema recomendador según la clasificación del grafico 3, según esta clasificación un sistema recomendador se tipifica de acuerdo al producto que se va a recomendar y el contexto en el que trabaja.

Recomendadores Demográficos

Esta técnica de recomendación utiliza el perfil demográfico del usuario, es decir información relevante del mismo como edad, sexo, estado civil, luego organiza usuarios con información en común y realiza la recomendación a los grupos con similitud en su clase demográfica.

Los beneficios de esta técnica es que se puede realizar la recomendación sin tener un historial de datos puntuados como se lo hace en técnicas de filtrado colaborativo y basado en contenido.

Recomendadores Colaborativos:

Esta técnica de recomendación es la más familiar, más utilizada y madura de todas las técnicas. Su metodología consiste en puntuar objetos o ítems a recomendar, reconoce similitudes entre usuarios basado en estas puntuaciones (rating), para finalmente generar una recomendación a un nuevo usuario (objetivo).

Un perfil de usuario típico en estos sistemas consta de un vector de ítems y sus respectivas puntuaciones y se va incrementando en función de la interacción del usuario con el sistema. Su sistema de puntuación puede ser binario (me gusta/no me gusta) o puede tener un rango de valores.

Los sistemas colaborativos puede ser de dos tipos: basado en memoria o basado en modelo. Basado en memoria hace una comparación directa de usuarios usando correlación mientras que en el basado en modelo se deriva un modelo de la información histórica del usuario y así poder realizar la predicción, este utiliza varias técnicas de aprendizaje como redes neuronales, indexado semántico y redes bayesianas.

El punto fuerte de esta técnica es su completa independencia de cualquier representación o contenido de objetos en un formato analizable por computador y trabaja muy bien con objetos complejos como música y videos.

Esta técnica ha sido reconocida como la más exitosa por algunos expertos.

Recomendadores Basados en Contenidos:

Es una de las aproximaciones más antiguas de filtrado. Estos sistemas de recomendación se basan en dos recursos. Uno es el contenido de un ítem, y el otro recurso es el perfil de usuario, sus gustos, realiza una comparación para realizar la recomendación.

Es muy importante que los objetos estén en cierto formato para que puedan ser leídos por la maquina o se puede agregar la valoración manualmente.

Establece una correlación entre el perfil del usuario y el contenido de un objeto, puede de esta forma recomendar objetos nuevos al usuario pero que su contenido esté relacionado a su perfil.

Recomendadores Basados en Conocimiento:

La técnica de recomendación basada en conocimiento realiza inferencia de acuerdo a las necesidades del usuario, utiliza un conocimiento anterior de como un ítem puede satisfacer estas necesidades y en base a eso generar la recomendación

Los recomendadores basados en conocimiento son bastante complejos por lo que utilizan características semánticas de los ítems u objetos a recomendar, por lo que para realizar recomendaciones los objetos tienen que estar bien estructurados. Para estructurar los objetos se utilizan las denominadas ontologías.

Para este tipo de recomendaciones es necesario aplicar ingeniería del conocimiento.

1.5. Agentes recomendadores en el campo educativo

Los agentes recomendadores en su mayoría se han implementado en el campo comercial para determinar gustos de usuario y relacionarlos con productos en la web, lo cual permite recomendar al usuario estos productos de entre varias masas de información y mejorar ventas, un claro ejemplo ha sido el sitio de comercio electrónico Amazon la cual cuenta con un sistema recomendador de filtrado colaborativo.

En el campo educacional el tema de sistemas recomendadores es reciente, y surge con los sistemas adaptativos para la Web (Adaptative Intelligence Web Based Educational Systems AIWEB), los cuales tienen sus antecedentes en los Sistemas Tutores Inteligentes, en donde se pretende realizar sistemas personalizados adaptados a las necesidades de los estudiantes dentro de un entorno virtual de aprendizaje.

Según (Santos & Boticario, 2008) los sistemas recomendadores en el campo educativo han incrementado su demanda por lo que son necesarios para ofertar al estudiante actividades y recursos adecuados, que mejoren sus metas de aprendizaje y den soporte a sus requerimientos de una mejor manera.

Además hace hincapié en los beneficios de los recomendadores durante el desarrollo de un curso tanto para profesor y estudiante:

Estudiante:

- Evitar Bloqueos
- Mejorar el funcionamiento del proceso de aprendizaje facilitando los contenidos del curso más apropiados y rutas de aprendizaje encaminadas a las necesidades del estudiante.
- Promover la colaboración entre pares.

Profesor

- Entregar recomendaciones al estudiante en el momento correcto
- Seguir el trabajo del estudiante dentro del curso.

En este trabajo, también nos menciona que aunque el objetivo es el mismo tanto en el campo comercial como en el educativo la de recomendar productos o ítems adecuados a las necesidades del usuario de entre una gran cantidad información, hay diferencias muy marcadas con un entorno educativo:

Requerimientos: las recomendaciones deben ser guiadas pedagógicamente y no solo por preferencias de usuario.

- Predisposición del usuario : los usuarios no les motiva calificar ítem en un sistema de aprendizaje como lo harían en un sistema de índole comercial
- El contexto estructural: la estructura de un entorno educativo no es la misma que la de un entorno comercial, en el comercial se hace fácil recomendar los productos tales como música o videos mientras que en un entorno educativo la estructura se basa en el curso y las actividades del mismo por lo que un agente recomendador maneja información un poco más compleja y sus recomendaciones deben buscar mejorar el aprendizaje del estudiante.

Al seguir analizando la información de esta investigación se tiene que han llegado a la conclusión de que “Los escenarios de aprendizaje electrónico deben de combinar diseño y

adaptaciones en tiempo de ejecución para un mejor soporte al estudiante en todo su ciclo de vida en el proceso de aprendizaje”

Con esta idea se ha creado un modelo para administrar recomendaciones en tiempo de diseño para soportar operaciones en tiempo de ejecución. Esto hace que el sistema sea dinámico al momento que el estudiante interactúa dentro del LMS.

Dicho modelo permite crear varios tipos de recomendaciones, en el caso de la investigación (Santos & Boticario, 2008) permite generar acciones en el LMS tanto a profesores como a estudiantes. Al ser implementadas se puede poner condiciones en tiempo de diseño para luego ser computadas en tiempo de ejecución contra el contexto actual (curso, usuario y dispositivo)

En el caso de este proyecto de tesis este modelo es importante por la similitud en las características y el objetivo que se busca el cual es brindar al estudiante soporte en sus actividades del EVA, se puede manejar de igual manera las recomendaciones en tiempo de diseño se establece que actividades requieren soporte para luego en tiempo de ejecución dar la recomendación en base a esas actividades.

Otro punto importante descrito del modelo es que “trabajar en escenarios de aprendizaje muestra que las recomendaciones en ellos, deben basarse en la información más relevante de cada estudiante y las actividades disponibles, información histórica acerca de estudiantes y actividades similares (plan de formación) guiado por reglas y estrategias de aprendizaje dirigido a la obtención de metas en este aprendizaje.”

Elementos del modelo de recomendaciones en un entorno educativo.

Se presenta el siguiente modelo de recomendación para escenarios de aprendizaje:

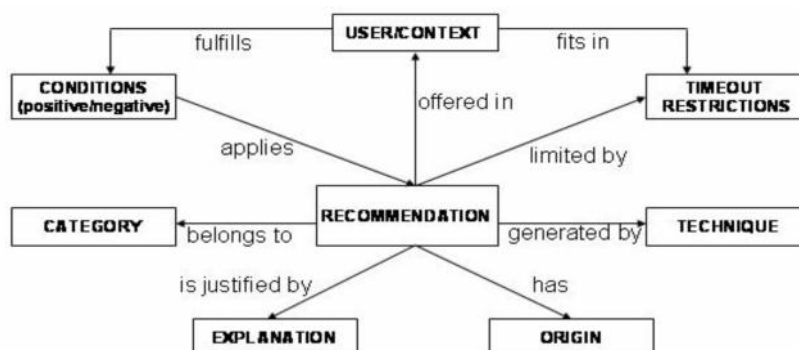


Gráfico 4 Modelo de recomendaciones para escenarios de aprendizaje (Santos & Boticario, 2008)

El gráfico 4 resume las opciones de modelado que caracteriza una recomendación en escenarios de aprendizaje se explica brevemente este modelo.

Una recomendación pertenece a una de las ocho categorías definidas y puede ser generada por una o la combinación de varias técnicas de recomendación. Durante el diseño el diseñador del curso selecciona las categorías a las cuales son aplicadas las recomendaciones, define las condiciones y restricciones de tiempo y agrega la explicación.

La estructura de la recomendación apunta a los tipos de recomendación disponibles, los cuales son los servicios del LMS. El origen y la técnica dependen de la forma en que se generan las recomendaciones. En tiempo de ejecución las condiciones y restricciones se comprueban con el modelo de usuario y el contexto en cuestión. Al final las recomendaciones son ofertadas al usuario en una interfaz usable, explicativa y accesible.

Se describe algunos elementos de este modelo que se utilizan en el proyecto de tesis

Categorías

Este elemento clasifica el alcance de la recomendación ofertada, en base a la experiencia la categoría depende de la situación del curso y algunas son más utilizadas que otras y ayudan al estudiante a llevar a cabo una tarea de manera eficiente, efectiva y satisfactoria

Se menciona 8 categorías, las cuales están abiertas a cambios si es que se encuentran coincidencia o hay algún valor faltante, depende del curso y del contexto que se requiera dar a las recomendaciones.

La Técnica

Este elemento del modelo tiene que ver con la técnica utilizada para generar la recomendación. Existen varias técnicas de recomendación que pueden ser basadas en modelo y basadas en memoria, las que mejor se acoplan a un entorno educativo según la investigación de (Drachsler, Hummel, & Koper, 2007) son las técnicas basadas en memoria y se describen detalladamente en lo posterior.

El Origen

El origen identifica la fuente que origina la recomendación, en el trabajo de (Santos & Boticario, 2008), se dan estos tipos de origen:

- Preferencia: se enmarca en las preferencias del usuario
- Popularidad: usuarios similares que ya han utilizado la recomendación
- Tutor: recomendaciones originadas por el tutor o profesor del curso
- Diseño del Curso: son recomendaciones que nacen en el diseño del curso

Al igual que la categoría este valor puede cambiar o agregarse otros valores que surgen del contexto de la recomendación.

Condiciones Aplicables

Son las condiciones para ofertar la recomendación, define que valores deben tomar lugar para un usuario y su contexto en tiempo de ejecución para darse la correspondiente recomendación.

Estas condiciones pueden darse identificando un conjunto de valores que deben tomarse para ser dada la recomendación (condiciones positivas) y el conjunto de valores que deben no tomarse para ser dada la recomendación (condiciones negativas)

En cuanto a información de usuario estos valores pueden ser los siguientes:

Plataforma: La recomendaciones son ofertadas cuando el estudiante está utilizando las instalaciones del LMS y pueden ser aplicadas cuando el usuario trabaja en un entorno específico y con los recursos dados. Se toman en cuenta también acciones como escribir, leer, postear, etc., realizado en objetos del LMS como foros, tareas, evaluaciones, mensajes, entre otros.

Interfaz de Usuario

Como en todo Sistema Recomendador una cuestión crítica es la forma de presentación de las recomendaciones. El enfoque más conveniente es mostrar un subconjunto de recomendaciones adecuadas a las necesidades de los usuarios, además se debe otorgar una explicación del origen de las recomendaciones para que la confianza del usuario en el sistema aumente.

1.6. Técnicas de recomendación en entornos educativos.

Se ha visto de manera general las técnica de recomendación pero una clasificación detallada de estas técnicas se la puede encontrar en (Adomavicius, 2005), este articulo hace un repaso de todas las técnicas de recomendación existentes, por lo concerniente a este proyecto se describe detalladamente las técnicas que utiliza el agente.

Existe diversidad de técnicas de recomendación que se describen en diferentes trabajos y proyectos de agentes recomendadores, en el caso de esta investigación es necesario referirse a las técnicas de recomendación en ambientes educativos, un trabajo interesante (Drachsler, Hummel, & Koper, 2007), donde se maneja un PRS (Personal Recommender System) en redes de aprendizaje describe técnicas de recomendación adecuadas para esta investigación.

Este articulo trata de las diferentes técnicas basadas en memoria, las cuales calculan la recomendación basándose en un conjunto de datos, propone una combinación de ellas para la creación de un PRS en redes de aprendizaje, todo esto en un contexto de aprendizaje electrónico (e-Learning).

Según (Itmazi, 2005), el nivel más alto en que se clasifican las técnicas de recomendación es:

- Técnicas basadas en modelo
- Técnicas basadas en memoria

1.6.1. Técnicas de Recomendación basadas en modelo

Las técnicas de recomendación basadas en modelo operan sobre una base de datos del usuario para establecer un modelo y en base a este modelo ofertar las recomendaciones.

Estas técnicas tienen la característica de aprender lentamente y dar predicciones rápidamente. El modelo se lo puede construir fuera de línea.

Generalmente necesitan de un gran cuerpo de datos cerca de 10000 para poder ofertar una recomendación, para evitar problemas de nuevo usuario o nuevo ítem que veremos más adelante.

Algunas técnicas que trabajan de buena forma con modelos basados en modelo son redes bayesianas, clustering y reglas.

Técnicas de clustering

En estas técnicas lo que se busca es formar grupos con similares afinidades, de acuerdo a su valoración de los ítems se generan clases. Una vez que se obtienen estas clases son independientes una de la otra, entonces se puede recomendar los ítem a clases que tengan gustos comunes.

Técnicas de redes bayesianas

Este modelo utiliza las denominadas redes bayesianas donde cada nodo corresponde a un ítem en determinado dominio, el estado del nodo corresponde a la valoración del ítem, también se incluye un estado de nodo “no valorado” para esos dominios donde no hay una interpretación natural para datos perdidos.

Luego se aplica un algoritmo de aprendizaje de la red bayesiana a los datos de entrenamiento. El algoritmo busca sobre varios modelos estructurales en términos de dependencia para cada ítem. Como resultado tenemos una red, en donde cada nodo o ítem tendría un conjunto de nodos o ítems padres que son las mejores predicciones en base a sus valoraciones. Cada tabla de probabilidad condicional es representada por un árbol de decisiones.

1.6.2. Técnicas de recomendación basadas en memoria

Las técnicas basadas en memoria operan sobre la base de datos entera del usuario para hacer las predicciones. Al contrario que las basadas en modelo aprenden rápidamente pero dan las recomendaciones lentamente.

Las técnicas de recomendación basadas en memoria son las más populares ya que se utilizan en entornos dinámicos, es decir en donde los datos se actualizan continuamente y pueden obtener predicciones relativamente exactas

Las técnicas más extendidas basadas en memoria son: filtrado basado en el vecino más cercano KNN (K-nearest-neighbour) denominada como los k vecinos más próximos y la técnica de filtrado colaborativo basado en usuario.

Un ejemplo donde se puede ver este tipo de técnica es GroupLens, el cual filtra noticias para ofertar a los usuarios las que se adecuen a sus preferencias.

Se ha descrito los dos tipos de técnicas de recomendación, en esta investigación se utiliza la que mejor se adapte a las necesidades del proyecto, en este caso se utiliza la recomendación en un entorno educativo y orientado a las actividades que realiza el estudiante dentro del mismo, la idea principal es dar soporte a estas actividades.

Se han revisado trabajos para ver que técnicas conviene utilizar en entornos educativos y se ha discutido el tema de que el enfoque de recomendaciones cambia en un entorno educativo que en uno comercial, al seguir el trabajo de (Drachsler, Hummel, & Koper, 2007), se especifica que "Todo sistema Recomendador sirve a un propósito específico y funciona en un contexto específico". Ellos operan de acuerdo a estrategias y técnicas de recomendación predefinidas. Cada técnica de recomendación utiliza un método único para la creación de la recomendación. Este trabajo sugiere la utilización de técnicas de recomendación basadas en memoria ya que son las que mejor se adaptan a entornos educativos debido a la necesidad de utilizar o identificar información completa del usuario como del ítem que se requiere recomendar en todo el ciclo de vida de aprendizaje.

En tecnologías para la educación un agente recomendador trata con información del estudiante como pueden ser sus actividades de aprendizaje y tiene que combinar varios niveles de dificultad para las diferentes situaciones de aprendizaje en el que está envuelto el estudiante.

El primer paso para definir un sistema recomendador es establecer los requerimientos del usuario y el propósito en el campo que se va a desarrollar el recomendador.

Este trabajo también nos menciona que una pregunta importante en tecnologías para la educación es: "De qué manera se ve el contexto y el dominio de los estudiantes en su ciclo de aprendizaje y cuáles son los principales stakeholders³"

Además se toma en consideración que una recomendación en un campo comercial es mucho más simple que una recomendación en un sistema de gestión de aprendizaje.

³ Stakeholders: personas que están involucradas o que interactúan con un sistemas

Toda esta literatura nos lleva a enfocarnos específicamente en los tipos de técnicas de recomendación basados en memoria, en las siguientes tablas veremos qué ventajas tienen estas técnicas enfocadas en su utilidad para el agente.

Tabla 1 Filtrado colaborativo basado en usuario

Descripción	Ventajas	Desventajas	Utilidad en Agente	Es utilizable
En esta técnica el usuario califica varios ítems. Una vez que se tiene los ítems calificados se recomienda a otro usuario similar ítems que no haya visto.	<ul style="list-style-type: none"> No se realiza análisis de contenido. Es independiente del dominio. Calidad en recomendaciones Tiene un enfoque ascendente Serendipity: ofertar recomendaciones por casualidad 	<ul style="list-style-type: none"> Problema de Nuevo Usuario Problema de Nuevo Ítem Gustos Populares Escalabilidad Escases Problema de arranque en frío 	Según el enfoque del agente esta técnica no sería muy aconsejable, por lo que es necesaria información explícita, es decir que los usuarios hayan calificado varios ítems para generar una recomendación. Problemas de nuevos usuarios o ítems, además el usuario tiene que atenerse a recomendaciones de gustos populares. Tal vez combinada con otras técnicas y en una fase inicial	No

Tabla 2 Filtrado colaborativo basado en ítems

Descripción	Ventajas	Desventajas	Utilidad en Agente	Es utilizable
Esta técnica de recomendación se enfoca en los ítems, al tener una alta calificación el sistema asume que son parecidos y lo recomienda al usuario.	<ul style="list-style-type: none"> No se realiza análisis de contenido. Es independiente del dominio. Calidad en recomendaciones Tiene un enfoque ascendente Serendipity: ofertar recomendaciones por casualidad 	<ul style="list-style-type: none"> Problema de Nuevo Ítem Gustos Populares Escases Problema de arranque en frío 	Al tener que calificar los ítems padece de los mismo problemas que el filtrado basado en usuario como es el de un nuevo ítem, atenerse a gustos populares y el problema de arranque en frío que significa si no hay suficientes datos calificados no es posible emitir recomendaciones	No

Tabla 3 Filtrado colaborativo basado en estereotipos o demográfico

Descripción	Ventajas	Desventajas	Utilidad en Agente	Es utilizable
Esta técnica de recomendación utiliza datos del usuario como es edad, estado civil, sexo,	<ul style="list-style-type: none"> No hay problema de arranque en frío 	<ul style="list-style-type: none"> Obtener información Información insuficiente 	Se tiene problemas al obtener información de cierta forma delicada del usuario, además el	No

entre otros para asociar recomendaciones en base a estos atributos demográficos	<ul style="list-style-type: none"> Independiente del Dominio Casualidad 	<ul style="list-style-type: none"> Solo Gustos populares Obtener información (Metadata) Mantenimiento de ontología 	enfoque del agente no está relacionado con la información demográfica que pueda tener un estudiante.	
---	---	---	--	--

Tabla 4 Razonamiento basado en casos

Descripción	Ventajas	Desventajas	Utilidad en Agente	Es utilizable
En esta técnica un ítems se recomienda en base a su correlación a un ítem que le haya gustado antes al usuario.	<ul style="list-style-type: none"> No hay análisis de contenido Independiente del Dominio Mejor Calidad 	<ul style="list-style-type: none"> Problema de nuevo usuario Superespecializacion Escases Problema de arranque en frio 	Tiene el problema de arranque en frio es decir que se necesita de bastante información calificada para realizar la recomendación	No

Tabla 5 Técnica de recomendación basada en atributos

Descripción	Ventajas	Desventajas	Utilidad en Agente	Es utilizable
Recomienda ítems basado en el emparejamiento de sus atributos con el perfil de usuario. Los atributos son ponderados por su importancia para el usuario	<ul style="list-style-type: none"> No hay problema de arranque en frio No hay problema de usuario ni ítem nuevo Sensible a cambio de preferencias Puede incluir características no relacionadas con el ítem Puede mapear las necesidades del usuario con el ítem 	<ul style="list-style-type: none"> No aprende Solo trabaja con categorías Es requerido un modelamiento y mantenimiento de ontologías Superespecializacion 	Según el enfoque del agente esta es una de las técnicas más aconsejables de utilizar junto al subtipo de esta técnica como la basada en conocimiento puede prevenir algunos problemas que padece esta técnica	SI

Técnicas de filtrado Colaborativo

Esta técnica de recomendación utiliza el comportamiento colectivo de los estudiantes en su entorno. Forman parte de las técnicas de filtrado colaborativo:

- Filtrado colaborativo basado en usuarios
- Filtrado colaborativo basado en ítem
- Filtrado colaborativo de estereotipos o basado en demografía

Filtrado colaborativo basado en usuario y en ítems: una de las ventajas principales de estas dos técnicas de recomendación es que utilizan información valorada de pies a cabeza por el usuario así que no necesitan información o análisis del contenido del ítem. Estas técnicas mejoran con el transcurso del tiempo.

Por otro lado el aspecto negativo de estas técnicas es que tienen varios problemas, en primer lugar tienen el problema de “arranque en frío”, es decir necesitan de un historial suficiente de datos antes de que se pueda dar una recomendación.

Aun cuando estos sistemas han estado en funcionamiento van a padecer de este problema con un nuevo usuario o un nuevo ítem. Otro problema para estas técnicas es la escasez de acciones pasadas de un usuario en la red. Dado que estas técnicas trabajan en comunidades dan soporte a gustos populares que gustos impopulares, lo cual genera problemas al momento que un usuario con gustos inusuales prueba el sistema. Otro problema es la escalabilidad el sistema se ve afectado al recibir una carga masiva de información como ejemplo en Amazon que tiene que ser capaz de dar recomendaciones en tiempo real de millones de artículos o ítems.

El uso de estas técnicas en entornos educativos tiene el punto fuerte de valerse de información histórica exitosa de otros estudiante pero como se ha dicho esta tiene que ser en gran cantidad, para resolver los problemas se la puede combinar con cualquier otra técnica como la basada en estereotipos o demografía del usuario.

Filtrado colaborativo basado en demografía: a través de estereotipos se pueden filtrar ítems para recomendarlos a usuarios parecidos basados en sus mutuos atributos. Las ventajas son que la independencia del dominio y comparado con filtrado basado en usuario e ítem está técnica no requiere de mucha información histórica para generar la recomendación. Por esta ventaja esta técnica es utilizada para resolver el problema de “Arranque en frío”, además tiene la destreza de recomendar ítems parecidos pero aún desconocidos para un usuario, lo que significa que un usuario descubre su ítem preferido por casualidad

La principal desventaja de esta técnica es que no les gusta a los usuarios dar información demográfica, se les hace un proceso tedioso y al no tener la suficiente información las recomendaciones se ven afectadas directamente.

En escenarios de educación esta técnica sirve de mucho para agrupar a los estudiantes si es que la información conductual no es suficiente. Esta técnica combinada con las que sufren de

“arranque en frío” puede ser buena opción como complemento y generar recomendaciones adecuadas desde un principio.

Técnicas de Filtrado basado en contenido

Estas técnicas de recomendación utilizan información del usuario o del ítem, existen dos tipos razonamiento basado en casos y basado en atributos.

Razonamiento basado en casos: esta técnica recomienda ítems con una alta correlación a ítems que gustaron al usuario antes. La similitud entre ítems está basada en sus atributos. Estas técnicas comparten algunas de las ventajas de otras técnicas como: son independientes del dominio, no necesitan análisis de contenido y la calidad de la recomendación mejora con el tiempo cuando el usuario ha valorado muchos ítems.

Entre las desventajas de estas técnicas cuentan la del problema de nuevo usuario, superespecialización y escasez, porque solo ítems con alta correlación con el perfil de usuario o interés pueden ser recomendados. A través de esta técnica el usuario está limitado a un gran conjunto de ítems que tienen similitud con ítems que ya conoce.

En un escenario educativo esta técnica sirve para mantener informado al estudiante de camino a las metas de aprendizaje. Las actividades de aprendizaje recomendadas a un estudiante son las mismas que otro prefirió anteriormente. El sistema recomendador también puede estructurar las recomendaciones en base a reglas pedagógicas.

Técnicas basadas en atributos: la principal ventaja de estas técnicas es que no tienen el problema de “arranque en frío”, ya que toman en cuenta solo los atributos de usuario o ítem para generar la recomendación. Por lo que agregar nuevas actividades de aprendizaje o estudiantes no debe causar ningún problema en estos sistemas. Los sistemas con estas técnicas son sensibles a cambios en los perfiles del estudiante. Esto se puede controlar cambiando estos perfiles o los pesos relativos de los atributos. Una descripción de las necesidades de los perfiles es asignada directamente a la actividad de aprendizaje disponible.

Una seria desventaja es que la técnica es muy estática y no permite el aprendizaje de comportamiento de la red. Es la razón por la que una recomendación altamente personalizada no se puede alcanzar, este tipo de técnicas trabaja solo con información que puede ser descrita en categorías. Los formatos como audio y video primeramente tienen que clasificarse a los temas en el perfil del estudiante, esto requiere de una modelización categórica y mantenimiento

que puede aumentar los problemas en un entorno de aprendizaje. Por último se da el problema de la superespecialización sobre todo cuando el estudiante no cambia su perfil.

En el campo educativo esta técnica solventa el problema de “arranque en frío” ya que no necesita información del comportamiento del estudiante. Puede asignar características de la interacción del estudiante con características de actividades de aprendizaje. Hay formatos que soportan esta técnica como el IMS-LD

Se ha descrito anteriormente otras técnicas de recomendación en (Burke, 2007), algunas de estas son subtipos de estas técnicas por ejemplo la demográfica es subtipo de la técnica de filtrado colaborativo basada en estereotipos o demografía y la basada en conocimiento al utilizar un conocimiento anterior entre las necesidades del usuario y el ítem que se va a recomendar es una subclase de las técnicas basadas en contenido.

1.7. Anotaciones Semánticas

Este tema es importante dentro del proyecto de investigación para el desarrollo de la parte semántica del agente recomendador ya que es un agente que utiliza la técnica basada en atributo y como subtipo utiliza la técnica de recomendación basada en conocimiento y debido a que estos atributos deben estar categorizado o estructurados, la mejor manera de hacerlo es convertirlos a ítems con contenido semántico es decir crear metadatos a partir de la información que se tiene en el entorno.

Para este fin se ha creído conveniente utilizar una herramienta de anotación semántica que ayude a extraer información relevante.

Una anotación semántica es agregar una descripción de algún texto plano para su posterior utilización en aplicaciones en la web.

Tiene la ventaja de estructurar información para que el usuario mejore sus búsquedas además ayuda a que un documento sea procesado de mejor manera por una máquina.

En la búsqueda de herramientas de la web semántica que colaboren con este propósito se ha encontrado el documento (Uren, y otros, 2005), en donde se describe algunos requerimientos de la gestión de conocimiento (KM) con respecto a las anotaciones semánticas y realiza un estudio de cada herramienta en base a estos requerimientos.

Este documento también hace una introducción de la importancia de las anotaciones semánticas en la gestión del conocimiento no solo económicamente sino también a nivel educativo, ya que estas tecnologías proveen de búsquedas de información más refinadas y además la estructuración de información en empresas acorde al contexto.

Definen que “un documento inteligente es aquel que conoce de su propio contenido con el fin de que procesos automatizados sepan qué hacer con este conocimiento”

Lo que se propone en la web semántica es anotar el contenido de un documento usando información semántica desde un dominio ontológico. Como resultado se obtiene documentos o páginas web con etiquetas interpretables por máquina que proveen de un material fuente con el cual agentes y servicios web pueden trabajar, en este caso el agente recomendador propuesto para la universidad puede trabajar con estos recursos.

Los sistemas de anotación semántica tienen beneficios de dos tipos: mejoran la recuperación de información como también mejoran la interoperabilidad.

El primer beneficio está relacionado a consultas o búsquedas del usuario, se explota ontologías y se realiza una inferencia desde datos heterogéneos. El segundo beneficio tiene que ver con estructuración de información, en las empresas se puede manejar gran cantidad de datos que no están estructurados al realizar una estructuración los miembros de la empresa pueden interactuar de mejor manera con esta información.

1.7.1. Frameworks para anotación semántica

Al seguir con la revisión de este documento, hay dos frameworks , de manera general un framework es una aplicación que integra varias tecnologías para brindar servicios y también colaborar para la mejora de sus funcionalidades, en este caso se discute de dos frameworks para integrar recursos anotados semánticamente, un framework puede ser utilizado de diferentes maneras y por diferentes herramientas.

Annotea: es un proyecto de la W3C⁴, que especifica una infraestructura para la anotación semántica de documentos de la web haciendo énfasis en la colaboración entre usuarios de estas anotaciones. Los estándares abiertos que promueve I W3C es un factor importante en este tipo de proyectos ya que favorece la extensibilidad y a interoperabilidad de los sistemas que utilizan este framework. Una debilidad es que utiliza documentos en lenguaje html y xml, lo que limita la gestión de conocimiento al no poder manejar otros formatos, pero se ha trabajado en ello y se ha

⁴ W3C: Word Wide Web Consortium

creado Xpointer, herramienta que permite el análisis de documentos para su anotación semántica.

Cream: incluye una infraestructura ideal para las anotaciones semánticas con una interfaz en donde se editan estas anotaciones, tiene soporte para anotaciones automáticas, gestión de documentos y servidor con interfaz para anotaciones. A diferencia de Annotea, este realiza anotaciones en la web profunda, es decir en las bases de datos de sitios web. Algunas herramientas que utilizan este framework son Onto-Mat y S-cream.

1.7.2. Herramientas de anotación semántica.

Debido a que el agente construido en esta investigación necesita de alguna herramienta que categorice las actividades del estudiante, es decir haga una anotación de estos atributos para poder recomendar los recursos adecuados a las necesidades que el estudiante tiene en base a sus actividades en el EVA.

Hay que realizar un estudio de la web semántica y las herramientas que tiene para las anotaciones semánticas.

El trabajo que se citó anteriormente ayuda con una comparación de herramientas de acuerdo a los siguientes requerimientos:

Formatos Estándar: este requerimiento tiene que ver con los formatos que se manejan en las diferentes herramientas, es preferible que se tengan formatos normalizados para no sufrir en lo posterior de restricciones de funcionalidad por formatos no estandarizados. Además este requerimiento en sistemas de anotación semántica permite compartir anotaciones, acoger documentos heterogéneos. Una entidad que trata de estandarizar internacionalmente los formatos es la W3C, algunos de los formatos de la actualidad son: OWL, RDF.

Diseño Colaborativo Basado en Usuario: este requerimiento tiene relación a la interfaz que se presenta al usuario para realizar las anotaciones, es preferible que sea de fácil uso y que el usuario pueda editar sus documentos y anotaciones. El diseño también tiene que permitir la colaboración entre usuarios, ya que es un aspecto clave en la gestión de conocimiento al contribuir profesionales de diferentes ramas con documentos inteligentes.

Soporte de Ontologías: este requerimiento es muy importante ya que una ontología en general representa conocimiento y la herramienta tiene que manejar diferentes ontologías y de varios campos de conocimiento.

Soporte de formatos heterogéneos de documentos: por lo general los estándares web semánticos asumen que un documento en la web esta anotado en el lenguaje web nativo html o xml esto dificulta la gestión de conocimiento. Los documentos tienen que estar en diferentes formatos como gráficos, presentaciones, texto, etc.

Evolución de Documentos: este requerimiento está relacionado al cambio repentino de documentos, la herramienta tiene que soportar estos cambios y realizar las anotaciones consistentes del documento.

Almacenamiento de anotaciones: este requerimiento tiene que ver con guardar las anotaciones en un repositorio diferente al del documento, ya que algunas herramientas tienen integrado en el mismo documento se hacen las anotaciones.

Al tener repositorios diferentes se maneja de mejor manera tanto documentos como anotaciones

Automatización: es un requerimiento crucial en una herramienta de anotación semántica, ya que las anotaciones de preferencia tendrían que ser procesadas de manera automática por una aplicación, este proceso es necesario cuando se tiene una gran cantidad de documentos o se trabaja en ambientes donde se generan bastantes documentos y no se dispone de tiempo para hacerlo manualmente.

Se puede encontrar dos tablas que hacen una comparación en base a los 7 requerimientos de las herramientas de anotación semántica que existen en (Uren, y otros, 2005)

Otro trabajo importante acerca de herramientas web semántica en donde se incluye una comparación de herramientas de anotación semántica es (García, 2008), se toma en cuenta también que herramientas son de código libre.

En base a estos dos trabajos investigativos de las herramientas de anotación semántica se ha realizado una tabla con las posibles características que puedan servir al desarrollo de la parte semántica del agente recomendador, solo se ha tomado en cuenta las herramientas que cumplen el requerimiento de ser automáticas.

Tabla 6 Herramientas de anotación semántica, expectativas dentro del proyecto

Herramienta	Requerimientos Cumplidos	Software libre	Cumple con expectativas del proyecto
Lixto	Cumple con el requerimiento de manejar formatos estándar. Para la automatización utiliza wrappers	No es de código abierto	No cumple con las expectativas ya que es una herramienta propietaria
MnM	Cumple con todos los requerimientos enfocados en la gestión del conocimiento	Si para fines no comerciales	Si cumple con las expectativas del proyecto
Melita	Cumple con 6 de los requerimientos, ya que no tiene almacenamiento de anotaciones semánticas	No	No cumple por la licencia
Smore	Cumple con apenas 5 de los requerimientos no posee almacenamiento de anotaciones	Si	No cumple al no tener almacenamiento de anotaciones semánticas necesarias para el proyecto
Shoe Knowledge annotator	No cumple con la evolución de documentos	Si	Cumple con las expectativas del proyecto
Kim	No cumple con la evolución de documentos	Si para fines no comerciales	Cumple con las expectativas

Una vez que se ha filtrado cada herramienta en base a los requerimientos y expectativas del proyecto se ha investigado sobre las que pueden satisfacer estas expectativas.

MnM: es una herramienta de anotación basada en ontologías, permite la anotación de páginas web con contenido semántico tanto de forma automática como de forma manual. Tiene integrado un navegador web el cual se puede editar ontologías y brinda API's abiertos para enlazarse con servicios de otras ontologías. En general MnM cumple con todos los requerimientos de un anotador semántico, es desarrollada por Knowledge Media Institute (KMI). Se realizó intentos por obtener el software para su posible uso en este proyecto pero no hubo la posibilidad no estaba disponible en la página de este proyecto.

Shoe Knowledge Annotator: se revisó la página concerniente a este proyecto, pero la información acerca de uso es muy confusa además que a primera vista parece limitada la herramienta para los fines del proyecto.

KIM: aunque no cumple con algunos de los requerimientos de los anotadores semánticos, esta herramienta cuenta con bastante información, en su página web tiene una descripción bien estructurada de sus métodos y arquitectura. No solo tiene la utilidad de hacer anotaciones

semánticas sino que maneja varias posibilidades para la gestión del conocimiento. Se bajó el software para su posterior instalación que no fue muy compleja.

Esta plataforma se encarga de hacer anotaciones semánticas y guardarlas en un amplio repositorio del conocimiento, tiene sus propias ontologías, en caso de que se quiera incorporar ontologías a la plataforma se requiere de una licencia.

De las varias herramientas de la cuales se buscó información esta cuenta con toda la información necesaria además de tener otras características para la ingeniería del conocimiento valido para un agente recomendador basado en conocimiento.

Por esta razón se ha decidido trabajar con esta plataforma, en si no es Kim el anotador semántico sino que Kim utiliza GATE (General Architecture for Text Ingenieering) dentro de su arquitectura para realizar las anotaciones semánticas y estas se pueden realizar automáticamente por medio de ANNIE que a su vez es una herramienta de extracción de información. GATE fue desarrollado por el grupo de NLP (Natural Language Procesing) y es de licencia libre, tiene una gran popularidad entre los sistemas de anotación semántica.

A continuación se revisa todas las características de la plataforma Kim, su arquitectura y el proceso para manejar conocimiento.

1.8. Plataforma KIM

Se ha decidido utilizar la plataforma Kim debido a su completo conjunto de características especialmente el de anotaciones semánticas muy útil en el proyecto.

Además Kim es una herramienta de código abierto para proyectos de investigación y de pago para proyectos comerciales.

Según su página oficial (Sirma Group , 2013) Kim de forma general tiene las siguientes cualidades:

- Crear enlaces semánticos entre documentos, datos, modelos de dominio y linked data.
- Encontrar mención a entidades, relaciones y características en un texto.
- Buscar y Navegar en la información de diferentes maneras.

Kim es una plataforma de gestión de conocimiento semántica y modular, que crea conexiones semánticas entre documentos, datos, modelos de dominio y linked data.

Encuentra menciones de entidades, relaciones y características en los textos y los vincula a un conocimiento de fondo. Maneja recursos de información complejos y heterogéneos, realiza recuperación de documentos en función de relaciones externas a esos documentos y permite varias opciones de búsqueda.

Lo interesante de esta herramienta es la posibilidad de crear anotaciones semánticas de cualquier texto, la idea de la solución es realizar estas anotaciones semánticas en el conjunto de tareas pendientes con el fin de extraer información relevante y en forma de metadato.

Kim permite utilizar sus herramientas a manera de servicio por cualquier aplicación para mejorar su funcionamiento, además la plataforma está equipada con ontologías de nivel superior y bases del conocimiento que cubren una gran cantidad de entidades importantes.

Las ontologías y bases del conocimiento son manejadas con tecnología semántica de vanguardia, diferentes estándares, repositorios RDF y razonamiento.

Técnicamente la plataforma permite que aplicaciones basadas en Kim use esta para anotaciones semánticas automáticas, recuperación de contenido bajo restricciones semánticas y consulta y modificación de ontologías y bases de conocimiento.

Un documento que ayuda a entender la arquitectura del KIM y todas sus características es (Popov, Kiryakov, Kirilov, Manov, Ognyanoff, & Goranov, 2010)

Este documento guía la descripción y elaboración de la parte semántica del agente recomendador, además de denotar las partes de la arquitectura que se puede utilizar en la aplicación.

Como en el caso de este proyecto de investigación la gran cantidad de metadatos permite a las aplicaciones semánticas emerger y tener una amplia aceptación, buscando nuevas e innovadoras formas de acceso relacionando y utilizando estos metadatos.

Existen diferentes tecnologías de IE (Information Extraction), que permiten el reconocimiento de entidades nombradas dentro de un texto, incluso relaciones, eventos y posibles escenarios en los que actúan. Por lo tanto los metadatos pueden ser asignados a un documento, presentando parte del contenido de su información adecuado para su posterior procesamiento. Los metadatos pueden variar desde una referencia directa al autor del documento, valores de dinero mencionados en el documento, ciudades o compañías dentro del texto, en fin esto propone una

cuestión importante de cómo hacer estos metadatos legibles a la maquina con el fin de una estructura efectiva, descubrimiento automático , integración y reusabilidad.

A esto apunta la investigación a generar metadatos debidamente estructurados para que sean legibles al Agente Recomendador y los utilice para el descubrimiento automático de recursos relacionados que debe recomendar.

Proceso de anotación semántica en KIM

La idea de anotación semántica parte de que teniendo varias entidades nombradas⁵ en algún texto, se le asigne una descripción semántica generando de esta manera un metadato que provea información de clase y de instancia de la entidad referida en el texto.

En el proyecto en la información de las tareas pendientes como es su descripción y título se pueden encontrar estas entidades nombradas, de aquí surge uno de los primeros problemas de la investigación que para poder encontrar estas entidades, las tareas pendientes tienen que estar bien descritas.

Pero el potencial de la plataforma permite la anotación semántica automática que a su vez posibilita nuevas aplicaciones: de indexación y recuperación, de clasificación, de generación de avanzados metadatos, sin problemas de recorrido entre texto no estructurado y conocimiento relevante. Se permite la anotación semántica de cualquier texto como son páginas web, bases de datos, texto plano, etc.

Arquitectura de la plataforma KIM

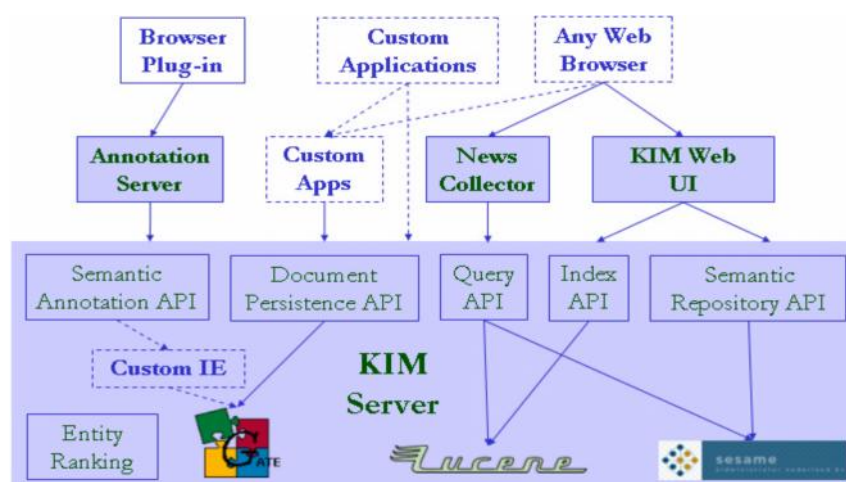


Gráfico 5 Arquitectura de la plataforma KIM (Popov, Kiryakov, Kirilov, Manov, Ognyanoff, & Goranov, 2010)

⁵ Entidades Nombradas: son textos importantes como autores, locaciones, valores, compañías.

La arquitectura de Kim consta de lo siguiente: Kim Ontology (KIMO), Base del Conocimiento, Kim Server (API para acceso remoto, embebido, e integración) y terminales (Plugin para el navegador, interfaz web de usuario con varios métodos de acceso, y explorador de conocimiento para navegar por las bases de conocimiento). El API de Kim permite anotaciones semánticas, indexación y recuperación e infraestructura. Las ontologías y las bases del conocimiento se guardan en repositorios de tecnología de punta y estándares de la web semántica, incluyendo repositorios RDF(s), (SESAME) y Soporte para Ontologías.

Kim provee de una infraestructura madura para una extracción de información escalable y personalizada como también la anotación y manejo de documentos basados en GATE, La máquina de recuperación de información LUCENE ha sido adoptada para indexar los documentos por tipo de entidades y medir la relevancia de acuerdo a la entidad. Cabe indicar que la plataforma KIM es de trabajo independiente como lo son LUCENE y SESAME.

KIM API

El API de KIM nos permite realizar lo que necesita nuestro agente que es las anotaciones semánticas, indexación y recuperación de información en base a como se quieran manejar los documentos en este caso la información de las tareas pendientes y la navegación en la base del conocimiento (KB), el API está ubicado en la mitad de la arquitectura donde se muestran todos sus módulos: El API del Anotador semántico permite realizar las anotaciones semántica de un documento con respecto a las ontologías de KIM y su Base de Conocimiento, también permite el manejo de estas anotaciones ya que las guarda mediante el API de Persistencia de Documentos. El API de indexación es basado en la máquina de recuperación de información LUCENE, la cual ha sido modificada para permitir la indexación en base a las entidades nombradas.

El API de consulta puede ser pensado como un API de recuperación de información semántico, el cual permite la búsqueda tradicional de palabras claves, como también el acceso mediante recorrido de ontologías. Provee una infraestructura para poder hacer búsquedas combinadas de palabras y entidades, búsqueda de palabras y búsqueda de patrones de entidades. Debido a la indexación respecto a entidades se puede requerir todos los documentos referidos a esa entidad a través de restricciones como su nombre, clase y atributos. Mejor aún se puede especificar u patrón entre entidades y las relaciones de las mismas y consultar documentos referenciando a tales entidades.

Por ultimo para el acceso a la base del conocimiento se tiene el API del Repositorio Semántico donde se utiliza un conjunto de métodos e infraestructura.

KIMO

La razón de ser de la ontología de la plataforma es establecer un mínima ontología, pero suficientemente adecuada para realizar las anotaciones semánticas en relación a un dominio en general.

Es por esto que la plataforma cuenta con una ontología propia que ha sido creada desde cero y se ha realizado en base a trabajos más grandes como OpenCyc, Worlnet, DOLCHE, entre otras. Se ha tratado de que sea sencilla pero de nivel superior.

Bases del conocimiento

Lo sofisticado de la Extracción automática de información no es muy relevante, lo que hay que prestar atención es como beneficiarse de una base de conocimiento de partida para representar a las entidades importantes en el contexto, de ahí poblar esa base de conocimiento.

El Kim cuenta con esta base de conocimiento con un total de 80000 entidades de general importancia donde se tendrá acceso mediante el API correspondiente.

Extracción de información semántica

La esencia de KIM es reconocer entidades nombradas en base a su ontología, relaciona la NE (Named Entity) con su clase en la ontología, la instancia de esta entidad tiene un identificador (URI) el cual permite que la anotación sea enlazada con el individuo exacto en la base de conocimiento.

La extracción de información en KIM se basa en el framework GATE⁶ la cual ha demostrado madurez, extensibilidad e independencia de otras herramientas de NLP (Natural Language Process). Posee varias funcionalidades de manejo de documentos y componentes genéricos de NLP como son: Tokenizer, Part-of-Speech Tagger, y Sentence Splitter.

⁶ GATE (General Architecture for Text Engenieering)

Estas capas de procesamientos son proporcionadas por la plataforma GATE, junto con el patrón de coincidencia de gramáticas, correferencia de entidades y otros permite la creación de aplicaciones sofisticadas de extracción de información

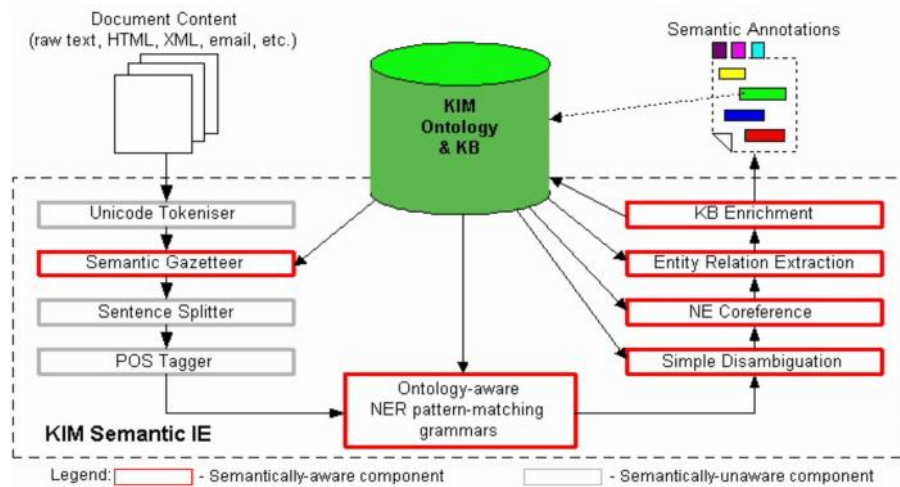


Gráfico 6 Diagrama de Flujo de Extracción de Información Semántica de KIM (Popov, Kiryakov, Kirilov, Manov, Ognyanoff, & Goranov, 2010)

En el gráfico 6 se puede observar el flujo desde que se ingresa el texto hasta que este ha sido procesado y como resultado se devuelve las anotaciones semánticas

1.9. Agentes inteligentes

Las tecnologías de la web semántica permiten que los agentes puedan comprender e integrar varias fuentes de información.

Esta información es más legible por el agente que la procesa con el fin de satisfacer algún propósito o necesidad del usuario. Debido también a que esta información está mejor estructurada beneficia a la interoperabilidad, es decir que la información tiene independencia de los aplicativos y puede ser utilizada por un agente y transmitida a otro agente para conseguir un objetivo en común en un entorno multiagente.

En el proyecto de tesis llevado a cabo el agente recomendador se vale de las tecnologías de la web semántica para cumplir su cometido, esto es utilizar ingeniería del conocimiento para generar recomendaciones a través de la inferencia entre la información de actividades que el estudiante tiene que realizar y recursos relacionados a esas actividades, de esta manera el agente realiza sugerencias de recursos para que el estudiante cumpla con las actividades subidas en su entorno de aprendizaje.

Según el libro (Russell & Norving , 2004) un agente “es cualquier cosa capaz de percibir su medio ambiente con la ayuda de sensores y actuar en el medio con la ayuda de actuadores”

Algunas de las características que deben tener los agentes son:

- Deben ser parte de un ambiente
- Deben censar su entorno y actuar sobre el
- Deben responder a los objetivos para los que fueron diseñados.

En (García, 2008), se manifiesta que toda la potencialidad de la web semántica no se hace efectiva hasta que un agente utiliza y procesa la información estructurada de la web semántica lo que hará de la web una mejor tecnología y con procesos automáticos, cada día son más los agentes que se implementan para solventar algunos de los problemas de la sociedad.

1.10. Trabajos relacionados

- En el trabajo de (Itmazi, 2005), se desarrolla un proyecto acerca de dar flexibilidad al aprendizaje electrónico (e-Learning), mediante la investigación de los diferentes sistemas de gestión de aprendizaje, en el que es seleccionado Moodle, se analizan sus cualidades y desventajas, una de estas desventajas es que carece de un sistema Recomendador, por esta razón se propone y construye un sistemas Recomendador basado en técnicas de recomendación híbrida que mezcla varios tipos de recomendadores, se elabora un esquema de la aplicación y se explican cada una de sus fases para el proceso de recomendación de cursos. Este trabajo es de mucha ayuda, ya que aporta teóricamente con información del Moodle, Agentes Recomendadores y también nos aporta de forma práctica ya que todo el código está documentado dentro del mismo trabajo, podemos analizar el código y adecuarlo a nuestro proyecto.
- En el trabajo de tesis (Gerling, 2009), se diseña y construye un agente Recomendador para que busque información en un repositorio de objetos de aprendizaje y los presente según el perfil que tenga el usuario. Se utiliza una arquitectura multiagente, es decir cuenta con varios agentes entre esos agentes está el Agente Recomendador, el cual utiliza una arquitectura BDI (Belief Desire Intention) según la investigación “Este modelo está basado en la representación explícita de las creencias del Agente (B), que representan el estado del entorno, sus deseos (D), representando sus motivaciones, y las intenciones (I), que modelizan sus objetivos o metas ” Este proyecto nos puede ayudar a ver un modelo de esta arquitectura y verificar si es aplicable a

un sistema de gestión de aprendizaje donde el repositorio serían los datos que genera este sistema.

- Un trabajo que está estrechamente relacionado con este proyecto de tesis es el elaborado en (ROMA Project, 2007), en el cual se experimenta simulando una red de aprendizaje en Moodle en su versión 1.5.4.1 y se realiza un Personal Recomennder System (PSP), el cual es encargado de recomendar actividades de aprendizaje de un curso de Introducción a la psicología, de este trabajo hemos extraído una gráfica con los componentes principales de Agente Recomendador y se ha descrito de forma general, la documentación de este proyecto viene con el instalador denominado “moodle-isis” el cual tendremos que probar para analizar en que nos puede ayudar con nuestro proyecto, trabaja con una técnica de filtrado colaborativo.

- En el trabajo (Guzmán, 2009), se plantea la creación de un sistema Recomendador para aconsejar a usuarios acerca de destinos turísticos de la ciudad de Valencia, aunque sale del dominio educativo en el cual se centra nuestra investigación es importante por las definiciones de la diferentes técnicas de recomendación que en él se presentan, además de la creación de un perfil de usuario determinando sus gustos y preferencias

- Un proyecto que también sale del dominio universitario, es el planteado en (Carrillo, Llovet, & Rodríguez, 2010), pero está orientado a la educación de pacientes con la enfermedad de diabetes, lo interesante de este trabajo es que se realiza e implementa en el sistema de gestión de Aprendizaje Moodle, sistema que nos compete, además nos da una alternativa de desarrollo como es la de utilizar el programa “jcolibri” hecho en java, podemos analizar si se puede aplicar a nuestro contexto.

Conclusión

De todo el estado del arte revisado y que puede ayudar a la construcción del agente recomendador se ha tomado en consideración el siguiente esquema que detalla todos los elementos acogidos y tomados en cuenta para el agente.

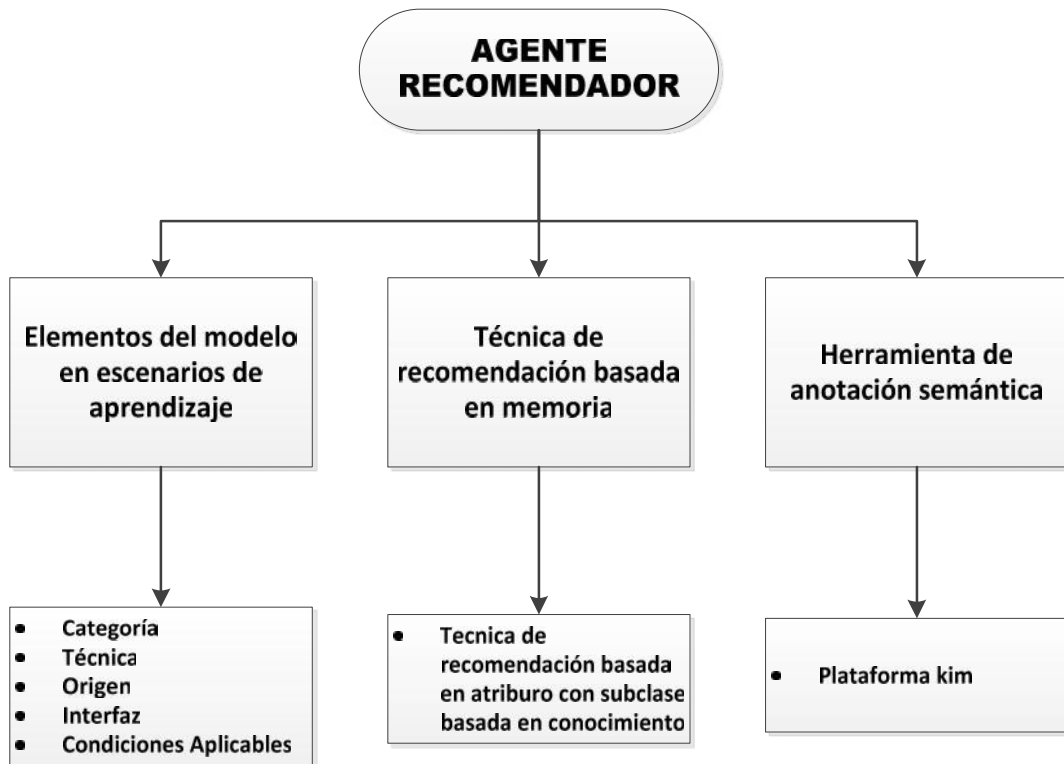


Gráfico 7 Esquema del agente en conclusión con estado del arte

CAPITULO 2: IDENTIFICACIÓN DE TAREAS PENDIENTES EN EL EVA

El siguiente capítulo está orientado a observar la interacción del estudiante con el EVA, para identificar las tareas pendientes, debido a que en la parte de construcción del agente se necesita de esta información para generar las recomendaciones.

Se describe de forma general el entorno virtual de aprendizaje de la UTPL, con una herramienta de gestión de base de datos se monitorea las actividades del estudiante, hay que mencionar que no se trabaja sobre todas las actividades sino solo en dos que han sido consideradas para el alcance de este proyecto por ser las más utilizadas en el EVA : tareas y foros.

Se ha considerado también revisar el modelo entidad relación de Moodle con el fin de observar que entidades intervienen al momento de realizar una actividad, lo que ayuda a saber que tablas monitorear y ver los campos que cambian.

Al final de este capítulo resultado de la identificación de las tareas pendientes se elabora un bloque en el EVA que muestra las tareas pendientes al estudiante que es el primer paso a la construcción del agente recomendador, por último se describe el modelo relacional del bloque REA del EVA, que también es necesario para la obtención de los recursos a recomendar.

2.1. Entorno virtual de aprendizaje en la UTPL

La Universidad Técnica Particular de Loja cuenta con un entorno virtual de aprendizaje EVA tanto en su modalidad presencial como en su modalidad abierta, en este entorno se lleva a cabo varias actividades académicas, sirve como ayuda de profesores y estudiantes para desarrollar los contenidos de los distintos cursos o materias que se dictan en cada periodo de la universidad.

De todos los entornos virtuales de aprendizaje que existen en la actualidad en la UTPL se ha implementado la plataforma Moodle por su popularidad, se ha tenido que adaptar algunas características a esta plataforma para que pueda funcionar adecuadamente uno de esas adaptaciones es el REA, un bloque que permite subir recursos educativos abiertos a la plataforma. A través del tiempo de uso de esta plataforma se han hecho más modificaciones que tienen que ver con la matriculación del estudiante a los cursos, bloques de ayuda al estudiante, son tantos cambios que se han realizado que en una instalación normal de moodle cuenta con 215 tablas en su base de datos, mientras que en el EVA de la universidad se cuenta con alrededor de 415 tablas.

2.2. Moodle (Modular Object Oriented Dynamic Learning Environment)

En (Itmazi, 2005), se menciona que el acrónimo en un principio significó: “Entorno de aprendizaje orientado a objetos y modular”, por eso se lo denominó de esa manera, se trata de un CMS (Course Management System), también se lo conoce como uno de los más populares LMSs o como un entorno virtual de aprendizaje (VLE).

En la Universidad Técnica Particular de Loja todos los estudiantes lo conocen como la última denominación traducida al español Entorno Virtual de aprendizaje (EVA).

Moodle tiene una prestigiosa popularidad en todas las instituciones educativas del mundo, ya que tiene la funcionalidad de crear y administrar cursos en sitios web y los estudiantes los puedan desarrollar en línea a través del internet.

Para trabajar con Moodle se puede instalar en cualquier servidor o utilizar cualquier hosting gratuito en la web.

Moodle se distribuye de forma gratuita como software libre Open Source, bajo licencia pública GNU, es decir el usuario puede copiar, utilizar y modificar Moodle siempre que acepte la condición de proporcionar el código fuente a otros, no modificar o eliminar la licencia original y aplicar esta misma licencia a trabajos derivados del software Moodle.

Esta herramienta es muy potente ya que permite llevar a cabo diferentes actividades de aprendizaje en la UTPL es utilizado para llevar cursos completos en línea y ha sido implementado en las dos modalidades presencial y a distancia, aunque en esta última se trabaja mucho más con la plataforma ya que los estudiantes llevan las actividades de sus cursos completamente en el sitio además rinden pruebas y su calificación se relaciona directamente con su trabajo en el EVA, mientras que en la modalidad presencial se ha ido implementando poco a poco, para llevar a cabo tareas extra clase hasta implementarlo completamente.

Moodle tiene funciones de comunicación entre estudiantes, interacción alumno- profesor como son foros, chat, entre otras que definiremos más adelante.

Su estructura modular es adecuada para realizar modificaciones o creaciones con las cuales es posible aumentar funcionalidades a la plataforma, estas son de varios tipos.

En la página oficial de la Plataforma Moodle, describe que puede ser instalado en cualquier computador donde se ejecute PHP y soporte bases de datos tipo SQL.

Por lo general es fácil de instalar y mantener para producir cursos en la web, tiene un diseño que facilita el desarrollo de pedagogías modernas e incluye diferentes módulos para la interacción del estudiante con esta plataforma.

En la UTPL, la adaptación de moodle especialmente a la modalidad abierta ha representado varias ventajas tanto a la institución como a los estudiantes, se han acortado las distancias con tecnología, ya que ahora cualquier usuario que quiera seguir una preparación a distancia depende de un ordenador y una conexión a internet.

Su desarrollo fue iniciado por Martin Dougiamas, es un proyecto activo y en constante evolución, así lo muestra sus diferentes versiones y la cantidad de idiomas en que se ha creado el paquete.

Moodle puede ser utilizado de forma libre, lo que incentiva cada día a más personas a sumarse a esta plataforma y desarrollar sitios mucho más personalizados y con diferentes características.

2.3. Características de EVA

Arquitectura vista desde administrador del EVA

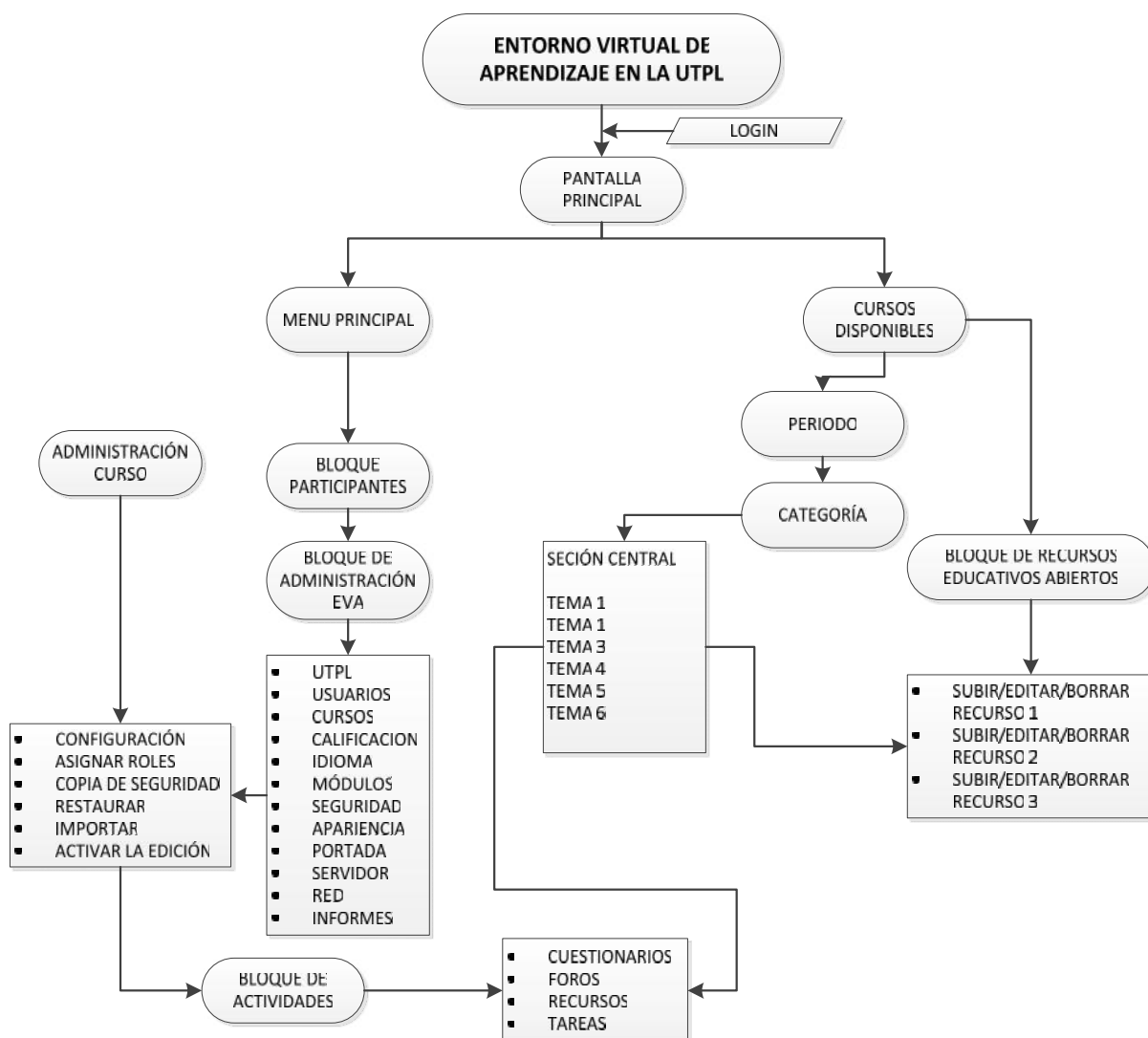


Gráfico 8 Arquitectura del entorno virtual de aprendizaje de la UTPL

Con la ayuda de una arquitectura ya descrita en (Itmazi, 2005), se ha elaborado una propia para el entorno de la universidad, con vista de administrador

En la primera parte de la arquitectura se muestra el ingreso al sitio, esto depende de cómo el administrador haya configurado el sitio o las reglas del establecimientos donde se lo está utilizando, el administrador es el que fija los diferentes roles de la plataforma, el rol de administrador se obtiene al momento de instalar la plataforma, según el rol se puede ingresar como administrador, profesor y estudiante, si es que no se tiene estos roles podemos ingresar como invitados con posibilidades de tan solo observar el sitio y los cursos, se puede presentar también un formulario de registro esto ya depende de la configuración que se le ha dado al sitio, en el caso de esta arquitectura el ingreso es como administrador por lo cual el menú principal permite realizar algunas configuraciones de todo el sitio, al ingresar a los diferentes periodos y

las materias se puede crear cursos y generar actividades para ser desarrolladas por los estudiantes, podemos establecer fechas límites de entrega de tales actividades, calificar, modificar los curso, en fin describiremos cada una de estas actividades en lo posterior.

La arquitectura del EVA se centra en el desarrollo de un curso en cualquier periodo activo de la universidad.

Al dar un vistazo de un curso en el entorno de pruebas de la UTPL, se puede observar todos los elementos que intervienen en la arquitectura.



Gráfico 9 Captura de pantalla curso en el EVA

La imagen muestra la estructura general de un curso del entorno virtual de aprendizaje de la UTPL, se puede apreciar que en la parte central se desarrolla todo el curso y es en donde el profesor coloca las diferentes actividades que el estudiante tiene que realizar durante el periodo de tiempo que toma la materia.

Se puede observar también que el entorno contiene lo que se denomina bloques en las partes izquierda y derecha, estos tienen información adicional del curso como son que participantes tiene el curso, las diferentes actividades, en este caso porque se ha entrado como administrador se tiene el bloque de administración y también se tiene el bloque REA utilizado en lo posterior en esta investigación.

Esto describe como se ha implementado el Moodle en la universidad, pero todas las características de esta plataforma son descritas en su página oficial (Moodle, Moodle), que se detallan a continuación.

Estructura básica del EVA

Como ya se mencionó la estructura general de un entorno que utiliza Moodle se encuentra organizada en base a los cursos. Los cursos son páginas o áreas dentro de Moodle, donde el profesor puede presentar sus recursos o actividades de aprendizaje al estudiante.

Estas páginas constan de tres secciones dos laterales y una central. Todas las actividades e información del curso es presentada en la parte central de la página, mientras que información adicional del curso y otras ayudas son presentadas en las partes laterales de la página, estas secciones laterales se denominan bloques.

El EVA además permite establecer los diferentes roles que van a ser utilizados dentro del curso como puede ser administrador, profesor o estudiante. La presentación de las tres secciones depende de estos roles ya que si se ingresa como administrador, la página muestra algunas utilidades de administración del sitio, como profesor tiene utilidades para elaborar actividades dentro del curso y como estudiante se encuentran las actividades que deja el profesor además de información referente al curso en la sección central.

La administración del EVA en la universidad ha sido responsabilidad del departamento de unidad de virtualización, aquí se lleva a cabo los cursos en línea de las dos modalidades además se trabaja con proyecto de adaptación.

El contenido de un curso se establece por el profesor o el establecimiento educativo, la estructura básica del curso se basa en módulos, describimos a continuación cada módulo.

Modulo recursos

Un recurso es un ítem que el profesor utiliza para dar soporte al aprendizaje, dentro de la Plataforma se maneja varios tipos de recursos como pueden ser archivos o enlaces a un determinado recurso.

Los recursos que se utilizan dentro de moodle son de todo tipo entre los cuales se tiene los siguientes.

- Libro (Book)
- Archivo (File)

- Carpeta (Folder)
- Paquete de contenido IMS
- Etiqueta
- Pagina
- Dirección URL

Adicionalmente también se pueden subir recursos al EVA mediante el bloque REA.

Actividades

Las actividades del curso son la parte primordial del mismo, ya que ayudan al estudiante a desarrollar e interactuar dentro del curso ya sea con sus compañeros o con el profesor, si bien no es un módulo sino un bloque dentro del mismo cada actividad se considera un módulo de Moodle.

En la plataforma de Moodle se cuenta con 14 actividades, las mismas que se las pueden encontrar en el menú de agregar actividades si es que se ingresa con el rol de profesor o administrador.



Gráfico 10 bloque de actividades en Moodle (Moodle, Bloque de Actividades)

De acuerdo a los cursos que se han revisado en el EVA de la UTPL, a la experiencia como estudiante en la modalidad presencial las actividades que generalmente son más utilizadas en el entorno son las siguientes:

- Tareas (Assignments)
- Foros (Forums)
- Lecciones (Lessons)
- Cuestionario (Quizes)
- Recurso (Resources)

El enfoque de este proyecto no abarca todas estas actividades, donde se va a dar soporte al estudiante por lo que no se describirán todas sino las que son especiales para la investigación, se ha decidido tomar las más utilizadas y populares en la universidad las que son foros y tareas, además se utilizan recursos del bloque REA.

Modulo foros (forums)

El módulo de foros es una actividad donde estudiantes y profesores pueden intercambiar ideas posteando comentarios. Existen 4 tipos básicos de foros.

- Un simple foro de discusión: es un único tema de discusión desarrollado en una página donde cada cual emite criterios cortos acerca del tema.
- Foros donde cada persona postea un tema: en este foro cada persona postea su tema de discusión y los demás pueden responder a este tema, este tipo de foros es necesario cuando se requiere que cada estudiante de su punto de vista con un nuevo tema.
- Foros de Pregunta y Respuesta: en estos foros en lugar de iniciar un tema de discusión cada estudiante plantea una pregunta, a la cual pueden dar respuesta todos los demás, sin embargo no pueden ver las respuestas entre si hasta que al final se pueden comparar todas las respuestas y tener una idea del tema.
- Foro Estándar de uso general: es un foro abierto donde cualquiera puede postear temas, este tipo de foros es el más utilizado en la plataforma.

En la universidad los tipos de foros que son generalmente de mayor uso son los dos primeros y el último, se establecen temas de discusión de las diferentes asignaturas, es un medio importante de comunicación entre compañeros y el profesor.

Su configuración depende del profesor y del desarrollo del curso que dicta.

Módulo de tareas (Assignments)

El módulo de asignaciones permite al profesor recoger trabajos de todos sus estudiantes, revisarlos y enviar una retroalimentación incluidas calificaciones. Estos trabajos pueden ser vistos solo por el profesor, los estudiantes no pueden ver trabajos de otros estudiantes a menos que la configuración de la tarea sea en grupo.

Los estudiantes pueden subir cualquier archivo digital como textos, presentaciones, audio, video, entre otros.

Las tareas enviadas por el profesor no necesariamente pueden ser subidas a la Plataforma, sino que hay la posibilidad de que el estudiante escriba directamente en un cuadro de texto en la plataforma Moodle.

2.4. Base de datos del EVA

Como se ha mencionado anteriormente el EVA cuenta con 415 tablas en la actualidad debido a sus continuas adaptaciones y proyectos que se han implementado.

En este proyecto se enfoca en las tablas que ayudan a determinar las actividades de un estudiante, se ha tomado en consideración el modelo entidad relación del Moodle en el Anexo 1, esto aporta con una idea de que entidades intervienen de acuerdo al objetivo de identificar las tareas pendientes.

Según este modelo y el objetivo de logra identificar las tareas pendientes, estas son las entidades principales que se han escogido.

- Estudiantes
- Cursos
- Tareas
- Foros

Cabe indicar que en el modelo entidad relación del EVA, hay distintas modificaciones tanto en tablas como en sus campos, pero de acuerdo al enfoque del proyecto estos cambios no repercuten con el objetivo, se menciona esto debido a que el monitoreo de las tablas ha sido llevado a cabo en una instalación normal de Moodle y la implementación del bloque que muestra las actividades pendiente ha sido en el entorno de pruebas de la UTP

La primera entidad que se toma en cuenta son los estudiantes, en el modelo entidad – relación de esta entidad es descrita por “USER”, la cual cuenta con diferentes tablas en el EVA son un total de 15 tablas, solo se toma en cuenta la tabla “mdl_user” porque es la que contiene todos los datos de los usuarios, las demás son para establecer roles y categorías que no son importantes a la investigación.

Para la entidad Cursos de igual forma existe gran cantidad de tablas, hay que enfocarse en la tabla “mdl_course” porque aquí se guarda toda la información de los cursos, las demás tablas no se consideran para el monitoreo.

En la entidad tareas no hay muchos inconvenientes ya que posee dos tablas en el EVA las cuales son “mdl_assignment” y mdl_assignment_submissions, se lleva el monitoreo en estas dos tablas.

Por ultimo tenemos los foros que poseen gran cantidad de tablas, pero las más importante para realizar el monitoreo son “mdl_forum” y “mdl_forum_discussions”

2.5. Escenario de prueba

Una vez definidas las tablas que hay que monitorear para revisar los cambios en sus campos, se ha decidido crear un entorno de pruebas instalando localmente Moodle, se lo ha realizado sobre esta plataforma inicialmente debido a que su estructura comparada con la estructura del EVA es similar y no afecta las tablas que se han venido incorporando por las diferentes adaptaciones del mismo, para interactuar con otros sistemas.

Después de realizar este monitoreo, se llegara a la conclusión de que tablas o campos sirven para identificar las tareas pendientes, se codificara este resultado para mostrarlo en el entorno de prueba de la UTPL .

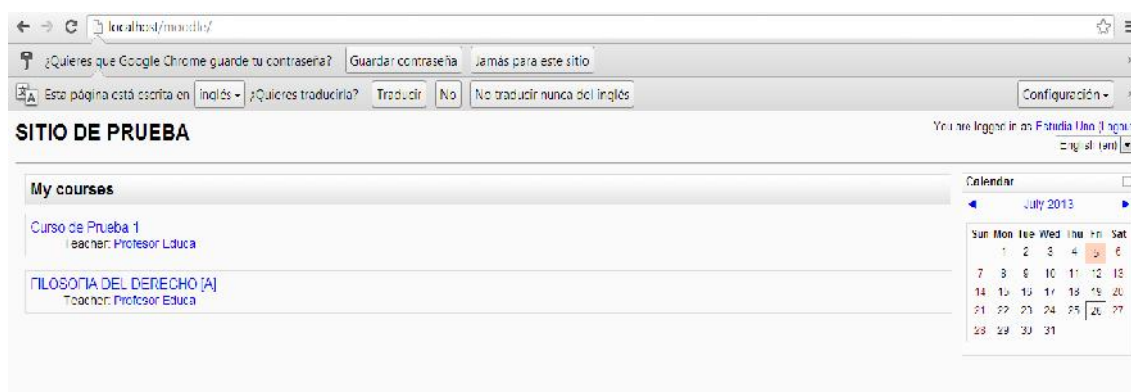


Gráfico 11 Escenario de pruebas

Un curso: el curso se ha implementado con información de un curso real de la UTPL en el cual encontramos gran cantidad de actividades de foros y tareas para el estudiante el curso de ejemplo se denomina “Filosofía del Derecho”

Un Profesor: se ha creado un usuario con el perfil de profesor y se ha designado este rol al curso de “Filosofía del Derecho” el usuario se denomina “Profesor Educa”

Estudiantes: para poder analizar el comportamiento de un estudiante se han creado varios usuarios con los roles de estudiantes y se han agregado estos roles al curso de “Filosofía del Derecho” los estudiantes se denominan “Estudia Uno” y “Estudia Dos”

Actividades: el curso de “Filosofía del Derecho” se encuentran varias actividades que el estudiante debe de realizar hay Foros y tareas.

- **Foros:** Se encuentran dos foros en el curso los cuales son los siguientes:
 - Función práctica de la filosofía del derecho
 - El estado socialista

- **Tareas:** el curso cuenta con 7 tareas:
 - Lectura de presentación del texto
 - Elaboración de un concepto personal de filosofía
 - Lectura del drama de Antígona y de la tragedia de Sócrates
 - Análisis de la teoría sobre la república de Platón
 - Análisis de la teoría pura del derecho
 - Teoría del realismo jurídico
 - Teoría tridimensional del derecho

Luego de crear el curso con sus diferentes características se utiliza la herramienta TOAD for Mysql, la cual se conecta a la base de datos y monitorea los cambios en los datos de las tablas.

2.6. Monitoreo de la base de datos mediante la herramienta TOAD

TOAD es una herramienta de desarrollo y administración de base de datos, permite la creación y ejecución de consultas rápidas a cualquier base de datos SQL.

Se utiliza esta herramienta para realizar las consultas a las tablas de las entidades descritas anteriormente y en base al escenario de pruebas creado con el fin de ver que campos cambian y poder identificar las tareas pendientes.

Se tiene 4 entidades: usuarios, cursos, tareas y foros los cuales se han elegido las siguientes tablas.

Tablas y campos consultados

Una vez instalado el entorno de pruebas con sus respectivos cursos y usuarios, se debe realizar las consultas a las siguientes tablas.

Tabla mdl_user

Esta tabla posee diferentes campos, uno de los más importante para el proyecto es el campo "id" por lo que es el identificador único del estudiante y muestra la relación con otras tablas si es que es tomado como campo en las mismas.

Entonces de esta tabla se considera los siguientes campos:

id: de este campo tenemos que tomar en cuenta los correspondientes a los estudiantes creados en el sitio de prueba para el "estudiante 1" el identificador es 4 y para el estudiante 2 el identificador es el 5

username: son los nombres de usuario con los que se ingresa a los curso, tenemos que considerar los de los estudiantes que "estudiante 1" y "estudiante2"

Tabla mdl_course

Los campos considerados en esta tabla son los siguientes:

id: identificador único de cada curso, nos sirve para verificar relaciones con otras tablas según sus campos, en el caso del curso de "Filosofía del Derecho" el valor de este identificador es 3.

fullname: nombre completo del curso en el caso "Filosofía del Derecho"

Tabla mdl_assignment

En esta tabla se pueden observar todas las tareas subidas al entorno de pruebas y los campos de interés son los siguientes:

id: identificador único de la tarea

course: este campo es muy importante, aquí podemos verificar que esta tabla tiene relación con “mdl_course”, ya que en este campo tiene el valor del id del curso donde se ha enviado la tarea.

name: es el nombre de cada una de las tareas

description: descripción de la tarea, este campo se lo llena en el entorno de prueba cuando se sube la tarea se da una descripción de la misma.

Tabla mdl_assignment_submissions

En la tabla anterior se describe la tarea y los cursos, pero no hay campos que describan la interacción con el usuario, al monitorear esta tabla se encuentra un campo del usuario lo cual demuestra su actividad con esta tabla.

La tabla posee los siguientes campos:

id: es el identificador único de esta tabla

assignment: este campo demuestra la relación de la tabla con mdl_assignment, ya que contiene el valor del id de la misma, describe en que tarea ha sido presentada una respuesta.

userid: demuestra la relación de esta tabla con la de usuarios ya que este valor corresponde al identificador único de los usuarios.

timemodified: establece la fecha que ha sido modificada la presentación de la tarea.

Tabla mdl_forum

Esta tabla describe todos los foros llevados en los cursos sus campos más relevantes son los siguientes:

id: identificador único del foro

course: describe la relación con el curso en donde se encuentra el foro, ya que este valor corresponde al identificador de la tabla mdl_course

type: establece que tipo de foro es

name: nombre del foro

intro: descripción del foro o una introducción del tema a discutir

Tabla mdl_forum_discussions

Al realizar el monitoreo de la tabla mdl_forum no se encontró interacción con el usuario, pero al revisar esta tabla tenemos un campo relacionado al usuario se describen a continuación:

id: identificador único de la tabla

course: demuestra la relación con la tabla de cursos ya que este valor es el identificador único de la tabla mdl_course

userid: se demuestra la relación de esta tabla con la de usuarios ya que este campo tiene el valor de identificador único de la tabla mdl_user

name: respuesta que se ha dado en el foro.

timemodified: fecha en que se modificó la respuesta

2.7. Proceso de identificación de tareas pendientes

Después de monitorear todas las tablas pertinentes mediante consultas a través de TOAD se ha concluido las siguientes relaciones según sus clave o identificador primario:

- mdl_assignment con mdl_course
- mdl_forum con mdl_course
- mdl_assignment_submission con las tablas mdl_assignment y mdl_user
- mdl_forum_discussions con las tablas mdl_forum y mdl_user

Estas relaciones han ayudado a interpretar de una mejor manera la interacción del estudiante en base a sus actividades dentro del EVA, en un principio se pensó que la información de las actividades realizadas por el estudiante estaban en la tablas referentes a las mismas, pero por medio de esta relaciones no se observa un registro del usuario en estas tablas, esto cambio al revisar las tablas relacionadas a los de foros y tareas, ya que en ellas si se muestra un registro del usuario mediante su identificador único las cuales son: mdl_assignment_submissions y mdl_forum_discussions.

Al revisar los demás campos de estas tablas se ha concluido que son las que llevan el registro de las actividades foros y tareas dentro del EVA.

Para poder demostrar esto se ha utilizado los usuarios creados con el rol de estudiante en el sitio de pruebas para con parte de estos estudiantes realizar todas las actividades del curso, en cambio con la otra parte no realizar ninguna de las actividades, lo que se obtiene como resultado es el registro de los usuarios que participaron en todas las actividades en las dos tablas antes mencionadas de los estudiantes que han realizado todas las actividades unos se identifica con el id "4" el que consta en la siguiente imagen .

id *	assignment *	userid *	timecreated *	timemodified *	numfiles *	data1
1	1	4	0	1372442327	1	
2	10	4	0	1374868470	0	hola esta interesante el tema del texto que hace una introduccion a lo c
3	12	4	0	1374868608	0	
4	13	4	0	1374868823	0	socrates un buen teorico del derecho
5	14	4	0	1374869270	0	Platon es tambien un buen cientifico del derecho
6	15	4	0	1374869311	0	Es la teoria que abarca todo la filosofia del derecho
7	16	4	0	1374869426	0	realismo juridico es lo que pasa en nuestra sociedad

id *	course *	forum *	name *	firstpost *	userid *	groupid *	assessed *	timemodified *	usermodified *
1	2	2	Contestacion al Foro	1	4	-1	<input checked="" type="checkbox"/>	1372731423	4
2	3	5	revolucion si filosofia	2	4	-1	<input checked="" type="checkbox"/>	1374868874	4
3	3	6	estado socialista	3	4	-1	<input checked="" type="checkbox"/>	1374869369	4

Gráfico 12 registro de actividad en las tablas mdl_assignment_submissions y mdl_forum_discussions

Como se puede observar en la imagen el registro solo se lleva del estudiante con el identificador 4.

Para finalizar con el proceso bastara con verificar su registro en estas tablas, ya que si no se encuentran quedaran identificadas como tareas pendientes del estudiante.

2.8. Elaboración del Bloque de Tareas Pendientes

Un Bloque es parte de la plataforma de Moodle, generalmente aparecen en la parte izquierda y derecha del entorno. Se utiliza para diferentes funcionalidades que realice el estudiante y el profesor, puede funcionar de manera independiente o interactuar con los diferentes bloques que se encuentren en la plataforma.

Para la creación de bloques existen dos formas, la una es dentro de la plataforma como administrador o profesor de un curso en específico y la otra es el desarrollo de un bloque mediante php.

En este trabajo se utiliza la segunda opción ya que se requiere crear un módulo con características propias y poder configurarlos según las necesidades del agente Recomendador.

En la página oficial de Moodle, se describe los pasos para la creación de un bloque, se ha construido el bloque para mostrar las actividades pendientes utilizando esta información.

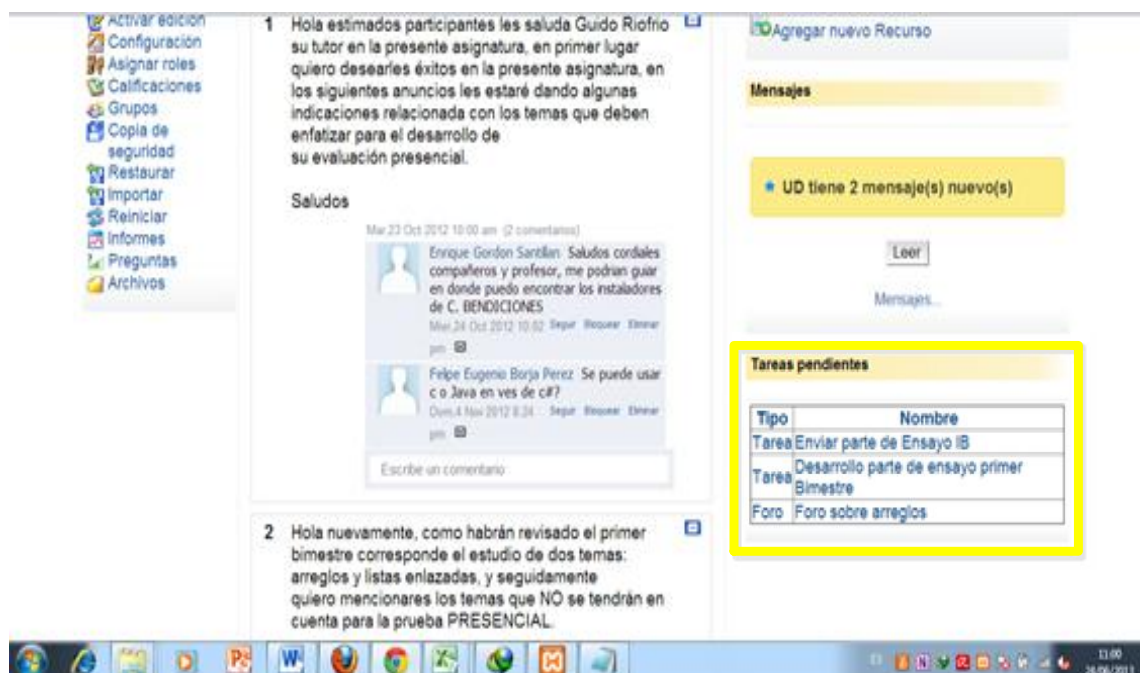


Gráfico 14 Bloque de tareas pendientes en el EVA

2.9. Recursos educativos abiertos

Como primer paso del proyecto se ha identificado las tareas pendientes del estudiante, pero ahora hay que dar solución recomendando recursos para que cumplan con estas tareas. Para esto se tiene que buscar de donde se puede sacar recursos y como en la universidad se ha implementado un proyecto acerca de recursos educativos abiertos conviene utilizar estos recursos para realizar las recomendaciones.

Los recursos educativos abiertos son material de cualquier tipo que se ofrecen abiertamente al público para colaborar con la educación, en la universidad se ha trabajado en proyectos de recursos abiertos desarrollando o colaborando con sitios donde se ubica material realizado por la Universidad y que se distribuye con licencia Creative Commons, en el EVA también se ha contribuido con el desarrollo de un bloque el cual permite la creación, edición y eliminación de material educativo el cual es subido como apoyo de un curso en el entorno.

Este bloque se ha denominado REA (recursos educativos abiertos), con esta herramienta el usuario con rol de administrador o profesor ayuda a los estudiantes subiendo material al curso como un apoyo educativo y con contenidos sobre temas tratados en tal curso.

Para la implementación del bloque con el modelo datos del mismo se han creado 11 tablas en el EVA, con el fin de guardar información del material que se sube se describe cada tabla a continuación

- **rea_recursos:** Cada vez que se crea un nuevo REA se genera una nueva entrada en esta tabla la cual guarda entre otras cosas el identificador único del recurso educativo, el curso, usuario y país desde donde ha sido creado; además una variable de tipo boolean que especifica si el recurso educativo ha sido eliminado (borrado lógico).
- **rea:** Guarda cada instancia del recurso educativo, específicamente el periodo, curso y usuario que lo creó; así como también el ranking que obtuvo en determinado tiempo.
- **rea_item_autor:** A un REA puede estar asociados más de un autor, en esta tabla se asocian los autores con el recurso educativo.
- **rea_keyword_recurso:** Almacena todas las palabras relevantes que permiten describir al recurso educativo.
- **rea_items_url:** Almacena todas las direcciones web asociadas a los recursos educativos.
- **rea_items_y_archivos:** En el caso de que se asocien archivos al recurso educativo, esta tabla guarda el nombre del archivo y la dirección generada en el servidor, su tamaño, formato y fecha.
- **rea_historico_recursos:** Aquí se registran todas las acciones (crear, editar, borrar) que los usuarios efectúan sobre los recursos educativos.
- **rea_tipo_recurso:** Es una lista predefinida y tomada del antiguo Dspace para describir la naturaleza del REA. Esta tabla tiene registros predefinidos.
- **rea_area_utpl:** Contiene una lista de las áreas y las carreras de la universidad, sirve para establecer el ámbito de conocimiento en que se ubican el REA.
- **rea_idioma:** Contiene las categorías de idiomas disponibles para describir un REA.
- **rea_ranking:** Aquí se almacenan las votaciones del estudiante por cada recurso educativo disponible en el curso.

Algunas de las principales tablas que se han aumentado en el EVA se muestran en la siguiente figura:

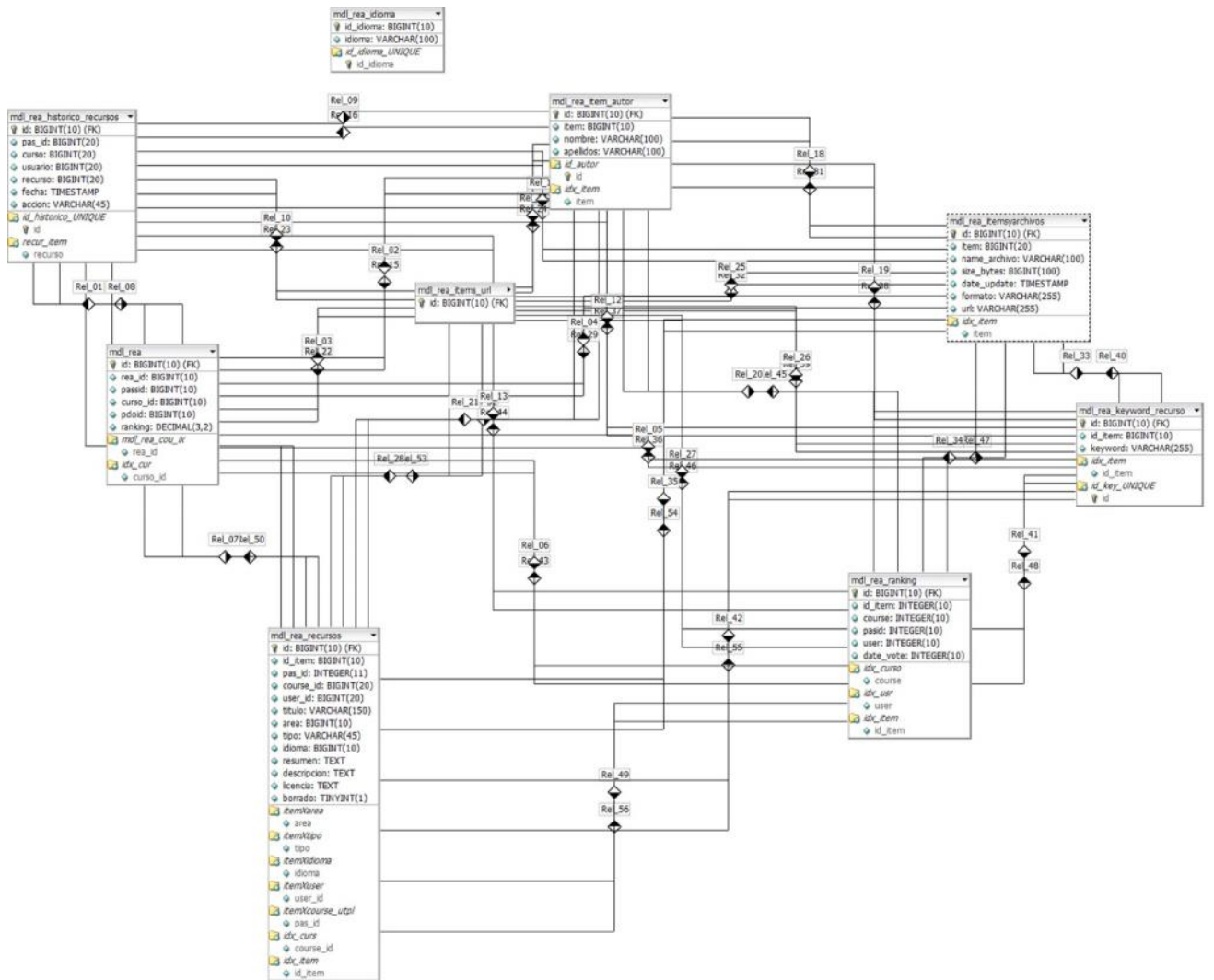


Gráfico 15 Modelo entidad - relación REA de la UTP (Suarez, 2011)

CAPITULO 3: DISEÑO Y DESARROLLO DEL AGENTE RECOMENDADOR EN EL EVA DE LA UTPL

3.1. Problemática

En la Universidad Técnica Particular de Loja, se cuenta con un entorno virtual de aprendizaje denominado EVA, en el cual interactúan todos los estudiantes ya que ha sido implementado para llevar a cabo los cursos de las carreras en los diferentes periodos.

Cada periodo el estudiante ingresa al entorno a desarrollar las actividades de aprendizaje para su formación. En la modalidad presencial es reciente su implementación, pero en la modalidad abierta es sumamente importante el uso del entorno por las distancias de los centros universitarios, brindando la facilidad al estudiante de enviar sus tareas por medio de internet.

Para desarrollar entornos educativos se utilizan los conocidos sistemas de gestión de aprendizaje o LMS. Existe una gran variedad de estos sistemas, pero su fin es el mismo crear cursos en línea para el desarrollo del aprendizaje, es una idea muy satisfactoria para el estudiante llevar sus cursos en línea debido a la comodidad que esto representa. Pero el uso de los sistemas de gestión de aprendizaje conlleva algunos problemas como es la adaptabilidad a las necesidades del usuario, como se ha revisado en la literatura de estos sistemas, solo permiten modificaciones de características no muy relevantes al estudiante como la visualización del entorno, en si no se adaptan a lo que el estudiante necesita, otro problema es la sobrecarga de información, debido al avance de las tecnologías de internet la información ha crecido de forma exponencial, esto afecta también a los entornos educativos los cuales se ven llenos de información que no interesa al usuario o que no cumple con sus expectativas.

El sistema de gestión de aprendizaje que viene utilizando la universidad es el Moodle, desde varios años atrás ha sido la plataforma para desarrollar los cursos. Este sistema no es ajeno a los problemas antes descritos de adaptación y de sobrecarga de información.

En el caso del primer problema el estudiante tiene que entrar a un entorno ideal donde se le puedan brindar las ayudas posibles en base a su perfil, es por esta razón que se busca adaptar características al entorno que cumplan con este objetivo.

Para el segundo problema que es el de sobrecarga de información, el objetivo es ayudar al estudiante en la toma de decisiones filtrando información que sea de utilidad al mismo.

El enfoque del proyecto son las actividades del estudiante en el entorno, dar soporte a estas actividades adaptando un agente el cual recomiende recursos relacionados al desarrollo de esas actividades, ya se tiene un módulo REA en el entorno en el cual el profesor sube recursos, pero a veces estos recursos no están relacionados con las actividades que el estudiante tiene que

realizar, haciendo que el estudiante busque información en motores de búsqueda tradicionales como google que también le mostrara una infinidad de recursos idóneos y no idóneos con sus tareas.

Por esta problemática, surge la necesidad del proyecto de un agente recomendador como soporte a las actividades de aprendizaje del estudiante en el EVA.

3.2. Diseño de la solución

El diseño de la solución viene dado por la estructura del agente recomendador, en que entorno va a trabajar y cuál es su objetivo principal.

El entorno en donde va a trabajar el agente es el EVA, específicamente en el sistema de gestión de aprendizaje Moodle, por lo que en una primera etapa se ha realizado la identificación de las tareas pendientes del estudiante, el cual se describe en el capítulo 2 de esta investigación.

Para el objetivo del agente recomendador el cual es dar soporte a las actividades del estudiante en el EVA, en una segunda etapa se ha requerido realizar un componente semántico del agente, el cual se encarga de realizar anotaciones semánticas de las tareas pendientes del estudiante y poder inferir sobre estos recursos, los cuales serán recomendados al estudiante. Ha sido preciso buscar herramientas que ayuden al desarrollo de la parte semántica del agente, entre las varias que hay dentro de la web semántica se ha decidido trabajar con GATE por lo popular que es y por la entidad que maneja esta herramienta la cual es el instituto de NLP (Natural Language Process) de reino unido. Esta herramienta se encarga de realizar las anotaciones semánticas, pero se necesita de más técnicas de ingeniería del conocimiento como son la indexación, extracción de información, entre otras. Por lo que se ha analizado la plataforma KIM el cual es una plataforma con tecnologías para la gestión del conocimiento muy completa entre sus herramientas están las anotaciones semánticas GATE, conjunto de ontologías y base de conocimiento KIMO, métodos de indexación, y otras que se describen en el capítulo 1.

3.3. Objetivos

Objetivo General

- Crear un agente recomendador que ofrezca soporte en las actividades de aprendizaje en el EVA de la UTPL

Objetivos Específicos

- Diseñar un esquema del agente recomendador
- Identificar actividades pendientes en el Entorno Virtual de Aprendizaje.
- Diseñar la arquitectura del agente en base a su esquema
- Construir el agente según su arquitectura
- Dar soporte a las actividades pendientes con la recomendación de recursos educativos.

3.4. Esquema del Agente

Según el libro de Inteligencia Artificial de (Russell & Norving , 2004), un agente tiene los siguientes componentes los cuales han sido adecuados al proyecto de investigación

Sensores: es la parte del agente en donde recibe la información introducida por teclado en el medio, lo que el agente acoge de los cursos que se llevan en el EVA, y esta información son las tareas pendientes.

Actuadores: los actuadores del agente es el resultado que se muestra al estudiante después de procesar información, los resultados esperados son los recursos adecuados a las tareas pendientes que se muestran en pantalla.

Entorno: el entorno donde se desenvuelve el agente es el entorno virtual de aprendizaje.

La función que tiene el agente recomendador es ofertar recomendaciones con respecto a las tareas pendientes del estudiante, para esto tiene que utilizar la técnica de recomendación basada en ítem y el subtipo basado en conocimiento, en este proceso utiliza a Kim para realizar las anotaciones y las búsqueda de los recursos adecuados.

Según las características el modelo del agente recomendador utilizado para este proyecto es el siguiente:

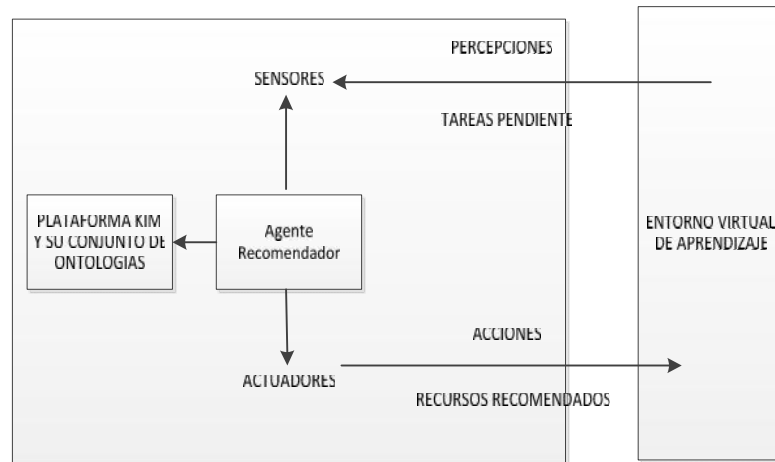


Gráfico 16 Esquema Agente Recomendador UTP

Con este primer esquema que puede guiar la construcción de una arquitectura completa del agente empezamos a definir cada una de los componentes que tendrá el agente y que serán construidas para los objetivos propuestos. Como se puede observar en el esquema el agente recibe información de las actividades desde el EVA, luego valiéndose de la plataforma KIM para gestión del conocimiento realiza las anotaciones semánticas y la búsqueda del recurso adecuado a esas actividades. El proceso de identificación de actividades pendientes se ha desarrollado en el capítulo 2 de esta tesis.

Para las recomendaciones el agente utiliza algunos elementos del modelo descrito en (Santos & Boticario, 2008), ya que en este trabajo se lleva a cabo en un entorno educativo, así que se puede utilizar para definir cómo se va a llevar las recomendaciones en el agente

3.5. Elementos parte de Recomendación del Agente

3.5.1. Categoría

Este elemento busca clasificar el alcance de la recomendación en base al contexto en el cual se trabaja, debido a que no se encuentra un valor que encaje en el proyecto se ha creado una categoría que es la siguiente:

- Soporte de actividades: las recomendaciones en este proyecto tienen el alcance de dar soporte al estudiante en sus actividades en el EVA

3.5.2. La técnica

En esta parte se ha tenido que investigar que demandas tiene un entorno educativo con respecto a un sistema recomendador es así que en (Drachsler, Hummel, & Koper, 2007) se tiene que “La motivación para cualquier sistema recomendador es asegurar el uso eficiente de recursos disponibles en una red”, en el caso del agente recomendador a utilizar dentro del entorno universitario su fin es que el estudiante utiliza de mejor manera los recursos y además estos sirvan para sus actividades dentro del mismo.

Además se tiene que el contexto individual de cada estudiante y el dominio en un entorno educativo tienen bastante influencia en el sistema recomendador.

Otro requerimiento específico dentro de un entorno educativo es el que el sistema recomendador de soporte al estudiante en su proceso de aprendizaje, por eso el sistema recomendador tiene que tomar en cuenta contenido educacional para generar sus instrucciones o recomendaciones.

Muchas de las veces resulta conveniente combinar varias técnicas de recomendación con el fin de minimizar los problemas que contiene cada una de estas técnicas y que se puedan generar recomendaciones de calidad según lo requerimientos se puede formar una estrategia de recomendación que internamente conoce cuales técnicas favorecen a un determinado tipo de condición.

Tomando todo esto en consideración y el estado del arte de los sistemas recomendadores, se ha decidido aplicar la técnica basada en Memoria de filtrado basado en el contenido de atributos y como subtipo de esta técnica, la de filtrado basado en conocimiento.

3.5.3. Origen

Este elemento tiene relación a la fuente que origina la recomendación en el caso del agente este valor es el siguiente:

- Necesidad del estudiante: lo que origina la recomendación es la necesidad del estudiante de algún apoyo para realizar sus actividades

3.5.4. Interfaz de Usuario

En el caso del agente recomendador para la UTPL se tiene una interfaz dentro del entorno virtual de aprendizaje donde en primer lugar se muestran todas las actividades pendientes del

estudiante y al deslizarse por las mismas se le muestran las recomendaciones de los recursos que puede utilizar, en la parte del desarrollo de agente se describirá un poco más detallado esta interfaz.

3.5.5. Condiciones Aplicables

Según este elemento se deben dar algunas condiciones idóneas para poder ofertar la recomendación en el caso del agente recomendador estas condiciones vienen al interactuar dentro de la plataforma por lo que se ha tomado la siguiente condición:

- Plataforma: que el estudiante ingrese a la plataforma y revise los cursos a los cuales esta matriculado, el agente mostrara las tareas pendientes y las recomendaciones que tiene para las mismas.

3.6. Arquitectura del Agente Recomendador de la UTPL

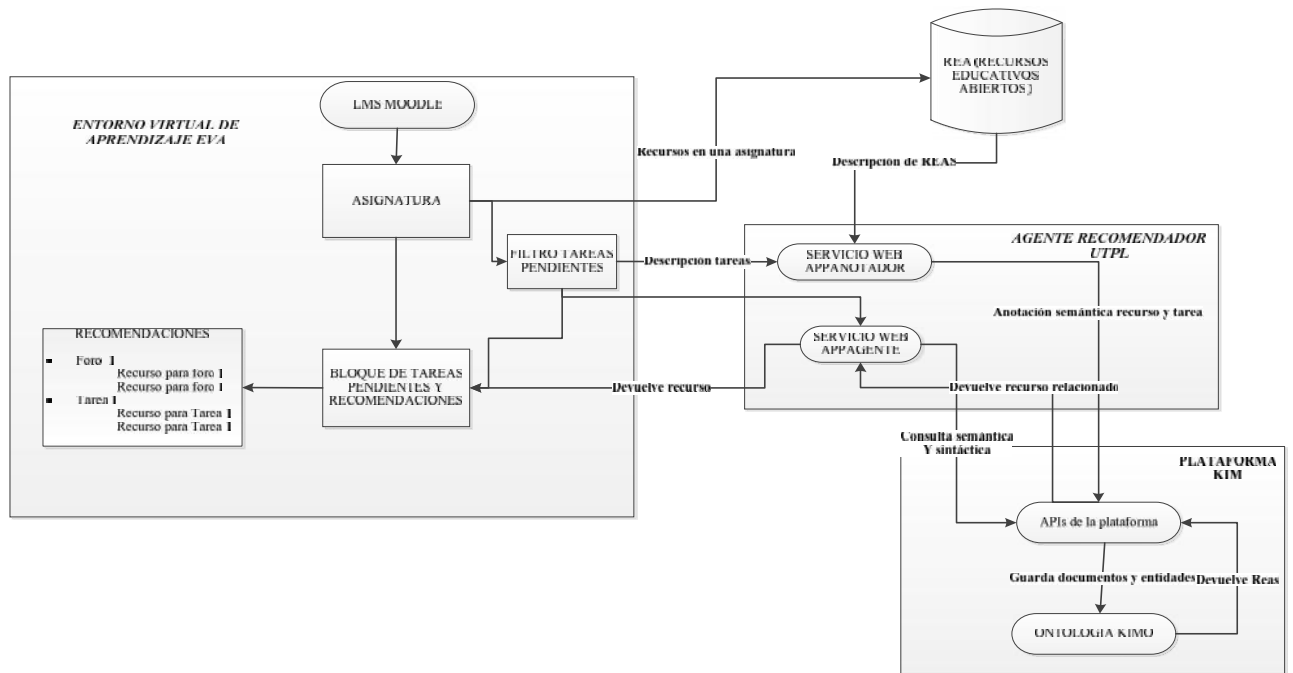


Gráfico 17 Arquitectura Agente Recomendador UTPL

La arquitectura del agente recomendador UTPL se ha modificado varias veces en el transcurso del desarrollo de esta tesis, resultando la mostrada en el gráfico 7.

Se puede observar que el agente esta interpuesto entre la plataforma Kim, mediante la cual se realiza las recomendaciones de los recursos y el EVA de la UTPL, en el cual se muestran las recomendaciones.

En una primera fase se ha desarrollado un filtro de actividades, para esto se ha identificado primero las actividades pendientes del estudiante en el EVA descrita en el capítulo anterior, también se puede observar el repositorio de REAs del EVA del cual se van a recuperar los recursos a recomendar.

El agente recomendador consta de dos servicios web, el uno se encarga de recibir las tareas pendientes y los recursos de los cursos para anotarlos semánticamente en los repositorios de conocimiento de kim es decir su ontología a la cual se le pueden agregar nuevas anotaciones y documentos. El otro servicio web tiene la funcionalidad de realizar una búsqueda semántica y luego una sintáctica de los recursos relacionados a las anotaciones de las actividades pendientes y recuperarlos. Todo esto se logra con sus interfaces de programación de aplicaciones APIs por sus siglas en ingles que posee kim.

Al final se muestra en un bloque en el EVA las actividades pendientes y los recursos recomendados para realizar estas actividades.

3.7. Desarrollo del Agente recomendador y sus componentes

Los componentes que forman parte de la arquitectura se describen a continuación;

3.7.1. Bloque de actividades Pendientes

El bloque de actividades pendientes ha sido desarrollado en el capítulo anterior pero forma parte del agente recomendador y ha sido añadido al bloque de recomendación ya que se realizó un solo bloque de recomendación en donde se muestra las tareas pendientes y las recomendaciones todo por tener una interfaz sencilla en el EVA, la codificación de este bloque se encuentra en el anexo 6 de la presente investigación.

3.7.2. APIs de la plataforma Kim y servicios web del agente

En esta parte del desarrollo del agente se ha pensado en la técnica de recomendación basada en conocimiento que es una subclase de la basada en atributos y define la parte semántica del agente.

Esto se ha realizado utilizando la plataforma Kim para realizar las anotaciones semánticas y la recuperación de los recursos.

Para realizar todas estas funciones del agente se debe de utilizar la diferentes APIS de la plataforma KIM, ya que de esta plataforma se usa los servicios para las anotaciones semánticas, anotación de entidades nombradas y las consultas. Se anota tanto las actividades pendientes como los recursos educativos abiertos del entorno y se hace una recuperación de los recursos que tienen relación con las actividades pendientes.

En general estas son las APIS que se pueden utilizar en servicios web con la plataforma KIM:

- SemanticAnnotationAPI service
- DocumentRepositoryAPI service
- SemanticRepositoryAPI service
- EntityAPI service
- QueryAPI service

Cada uno de estos servicios tiene diferentes métodos los cuales pueden ser invocados desde cualquier servicio web, Kim utiliza RMI (Remote method invocations), para tener acceso a sus diferentes servicios. Para el desarrollo de esta fase del agente se tiene que invocar algunos de estos servicios o consumirlos mediante los servicios web que se desarrollaron como parte de este proyecto.

SemanticAnnotationAPI service

Esta API es muy importante y está integrada al agente recomendador ya que sirve para invocar procesos de anotación semántica. En una rutina estándar de población, la anotación debe ser llamada después de que el cuerpo del documento y las características son guardados en un documento tipo KIM. Se distinguen dos métodos principales:

Execute: este recibe como entrada un documento tipo KIM, devuelve como resultado un documento con el mismo tipo pero con las nuevas anotaciones semánticas, adicionalmente se guarda todos estos componentes en un repositorio semántico (nuevas instancias y relaciones)

ExecuteText: recibe como entrada un texto plano y da como resultado un arreglo de tipo kim el cual contiene las anotaciones semánticas creadas, adicionalmente se guardan todos estos componentes en un repositorio semántico (nuevas instancias y relaciones)

DocumentRepositoryAPI service

Se encarga de guardar todos los documentos anotados para ser recuperados en lo posterior con métodos como loadDocument o getDocument. Tiene los siguientes métodos:

addDocument: permite agregar o guardar documentos a un repositorio, antes de esto los documentos tienen que ser anotados, ya que no hay manera de modificarlos cuando están guardados.

getDocuments: permite el acceso al contenido estructurado de todos los documentos anotados y guardados, mediante el uso de consultas estructuradas que dan como resultado respuestas estructuradas

getDocumentsIds: este método permite obtener el identificador único de un documento guardado en un repositorio, con este identificador podemos tener acceso al contenido del documento mediante métodos como loadDocuments.

loadDocument: tiene la capacidad de acceder al contenido estructurado de un documento tipo kim mediante su identificador único o id

getDocumentFeatures: permite acceder a las características de un documento como son título, autor, descripción, entre otras. Obtiene como entrada un id y da como resultado un arreglo de características.

getEntities: permite el acceso a las entidades encontradas en un conjunto de documentos, se puede obtener todas estas entidades o aquellas que satisfacen la consulta de entrada de este método.

SemanticRepositoryAPIservice

Se encarga de grabar la información semántica extraída como RDF. El servicio también permite navegar y modificar el repositorio. Tiene los siguientes métodos:

evaluateSelectSeRQL: tiene como entrada una consulta en SeRQL en texto plano y da como resultado una tabla en una estructura de datos SemanticQueryResult

importData: permite importar RDFs directamente al servidor de KIM toma en cuenta 4 parametros: datos RDF como string, un URI base para las instancias creadas, constante que especifica el formato RDF y una bandera que por lo general es falsa.

EntityAPIservice

Provee de una abstracción sencilla en lo alto de la SemanticRepositoryAPI. Permite la carga de toda la información semántica de una entidad dada con una sola llamada.

getEntityDescription: permite el acceso a la descripción o contenido estructurado de una entidad. Tiene como entrada el URI de la entidad y da como resultado la descripción mediante el objeto EntityDescription.

addEntityDescription: permite agregar una nueva entidad en el repositorio semántico, recibe como entrada un objeto Entitydescription y no da resultados solo agrega la entidad.

serializeEntities: permite la transformación (serializacion) de las descripciones de entidades estructuradas en cadenas de texto

QueryAPIservice

Mediante este servicio se pueden formalizar consultas complejas tanto al repositorio semántico como al repositorio de documentos. Combina funcionalidades de consulta del DocumentRepositoryAPI y del SemanticRepositoryAPI , se pueden realizar consultas semánticas y sintácticas. Las consultas deseadas son entradas a traves de objetos tipo SemanticQuery y da como resultado una compleja estructura de objeto tipo SemanticQueryResult, estos son sus métodos:

getEntities: permite la recuperación de entidades relacionadas a una consulta semántica de tipo SemanticQuery y nos da como resultado un objeto de tipo SemanticQueryResult

getDocuments: permite la recuperación de resultados semánticos como un conjunto de filas de entidades. A cada fila de entidades corresponde un conjunto de documentos en donde se especifican dichas entidades. Las restricciones en la consulta se dan en el objeto SemanticQuery. Este servicio da como resultado un objeto SemanticQueryResult, para este método ambos el resultado de entidades y la parte del conjunto de documentos están poblados en los repositorios de KIM

getDocumentIds: es una versión reducida del método anterior con la diferencia que solo se obtienen los identificadores únicos de los documentos para luego con estos poder obtener su contenido.

Luego que se ha descrito cada uno de los APIS de Kim y para que pueden ser consumidos se ha desarrollado los servicios web que van a utilizarlos para integrarlos al agente recomendador.

3.7.3. Servicios web del agente y sus métodos principales.

La parte de recomendación semántica viene dada por la utilización de los servicios de la plataforma KIM, para ello se han creado dos servicios web encargados de sobrellevar esta fase del proyecto, se ha realizado dentro de la herramienta Net Beans, además se ha instalado la plataforma KIM para poder utilizar todos sus servicios, la instalación de kim y sus configuraciones se muestran en el anexo 4.

Los siguientes son los servicios web:

- appAgenteWS
- WSanotador

Empezaremos por describir el servicio web WSrecomendador ya que este posibilita anotar semánticamente tanto los recursos educativos abiertos como las actividades pendientes de los estudiantes.

3.7.4. Agente Anotador WSanotador:

A continuación se describe en la tabla los métodos principales de este servicio que en la arquitectura es utilizado como un pequeño agente que se encarga de realizar las anotaciones semántica de las tareas y de los recursos del bloque REA.

Tabla 7 Tabla descriptiva del Servicio Anotador

Servicio Web Anotador		
Datos de entrada	Método	Salida
server (información del servidor)	Conectar	rmi Objeto tipo kim
datos (información de los recursos)	Procesar Reas	msg, este método envía los recursos al método anota
rmi, ítem, contenido y modo información de las tareas y recursos	Anota	anotado, este método se encarga de utilizar las funciones de kim para anotar y guardar documentos.
Tarea	Recomendar	Este método se encarga de enviar las tareas al método anota_tarea del kim

Este servicio web es creado para trabajar con las actividades pendientes y los recursos educativos abiertos del entorno virtual, tiene la funcionalidad de anotar semánticamente las actividades y los recursos, para posteriormente utilizar esta información y realizar la correspondencia para recomendar.

Tiene dos variables globales las cuales son:

- rmi: variable que es de tipo Kim, es decir se la va a utilizar para acceder a los varios servicios de Kim a través de su conjunto de métodos.
- lecto: es una variable para leer los datos que se van generando en la aplicación y reportar errores.

A continuación se describe que métodos utiliza este servicio web y que métodos de la plataforma Kim integra al proyecto.

Conectar: es el primer paso para el web service ya que debe conectarse a kim para poder utilizar todos sus servicios tanto de anotaciones, guardar estas anotaciones y utilizarlas cuando se las requiera. Como se observa en la tabla este método utiliza información del servidor que se quiere conectar a kim en este caso el servidor es localhost.

Procesar Reas: este proceso se encarga de procesar las Reas (recursos educativos abiertos), en este paso se obtiene información de los recursos de la universidad para su posterior anotado semántico mediante el método anota.

Anota: este método se encarga de anotar los recursos y es utilizado por el método anterior.

Recomendar: este procedimiento también se encarga de hacer las anotaciones semánticas pero no de los recursos sino de las tareas, es decir busca en la descripción de las tareas entidades importantes y las anota.

3.7.5. appAgenteWS _recomendador

Tabla 8 Tabla descriptiva del Servicio web Agente_recomendador

Servicio Web Agente_recomendador		
Datos de entrada	Método	Salida
server (información del servidor)	Conectar	rmi Objeto tipo kim
actividad (información de las actividades en el EVA)	Recomendar	resp, en este método se realiza la búsqueda semántica de documentos relacionados con las entidades anotadas de las tareas y las recupera en un arreglo de datos. Para transformar los objetos de un tipo a otro en php se utiliza json Si no se obtiene resultado se envía a consulta sintáctica
anotations	Consulta Sintáctica	resp, en este método se realiza una consulta sintáctica y se devuelve un arreglo con los documentos recuperados

Este servicio es el encargado de realizar las consultas de los documentos y entidades guardadas en los repositorios de KIM, además devuelve los documentos que tienen relación con las entidades mencionadas en las tareas pendientes del estudiante, tiene tres variables globales:

- Kim: variable de tipo KIM que sirve para acceder a los APIS de la plataforma
- Data: variable de tipo Gson, nos permite transformar objetos en cadenas de texto y viceversa
- resp: variable de tipo string que es donde se reciben las respuestas.

Este servicio tiene los siguientes procedimientos.

Conectar: es igual que el otro servicio web ya que debe conectarse para utilizar los servicios de la plataforma Kim

Recomendar: en este procedimiento se utiliza las anotaciones, para realizar una búsqueda en primera instancia semántica para luego hacer una búsqueda sintáctica entre las entidades anotadas y los documentos, para luego dar como respuesta los documentos recuperados en donde se especifican algunas de las entidades de la consulta.

Consulta sintáctica: se encarga de hacer una consulta sintáctica si es que no se realizó la semántica

En este método se utiliza el QueryAPIservice y SemanticRepositoryAPIservice

En el anexo 6 se encuentra la documentación de los códigos de ambos servicios.

3.8. Integración del agente recomendador al EVA

Para realizar la integración del agente al entorno se ha tenido que instalar nusoap , ya que al trabajar con servicios web hay que establecer un protocolo para comunicarse entre aplicativos en este caso se utiliza este protocolo para comunicar el EVA con los servicios web del agente.

Los detalles de la instalación de nusoap se encuentran en el anexo 6.

Adicionalmente se ha creado una nueva tabla en el entorno la cual se encarga de guardar todos los REAS anotados por el agente, esto es con el fin de que no se repitan las recomendaciones en el entorno.

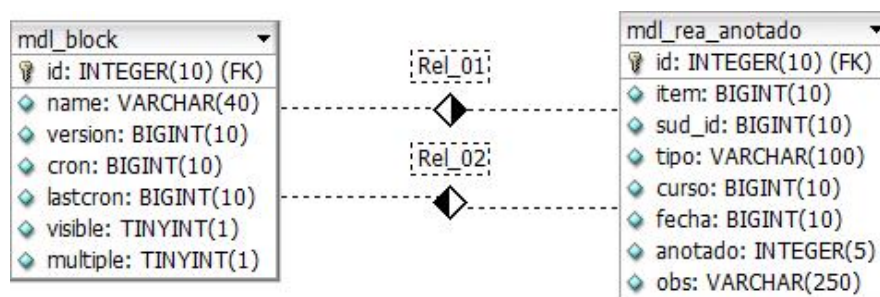


Gráfico 18 tabla de recursos anotados

Como se muestra en la imagen esta tabla tiene relación con el bloque en donde se realizan las recomendaciones.

Una vez instalado el nusoap, se tiene en la aplicación los siguientes elementos que hay que integrar

- Cliente: se ha creado un cliente en el bloque de recomendaciones el cual consume los servicios web creados anteriormente
- Bloque de recomendación: bloque el cual ha sido modificado añadiendo los recursos recomendados en las tareas pendientes que en un inicio se realizó.
- Servicios web

Una vez desarrollados todos estos elementos el cliente que se encuentra en la carpeta del bloque recomendador denominado "process.php" se comunica con los servicios web a través de las siguientes líneas de código:

```
$client = new SoapClient("http://localhost:8084/appanotador/WSanotador?wsdl");  
$client = new SoapClient("http://localhost:8084/app_agente/appAgenteWS?wsdl");
```

3.8.1. Problemas durante la integración

En la integración del agente al EVA han surgido los siguientes problemas:

- Para la ejecución de un script en php se da un tiempo límite de 5 minutos, lo cual dificultó realizar la búsqueda semántica, ya que se cortaba la comunicación y no daba resultado.
- El gasto computacional de herramientas semánticas han dificultado pruebas en computadoras con bajas prestaciones.
- Las recomendaciones se repetían cuando el estudiante ingresaba varias veces a la plataforma.

Para solucionar estos problemas se ha realizado lo siguiente:

- Se ha modificado la codificación y aumentado una línea donde se da un tiempo ilimitado hasta que haya una respuesta del servidor php
- Se ha realizado el proyecto en una computadora con mayores prestaciones.
- Se ha creado una tabla donde se verifica que si el documento ya está anotado no lo vuelva a recomendar.

Al final de la integración del agente en el EVA tenemos la siguiente captura donde se muestra su interfaz y se describe cada una de las partes de la misma:

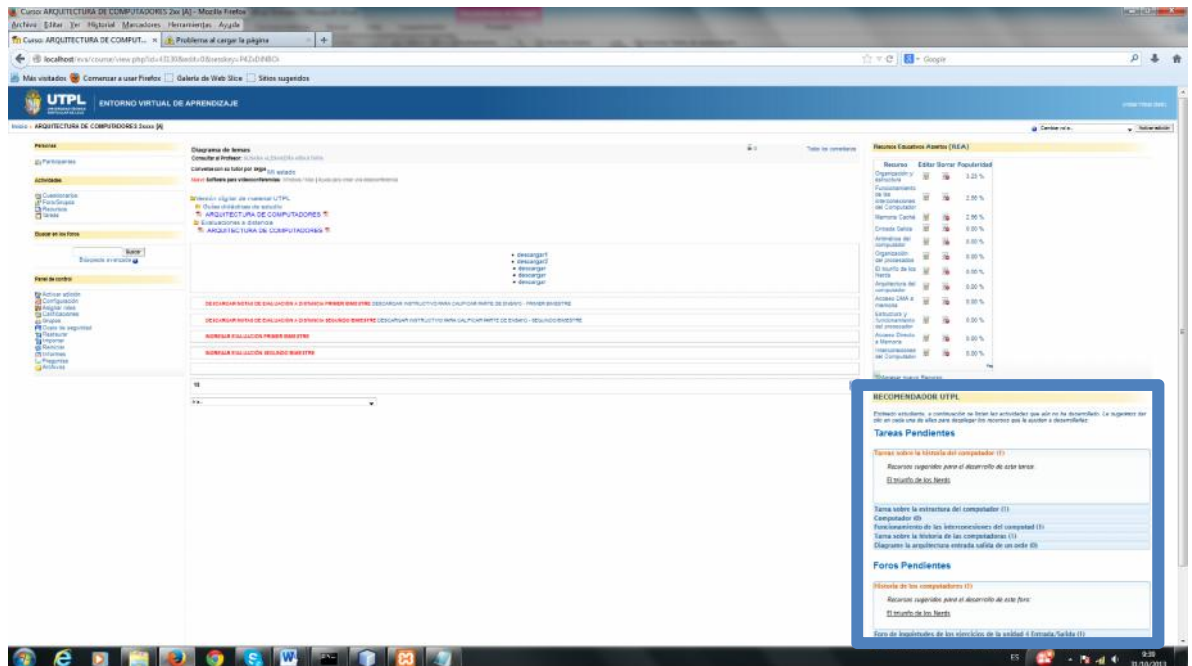


Gráfico 19 Captura del agente recomendador integrado al EVA

En la gráfica se observa el bloque “Recomendador” tiene las Tareas y Foros pendientes, al realizar clic en las actividades despliega una lista de recursos relacionados a la tarea.

En el gráfico se describe las actividades pendientes que tiene un estudiante en la materia de “Arquitectura de Computadores”, con un mensaje introductorio en donde se sugiere al estudiante realizar un clic sobre cada actividad tanto tareas como foros con la finalidad que se le desplieguen los recursos recuperados para cada actividad. Como se puede observar en la imagen hay una tarea y un foro desplegados que poseen un recurso respectivamente para su desarrollo.

3.9. Pruebas con el agente recomendador

3.9.1. Pruebas de Precisión

Para realizar las pruebas, se han elegido tres materias del entorno de pruebas de la UTPL creado localmente:

- Arquitectura de Computadores
- Planeación Estratégica
- Fundamentos de Informática

Cada uno de estos cursos ha sido configurado para que contengan la cantidad de 10 actividades y 15 recursos abiertos, para ver el funcionamiento del agente.

Las pruebas a realizarse en el agente, ya que basa su estructura en la recuperación de información una de las métricas más extendidas en este tipo de sistemas son las pruebas de precisión según (Tolosa, 2008).

Estas pruebas se llaman de Precisión y su evaluación es en base a los documentos relevantes recuperados por cada actividad en la cual se recomienda el documento.

La ecuación de esta prueba es la siguiente:

$$\text{Precisión} = \frac{\text{Documentos Relevantes Recuperados}}{\text{Documentos Recuperados}}$$

3.9.2. Pruebas de Usabilidad

En el libro (Pressman, 2007), se presentan diferentes métodos de evaluación de software, uno muy interesante son las pruebas de usuario, en las cuales el desarrollador permite a los posibles usuarios finales utilizar el aplicativo, luego se realizan preguntas referentes a la usabilidad del sistema y en base a sus respuestas se puede evaluar que tan usable es el agente recomendador.

Se han tomado en cuenta que este agente lo van a utilizar los estudiantes y se ha realizado las pruebas con dos estudiantes y un docente de la UTPL.

Las siguientes son las preguntas que se han realizado:

Preguntas de usabilidad

1. El módulo de recomendación es intuitivo
2. Se entiende a primera vista el objetivo de este modulo
3. Se siente cómodo utilizando el módulo de recomendación
4. El modulo es fácil de utilizar
5. Los recursos recomendados son relevantes
6. El recurso está relacionado con la actividad
7. Cree que los recursos sugeridos ayudan a resolver las actividades pendientes.

Preguntas de navegabilidad

1. Al momento de navegar a través de los ítems sugeridos en el módulo de recomendación presentó errores de navegación:

Si () No ()

Si su respuesta es Si, en que opciones se dieron los errores

2. La navegación por los recursos recomendados es:

Rápida ()

Lenta ()

Adecuada ()

Otra ()

3.10. Resultados de las pruebas

3.10.1. Resultados de la prueba de Precisión

Para el resultado de las pruebas de precisión se ha verificado cuantos documentos ha recuperado la aplicación y se ha realizado un cuadro con estos resultados:

Tabla 9 resultados de las pruebas de precisión

Curso	Actividades	Documentos Relevantes Recuperados	Documentos Recuperados
Arquitectura de Computadores	10	7	8
Planeación estrategica	10	3	5
Fundamentos de Informatica	10	14	16

Según la ecuación, se muestra el siguiente cuadro con la precisión en cada uno de los cursos que se han realizado las pruebas:

Tabla 10 Precisión en cada curso de prueba

Curso	Precisión
Arquitectura de Computadores	88%
Planeación estrategica	60%
Fundamentos de Informatica	87%
Promedio Total	78,6%

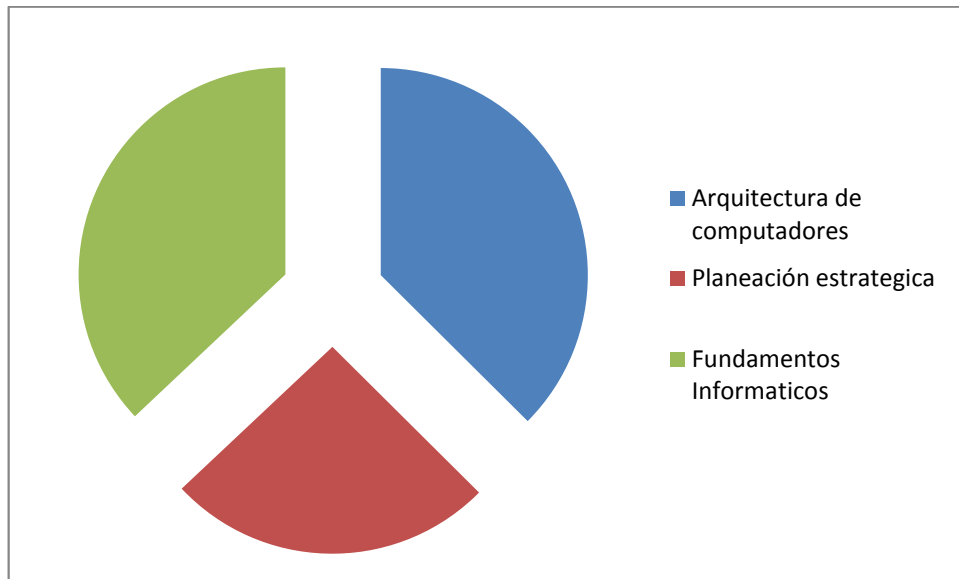


Gráfico 20 Porcentajes de la precisión en cada curso de prueba

Según estas pruebas el agente recomendador tiene una precisión general del 78,6%, que según el documento (Tolosa, 2008), se encuentra en un porcentaje aceptable de recuperación de documentos relevantes al usuario.

Cuando de todos los documentos recuperados por el agente se tiene que todos son relevantes a la actividad, se dice que el agente es completamente preciso, pero esto no se puede lograr debido a problemas de falsos positivos y ruido documental, que es cuando se recupera documentos que parecieran importantes pero llegado a la realidad no lo son, también se da el silencio informativo donde un documento que es relevante no se lo recomienda.

3.10.2. Resultados de la pruebas de usabilidad

De la encuesta realizada a un grupo de estudiantes y a un docente de la universidad que en total conforman una muestra de 10 personas se han obtenido los siguientes resultados:

Acerca de que si el bloque de recomendación es intuitivo, 8 usuarios consideran que es muy intuitivo y 2 consideran que es poco intuitivo.

De los 10 usuarios que utilizaron el bloque, 5 consideran que se entiende el objetivo del mismo a primera vista y 5 consideran que se entiende poco.

De los 10 usuarios que interactuaron con el bloque, todos consideran que se sintieron cómodos al utilizarlo.

De los 10 usuarios que utilizaron el bloque, todos consideran que es muy fácil de utilizar.

De los 10 usuarios que utilizaron el bloque, 7 consideran que los recursos son muy relevantes y 3 consideran que son poco relevantes.

De la muestra, 8 usuarios consideran que los recursos están muy relacionados con la actividad, mientras 2 consideran que es poca esta relación.

De la muestra, 8 consideran que los recursos sugeridos ayudan mucho a resolver sus actividades pendientes, mientras que 2 consideran que les ayuda poco.

Los resultados de las pruebas de usabilidad han ayudado a mejorar la forma de presentación del bloque de recomendación y del funcionamiento del agente recomendador, haciendo que la interfaz sea más amigable al usuario, debido a que al realizar las pruebas el grupo de estudiantes sugirió algunos cambios.

Un cambio ha sido en los documentos recuperados, en la materia de “Fundamentos Informáticos” se recomendaba algunos recursos que poco tenían que ver con la actividad, esto se solucionó mejorando las descripciones de cada actividad.

Otra sugerencia fue que se mejore la interfaz gráfica, debido a que en una primera impresión no lograron captar la funcionalidad del agente por lo que se realizó un trabajo de carpintería a la aplicación en el cual se especifica de mejor manera el bloque de recomendación

Esta prueba ayudó a mejorar la aplicación y según las preguntas realizadas el agente tiene un nivel de aceptación del 75,25%, lo que se debe a la falta de información acerca del agente y lo que realiza, por ultimo un estudiante hizo la sugerencia de realizar un folleto informativo para el usuario en el que se detalle las características del agente



Según los resultados el nivel de aceptación que tienen los usuarios es del 75,25%

Conclusiones

- Mediante los resultados se ha podido comprobar que el agente puede dar soporte al estudiante en sus actividades en el EVA, con la recuperación de un 78,6% de recurso ideales para realizar sus tareas, que en conclusión es un rango aceptable para sistemas de recuperación de información y puede sentar un precedente hacia la construcción de agentes en sistemas de gestión de aprendizaje
- Se concluye que existen diferentes técnicas de recomendación, por lo que en algunos casos hay que utilizarlas en conjunto o extender algún subtipo de las mismas para minimizar los problemas que tienen individualmente.
- Debido a los diferentes problemas que presenta cada una de las técnicas de recomendación se puede utilizar herramientas de la web semántica para solucionar algunos de estos problemas como el “arranque en frío”, que se relaciona con la falta de información puntuada por el usuario, así como también problemas de nuevo usuario o nuevo ítem.
- El agente recomendador obtuvo una precisión del 78,6% y un error del 21,4% esto se debe a los problemas que existen al recuperar información como falsos positivos, ruido documental y silencio informativo. Los problemas de falsos positivos y ruido documental se relacionan a cuando se obtiene un documento que pareciera relevante pero en la realidad no lo es y el problema del silencio informativo es cuando existen documentos relevantes que el agente recomendador no los recomienda.
- Se concluye que un requerimiento indispensable para el funcionamiento del agente es que el entorno le brinde información importante ya que así podrá realizar las inferencias necesarias y poder recuperar los documentos.
- Las tecnologías web semántica como los anotadores semánticos deben ser utilizados por agentes para explotar sus funcionalidades, de otra forma su uso se ve muy limitado.
- Los sistemas recomendadores se basan en algoritmos predictivos que buscan satisfacer necesidades de los usuarios para la toma de decisiones y solucionar la sobrecarga de información, en conclusión este proyecto es un paso inicial a solventar esos problemas en el entorno virtual de aprendizaje.
- Por el crecimiento de información en internet se han dado problemas de sobrecarga de información, por lo que se concluye que se necesita herramientas que ayuden con estos problemas, hay diferentes opciones y hay que apoyar de alguna manera con proyectos encaminados a sobrellevar los problemas que causa toda esa información en internet.
- Se puede concluir que los sistemas recomendadores y principalmente los basados en conocimiento representan una alternativa muy potente para empresas e instituciones educativas que deseen manejar su información de manera organizada aumentando así la interactividad con sus bases de información.
- En entornos donde se incluye una gran cantidad de información, se debe tener la capacidad de seleccionar la mejor y de una forma rápida, es por esto que se han enfocado proyectos de sistemas de recomendación que filtren esta información y el proceso se pueda dar de una forma rápida.

En el transcurso del desarrollo de este tema de tesis se ha podido concluir que las técnicas de recomendación más adecuadas a un entorno educativo son las basadas en memoria, a las cuales se les puede extender con la web semántica y minimizar sus falencias.

- Si bien tiene un funcionamiento aceptable en el entorno de la universidad, se puede mejorar con trabajos a futuro, que pueden comenzar tomando en consideración este proyecto o integrarlo con otro.

Recomendaciones

- Una recomendación es que a futuro se describan de mejor manera las actividades enviadas al estudiante en el EVA, para que de esta forma si se adapta nuevos agentes, estos tengan información con la cual trabajar y realizar las anotaciones semánticas.
- El agente recomendador se ha llevado en un entorno de pruebas, se recomienda implementarlo en el entorno real de la UTPL.
- Puede ser una buena opción para la UTPL adquirir la licencia de la herramienta KIM para explotar todas sus funcionalidades.
- Incentivar a los estudiantes en proyectos de la web semántica y sistemas recomendadores, debido a que son tecnologías que recién empiezan y tiene muchos campos donde aplicarla.
- Relacionar más proyectos con la plataforma KIM, ya que en este proyecto solo se han utilizado algunas de sus características, en lo posterior se puede utilizar otras como las de creación de ontologías.
- El presente trabajo se ha utilizado el repositorio de recursos REA del EVA, por lo que sería una buena idea trabajar con otros repositorios como los de OCW, los cuales son cursos abiertos en línea que puedan ayudar al desarrollo de las actividades en un futuro plan de mejoras al agente.
- El presente trabajo por problemas de licencia se ha limitado a utilizar la ontología del KIM, en lo posterior se pueden crear ontologías de cursos en específico y esto ayudaría a mejorar las recomendaciones.
- Si se piensa en trabajar con la plataforma kim, se tiene que utilizar computadores con características bastante aceptables por el gasto computacional que representan las diferentes herramientas semánticas como kim

Trabajos Futuros

Al realizar este proyecto de tesis se ha visto las diferentes capacidades que tienen los sistemas recomendadores al agregar herramientas de gestión del conocimiento.

Una de las restricciones del proyecto ha sido utilizar la ontología de Kim, que se basa en entidades nombradas como personas, locaciones, fechas importantes, valores, nombres de empresas, entre otras. Al tener la posibilidad de crear ontologías propias adecuadas a los conceptos manejados en un curso y no solamente a entidades nombradas, se puede mejorar en un gran nivel el bloque de recomendación, por ejemplo se puede estar en un curso de Biología y crear una ontología solo de este campo, al momento de enviar las tareas si en la descripción algún concepto está relacionado con la ontologías se puede recomendar mejores documentos con un alto nivel de relación semántica con la actividad.

Se puede construir otro tipo de agentes que interactúen con el realizado en esta investigación para realizar un monitoreo de estudiantes que no realizan sus actividades de aprendizaje y sacar estadísticas de que carreras presenta estudiantes con bajo nivel académico o baja predisposición a realizar sus actividades.

BIBLIOGRAFIA

- (2013). Obtenido de <http://www.ontotext.com/kim>
- Adomavicius, G. (2005). Toward the next generation of recommender system a survey of state of art and posible extensions .
- Álvarez, A. I. (2010). Magadi, una propuesta de sistema adaptativo muti-agente de apoyo al aprendizaje en un marco combinado. Donostia, España.
- Arana, A. I. (2010). Magadi, una propuesta de sistema adaptativo muti-agente de apoyo al aprendizaje en un marco combinado. Donostia, España.
- Burke, R. (2007). Hybrid Web Recommender System . Illinois , Chicago , USA .
- Carrillo, A. F., Llovet, J. M., & Rodríguez, I. (2010). Sistema Recomendador Orientado a la Educación Terapeutica del Paciente Diabetico. Madrid, España.
- Drachsler, H., Hummel, H. G., & Koper, R. (2007). Personal Recommender System for learners in lifelong learning networks: the requirements, techniques and model. The Netherlands.
- García, R. (2008). TFC: XML y web semantica Estudio del impacto de las aplicaciones comerciales basadas en tecnologias de la web semantica.
- Gerling, V. B. (2009). Un Sistema Inteligente para Asistir la Búsqueda Personalizada de Objetos de Aprendizaje.
- Guzmán, C. A. (2009). Sistema de Recomendación y Planificación Turistica de la Ciudad de Valencia. Valencia, España.
- Itmazi, J. A. (2005). Sistema Flexible de Gestion del e-learning para soportar el aprendizaje en las universidades tradicionales y abiertas. Granada.
- Llidó, D. M. (2002). Extracción y Recuperación de Información Temporal. Castellón.
- Moodle. (2013). *Moodle Docs* . Obtenido de http://docs.moodle.org/dev/images_dev/5/5a/Moodle2erd.png
- Moodle. (s.f.). *Bloque de Actividades*. Recuperado el 13 de 03 de 2013, de http://docs.moodle.org/24/en/Activities_block
- Moodle. (s.f.). *Moodle*. Recuperado el 13 de 03 de 2013, de Acerca de Moodle: http://docs.moodle.org/24/en/About_Moodle
- Nwana, H. S. (1990). Intelligent Tutoring Systems: an overview . Liverpool .
- Olga C Santos, J. G. (s.f.). User´s expirience with a recomender system in an open source standar-based learning management system . Madrid , España .
- Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D., & Goranov, M. (2010). KIM – Semantic Annotation Platform. Bulgaria.
- Pressman, R. (2007). Ingenieria de Software .
- ROMA Project. (2007). Personal Recomender System to Advice Learning Task in Moodle. Netherlands.
- Rusell , S., & Norving , P. (2004). *Inteligencia Artificial Un Enfoque Moderno* . Madrid : Pearson .
- Santos, O. C., & Boticario, J. G. (2008). User´s expirience with a recomender system in an open source standar-based learning management system. Madrid, España.
- Setten, M. V. (2005). Supporting People In Finding Information: Hybrid Recommender Systems and Goal-Based Structuring. Enschede, Netherlands.
- Sirma Group . (2013). Obtenido de <http://www.ontotext.com/kim>
- Suarez, S. (2011). Manual de Programación del bloque REA de la UTPL.
- Tolosa, G. (2008). Introducción a la recuperacion de información . Buenos Aires , Argentina .
- Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas, M., Motta, E., y otros. (2005). Semantic annotation for knowledge management: Requirements and a survey of the state of the art.
- Wang, S.-L., & Wu, C.-Y. (2001). Application of context-aware and personalized recommendation to implement an adaptive ubiquitous learning system.
- Wang, S.-L., & Wu, C.-Y. (2011). Application of context-aware and personalized recommendation to implement an adaptive ubiquitous learning system. Taiwan .

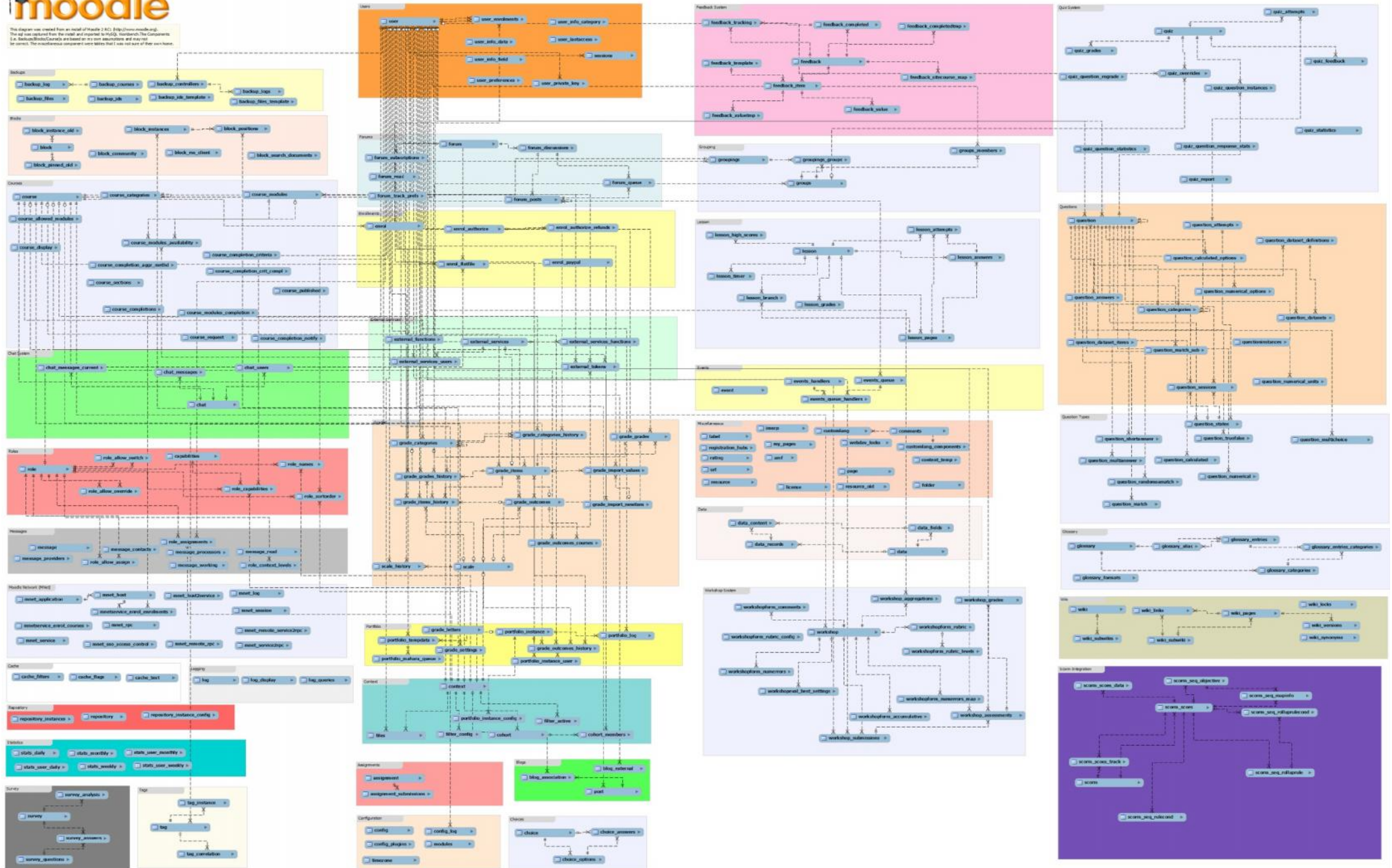
ANEXOS

ANEXO 1

Modelo entidad – relación moodle 2.2 (Moodle, Moodle Docs , 2013)



This diagram was created from an install of Moodle 2.2.1. It is for informational purposes only. The data was obtained from the moodle and exported to MySQL. The contents of the database are not guaranteed to be accurate and may change over time. The relationships between tables that are not shown here are not shown.



Anexo 2

Codificación e instalación del bloque de tareas pendientes

Con la identificación de las tareas pendientes, se ha realizado un bloque, el cual muestre las actividades pendientes del estudiante.

```
<?php
/*
Descripcion: Bloque que permite visualizar las tareas pendientes que tiene un estudiante.
Luego de obtener las tareas pendientes se podra recomendar sobre cada una de las tareas.
*/
class block_pendiente_tareas extends block_base {
    /*
    ** Descripcion: Permite definir como se inicializara el bloque.
    */
    function init() {
        $this -> title = get_string('blockname', 'block_pendiente_tareas');
        $this -> version = 2013011548;
    }
    /**
    ** Descripcion: Permite obtener el contenido que se visualizara del bloque
    **
    */
    function get_content() {
        global $USER, $CFG, $COURSE;
        require_once($CFG->dirroot . '/mod/forum/lib.php');
        require_once($CFG->dirroot . '/mod/assignment/lib.php');
        if ($this -> content !== NULL) {
            return $this -> content;
        }
        $course = $COURSE;
        $forums = get_records('forum', 'course', $course->id);
        $subscritos = "Foros: ";
        $tblPendientes = "<table border='1'><tr><th>Tipo</th><th>Nombre</th></tr>";
        $modinfo = get_fast_modinfo($course);

        foreach ($modinfo->instances['assignment'] as $cm) {
            //print_r($cm);
            if(!record_exists("assignment_submissions", "userid", $USER->id,
"assignment", $cm->instance)){
                $tblPendientes .= "<tr><td>Tarea</td><td>".$cm-
>name."</td></tr>";
            }
        }

        foreach ($forums as $forum) {
            if(!record_exists("forum_discussions", "userid", $USER->id, "forum",
$forum->id)){
                $tblPendientes .= "<tr><td>Foro</td><td>".$forum-
>name."</td></tr>";
                $subscritos .= ' '.$forum->name;
            }
        }
        $tblPendientes .= "</table>";
        $this->content = new stdClass;
        $this -> content -> text = $tblPendientes;
    }
}
```

```

$this->content->footer = "";
    return $this -> content;
}
}
?>

```

Instalación de Bloques en el EVA

Para llevar a cabo la instalación de este bloque se sigue una guía de elaboración de bloques, la cual se encuentra en la documentación en línea de Moodle. Se describe paso a paso como hacer la instalación.

Como primer paso se tiene que copiar la carpeta que contiene el script del bloque en la carpeta "blocks" del EVA que se encuentra en esta ubicación "C:\xampp\htdocs\eva\blocks"

En el Entorno Virtual de Aprendizaje que está instalado localmente, ingresamos como administrador y accedemos a notificaciones en la parte derecha, esto hará que se instalen nuevos componentes del entorno en este caso la carpeta "pendiente_tareas" y el bloque quedara instalado.

La siguiente pantalla muestra que nuestro bloque se ha instalado correctamente:

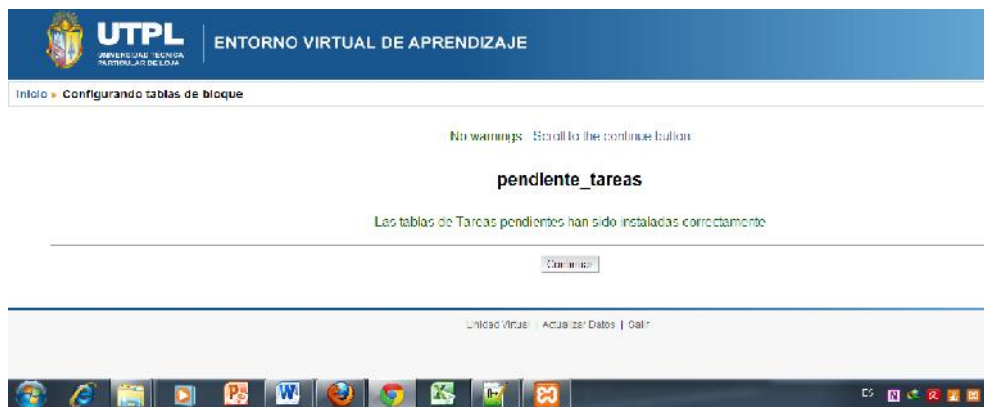


Gráfico 21 Instalación bloque tareas pendientes en el EVA

Se pone en continuar, ahora para finalizar con la instalación se tiene que activar la edición del Entorno, esto se realiza en la pantalla de inicio "activar edición" para luego ubicarnos en la carpeta Módulos de la barra de administración aquí nos ubicamos en bloques, luego se da click en bloques Sticky, aparecen dos opciones se elige las opción páginas del curso.

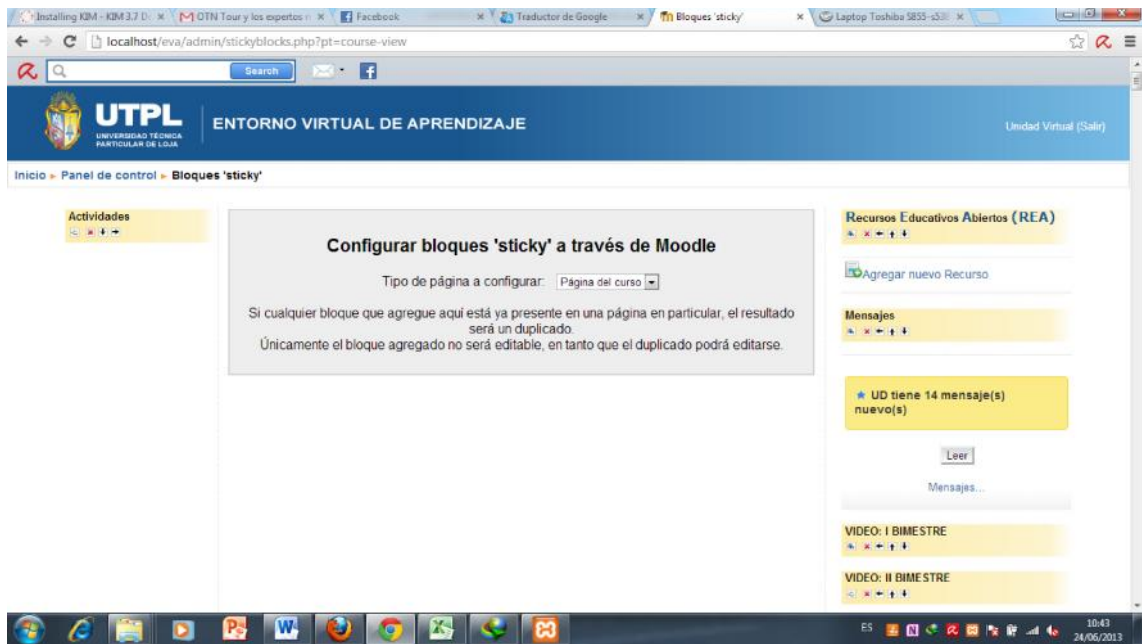


Gráfico 22 Colocar el bloque en la página de los cursos

En la parte derecha nos aparece la opción de bloques y una pestaña para agregar el bloque de tareas pendientes

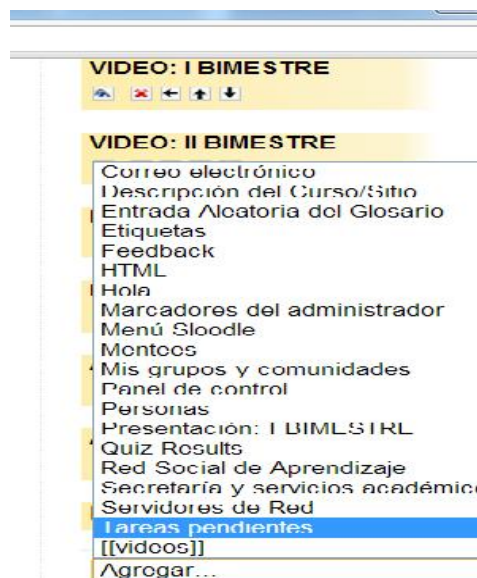


Gráfico 23 Lista de bloques para agregar a la página de los cursos

Con esta opción se mostrara las tareas pendientes de cada curso en el Entorno pero en la página de cada curso. Se ha activado un curso para que muestre todos sus componentes y también muestre las tareas pendientes de este curso así como se muestra en la siguiente imagen:



Gráfico 24 Bloque instalado en el EVA

Como se puede observar en este curso “Estructura de Datos II” hay dos Tareas y un foro que el estudiante no ha resuelto. Se tiene que considerar cursos donde las tareas pendientes cuenten con una descripción o títulos bastante específicos ya que esta información es utilizada para crear las entidades que luego van a ser buscadas dentro de bloque REA.

Anexo 3.

Instalación y configuración de un entorno de pruebas local

Para continuar con el desarrollo del Agente Recomendador se ha creído conveniente la construcción de un ambiente de pruebas donde se lo aplique a datos reales tomados de cursos que se han realizado en el Entorno Virtual de Aprendizaje (EVA) de la UTPL.

En este capítulo se describe la creación de tal ambiente que es una réplica del sistema que se maneja en producción en la Universidad, la descripción de las herramientas a utilizar y sus versiones, los diferentes cambios realizados para el levantamiento del sistema y activación del curso sobre el cual trabajar con el Agente.

Como primer paso se han instalado las herramientas necesarias para instalar el EVA de forma local, además herramientas que sirvan para modificar información en bases de datos, al final se ha instalado el sitio del entorno para usarlo en cualquier navegador del computador.

Instalación de las herramientas a utilizar

El Entorno Virtual de Aprendizaje al estar basado enteramente en Moodle necesita de las mismas herramientas para su instalación, que son conocidas como plataforma LAMP (Linux Apache Mysql y PHP). Moodle ha sido desarrollado en Linux por esta razón aparece en las herramientas, no obstante hay como instalarlo en cualquier sistema operativo. Para este entorno se utiliza Windows como sistema operativo.

Como servidor Web se utiliza Apache, el lenguaje que se utiliza es PHP y finalmente se utiliza un motor de base de Datos en este proyecto se utiliza el Mysql.

Hay que mencionar también que en la universidad se trabaja con el Moodle en su versión 1.9.

Hay diferentes aplicaciones que ofrecen en conjunto todas estas herramientas se ha decidido utilizar Xampp que trae consigo al servidor Apache, PHP, Mysql y Perl.

Es fácil de instalar y de usar, la versión que se ha instalado es una acorde a las necesidades del Entorno de la Universidad esta es la versión 1.6.8

Xampp 1.6.8: es un conjunto de herramientas para facilitar al desarrollador la creación de entornos basados en el servidor web Apache, PHP y Mysql. El paquete de esta versión viene con las siguientes herramientas

- PHP en su versión 5.2.6
- Mysql en su versión 5.0.6
- Apache 2.2.9

Mysql TOAD: con la instalación de Mysql, se necesita de una herramienta para manejar los datos del entorno y también modificar algunas tablas ya que es necesario para activar periodos y cursos.

Mysql Workbench: igualmente esta herramienta nos permite el manejo de base de datos

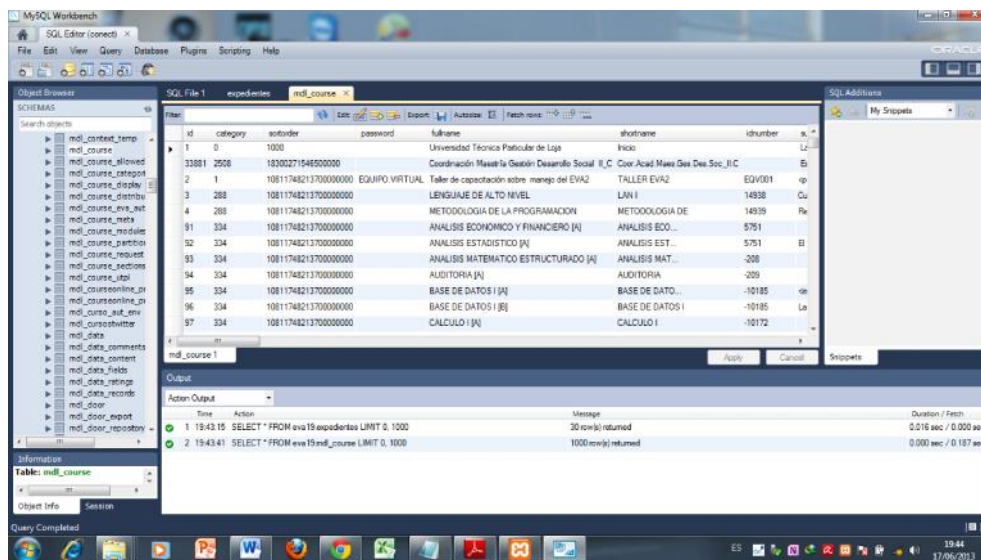
Una vez que se ha realizado la instalación de estas herramientas, se han pedido dos carpetas que contienen la base de datos y la información del sitio para migrarlo a forma local, estos son

- Bases de EVA: archivos de tipo SQL, con información de toda la base de datos del Entorno de la UTP
- EVA: carpeta del LMS adecuado a la Universidad con la información del sitio del Entorno Virtual de Aprendizaje

Con esta información se empieza por el levantamiento de la base de datos.

a) Levantamiento de la base de Datos

El Sistema de Gestión de Información Moodle utiliza una base de datos para poder llevar a cabo los diferentes cursos, el entorno de la UTP basado en Moodle 1.9 cuenta con esa base de datos con un tamaño bastante grande por la interacción de estudiantes en el entorno y la cantidad de recursos que se utilizan en cada curso, la base que se obtuvo es de 20GB de tamaño para su importación se ha utilizado la herramienta Workbench, la importación dura bastante tiempo.



Luego de la importación en la imagen se muestra las diferentes tablas del Entorno Virtual de aprendizaje de la UTP, las cuales se utilizan para el desarrollo de la presente investigación, además cabe indicar que en esta base de datos se encuentran las tablas del bloque REA (Recursos Educativos Abiertos) de donde se van a utilizar los recursos para hacer las recomendaciones.

Para poder levantar el sitio del Entorno Virtual de Aprendizaje se ha necesitado realizar algunas modificaciones en la base de datos los cuales se muestran en el anexo B. Estos cambios se relacionan con el acceso de administrador al sitio y con la activación de cursos debido a que esos cursos fueron del periodo anterior.

b) Levantamiento del sitio Eva

Para el levantamiento de un sitio Moodle, se copia la carpeta de Moodle en las carpeta httdocs del Xampp, para el entorno virtual de Aprendizaje es lo mismo se copia toda la carpeta EVA, en la carpeta httdocs y se realizan cambios en el archivo "config.php" en las siguientes líneas:

```
CFG->dbtype = 'mysql';
$CFG->dbname = 'eva19'; // aqui tenemos que poner el nombre de la base de
datos
$CFG->dbuser = //'root'; nombre del usuario de la base de datos
$CFG->dbpass = ""; // si la base tiene clave se la coloca aqui
$CFG->prefix = 'mdl_'; es el prefijo de nuestra base de datos

$CFG->wwwroot = 'http://localhost:80/eva'; la dirección con la cual ingresamos
al entorno
$CFG->wwwservices = 'http://172.16.95.14:8080'; //172.16.56.14 //para cabiar
la url de los servicios web
$CFG->dirroot = 'C:\xampp\htdocs\eva'; // ruta de la instalacion
```

Hay que también cambiar el límite de memoria ya que el Entorno Virtual de aprendizaje necesita más de la que se da por defecto en el archivo "php.ini", esto se lo hace con cambiar la línea siguiente:

```
memory_limit = 1024M ; Maximum amount of memory a script may consume
(16MB)
```

En nuestro caso se ha modificado el valor a 1GB de memoria

Luego de haber realizado estas modificaciones, ya tenemos nuestro sitio montado localmente que es una réplica del sistema utilizado en producción en la UTPL, sobre este realizaremos la implementación y pruebas respectivas del agente Recomendador

La siguiente imagen nos muestra la pantalla principal para poder ingresar al entorno virtual de aprendizaje a los diferentes cursos y al ingresar como administrador se pueden realizar cualquiera de los cambios que se debe realizar para el funcionamiento del Agente Recomendador.



Gráfico 25 Pantalla principal de acceso al EVA

Para el ingreso se ha utilizado la cuenta de administrador con el nombre de usuario “unidad.virtual” el cual otorga toda la administración del sitio y el amplio manejo del mismo.

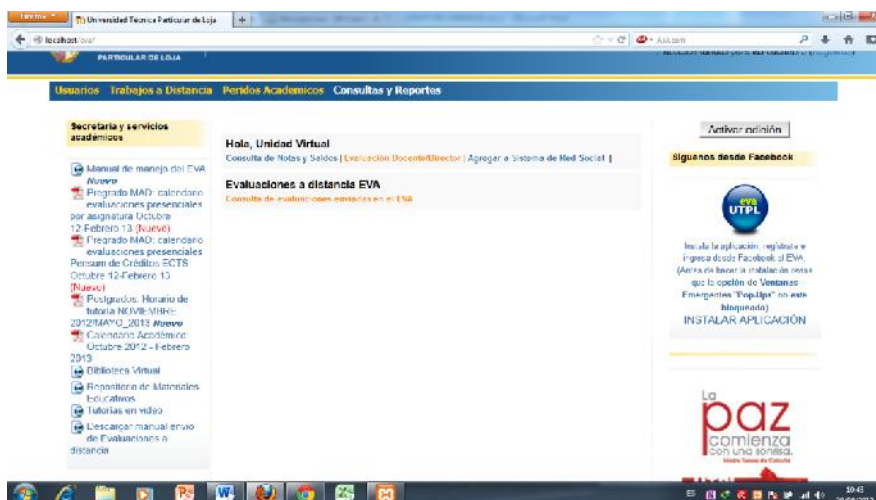


Gráfico 26 Pantalla de administración del EVA

En la parte izquierda encontramos una barra donde podemos administrar todo el sitio y el que se utiliza para realizar la investigación, se trabaja sobre los periodos y cursos que serán de gran ayuda para realiza las recomendación de recursos.



Gráfico 27 bloque de administración del EVA

Configuraciones de la base de datos entorno de pruebas

Hubo que realizar algunas configuraciones en la Base de datos del EVA, que se han realizado para poder acceder como administrador y para poder activar el curso sobre el cual se desea trabajar, estas modificaciones se han realizado con la herramienta TOAD

En la tabla “mdl_user” se ha cambiado el campo “password” con el campo de nuestro usuario en el EVA, esto para poder acceder con privilegios de Administrador al sitio. La cuenta que se ha modificado es la “unidad.virtual” la cual cuenta con todos los permisos de administración

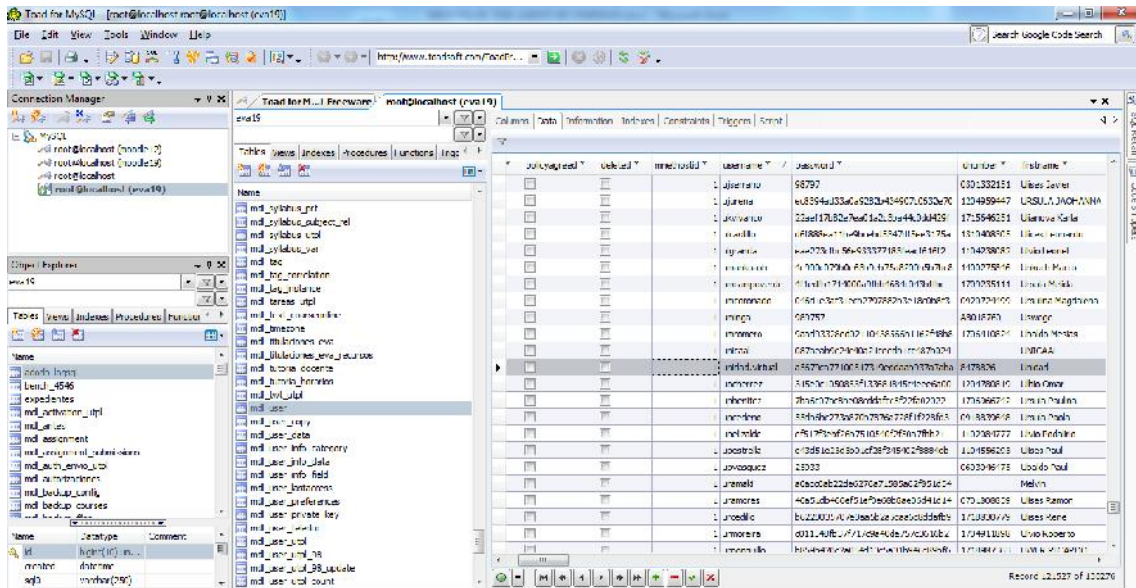


Gráfico 28 Cambio clave para acceder como administrador

El segundo paso que se ha realizado es cambiar los datos en el periodo académico en este caso la tabla “mdl_syllabus_pdo”, se cambia la fecha del fin del periodo académico

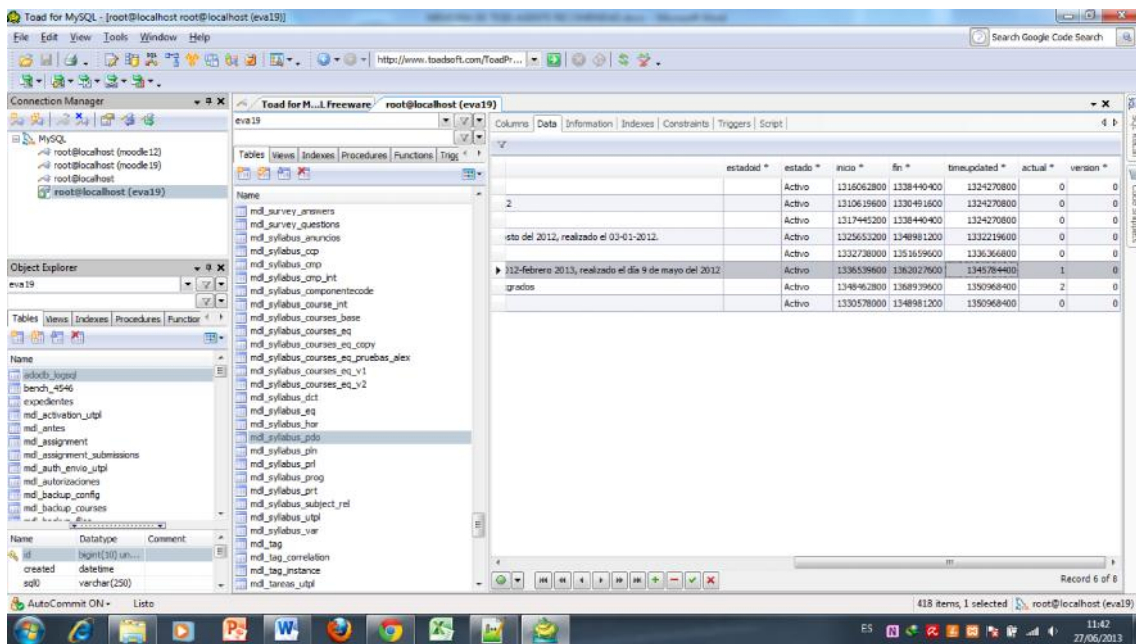


Gráfico 29 Cambio de fecha del periodo

Por ultimo cambiamos en el curso la fecha de finalización, es decir el campo “enrollenddate”, en este caso se ha realizado los cambios sobre el curso de “Estructura de Datos”

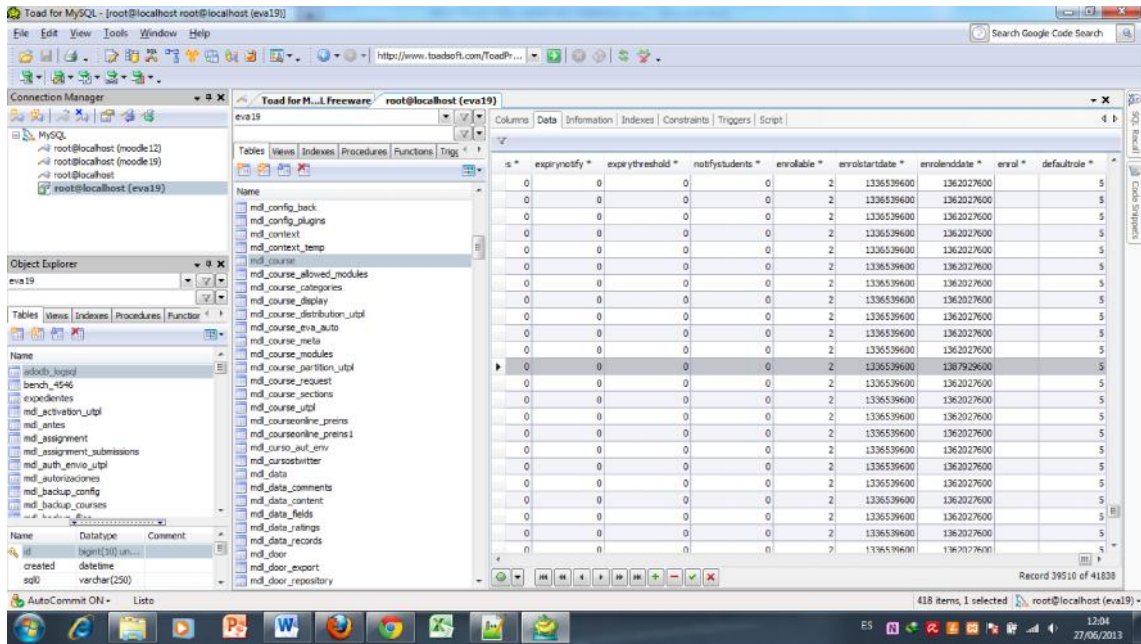


Gráfico 30 Cambio de fecha en los cursos

Anexo 4

Instalación de la plataforma KIM

Para la instalación de KIM se requiere del java development Kit 1.6 , es necesario utilizar esta versión ya que con la reciente que es la 1.7 no funciona. Se ha instalado JDK1.6.0_43 y se ha agregado la respectiva variable en variables de entorno del sistema.

Para realizar esto primero nos ubicamos en variables de entorno, se hace click derecho en propiedades del equipo, en la parte izquierda se da click en configuración avanzada del sistema y a continuación en variables de entorno como se muestra en la siguiente imagen.

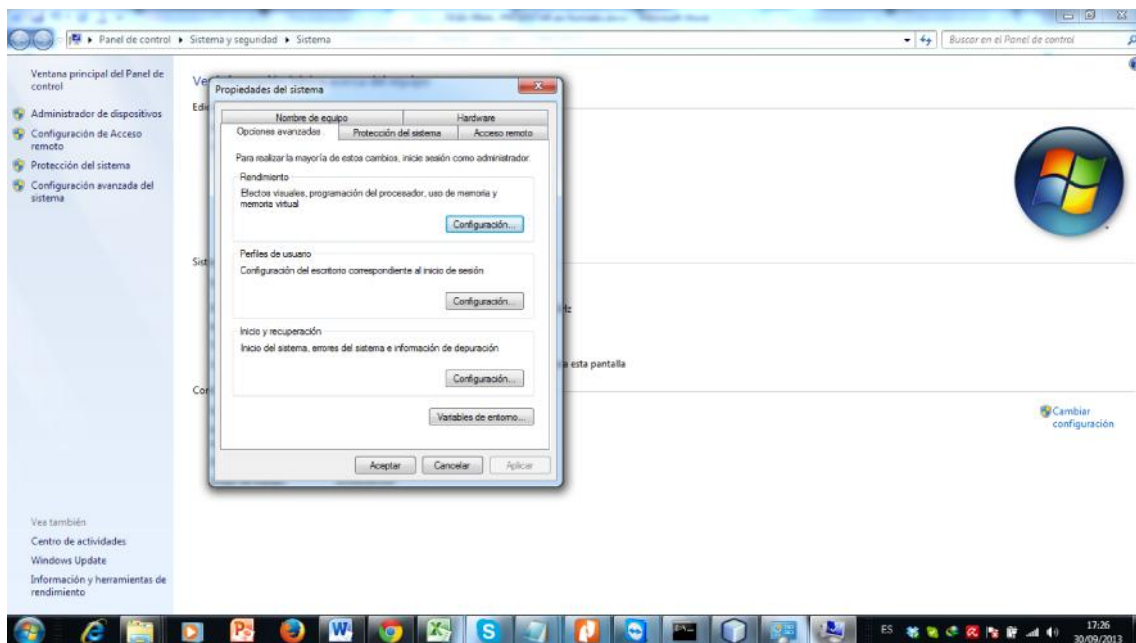


Gráfico 31 Agregar variables de entorno

Ahora creamos una nueva variable con el nombre JAVA_HOME y la ruta le ponemos donde esta instalado el kit de desarrollo de java
C:\Program Files\Java\jdk1.6.0_43

Se ha descargado la plataforma Kim desde la página de <http://www.ontotext.com/kim/KIM-downloads.html>, los requerimientos de esta plataforma son los siguientes:

Intel Pentium IV 1GHz or compatible processor (over 2,5GHz recommended)

- 1,5 GB total system RAM as a minimum (2 GB recommended)
- 500 MB de espacio en disco
- Linux, Mac OS or Microsoft Windows (2000 SP2 / XP / 2003 / Vista / 7)
- JDK SE 1.5.0 or 1.6.0
- Servlet container or a J2EE/JEE application server (for the Web User Interface):
 - Apache Tomcat Server - 6.0 or later
 - Mortbay Jetty 6.1 or later
 - Sun Glassfish v2 or later
 - RedHat JBoss 4.2.3 or later

En el caso del proyecto de tesis se ha utilizado el jdk 1.6, y como servidor web apache tomcat 7.0.34

Para el desarrollo de los web services se ha utilizado net Beans en donde esta integrado el tomcat

Instalación

Se debe de extraer el contenido del paquete bajado desde internet en la ubicación donde se desea instalar: C:\kim_home\kim-platform-3.7

Crear la variable de entorno para java para que pueda ser utilizado Kim, también se puede modificar en el archivo config de las carpetas kim-platform-3.7/bin/config/ descomentando la línea: Set java_home="ponemos la dirección de java"

Para inicializar kim, se pone "kim start" en la consola de comandos o entramos a las carpeta bin y lo ejecutamos desde ahí.

La pantalla muestra que el kim se ha iniciado satisfactoriamente si todo ha salido bien.

Configuración de la interfaz WEB

Para la configuración de interfaz del KIM donde podemos acceder a la población es decir a todos los documentos que tenemos anotados, se tiene que compilar el

archivo Kim.war en nuestro servidor tomcat para esto tenemos que copiar el archivo que se encuentra en C:\kim_home\kim-platform-3.7\services\http\webapps y copiarlo en la carpeta de aplicaciones web del tomcat que es la siguiente: C:\Users\Freddy\AppData\Roaming\NetBeans\7.3.1\apache-tomcat-7.0.34.0_base\webapps

Luego de esto al iniciar nuestro servidor web tomcat podemos acceder a esta interfaz mediante la dirección localhost:8084/KIM

Se presenta una página como la siguiente la cual es del KIM WEB UI

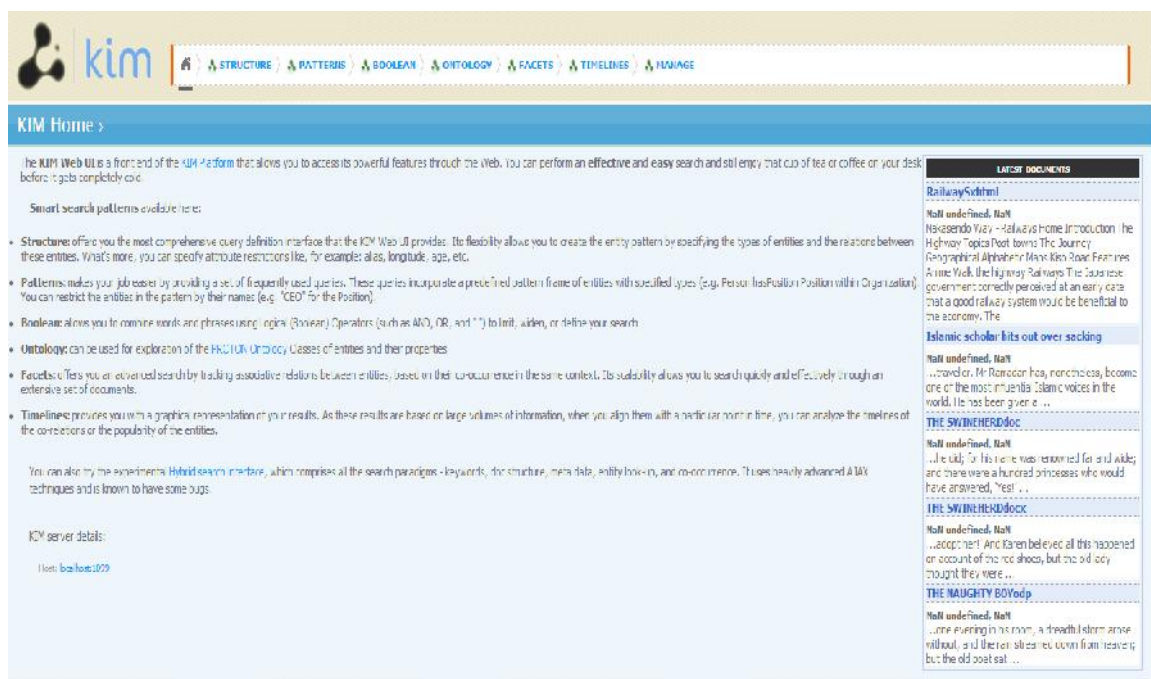


Gráfico 32 Interfaz de usuario de kim

Mediantes esta interfaz de usuario podemos ver todas las anotaciones que hemos realizado en el proyecto, como también todos los documentos que están guardados en el repositorio de kim

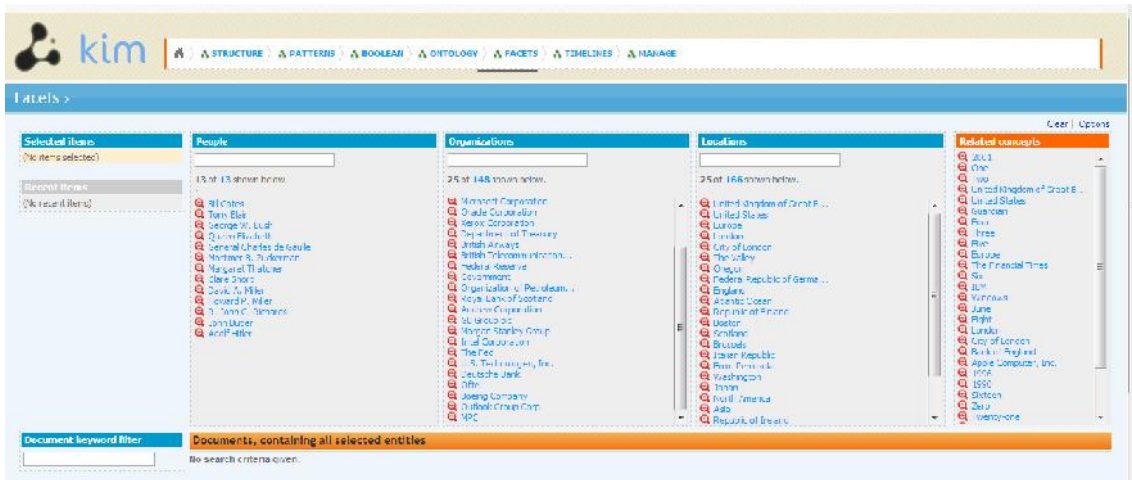


Gráfico 33 Entidades en la interfaz de kim

También se puede observar la ontología de Kim

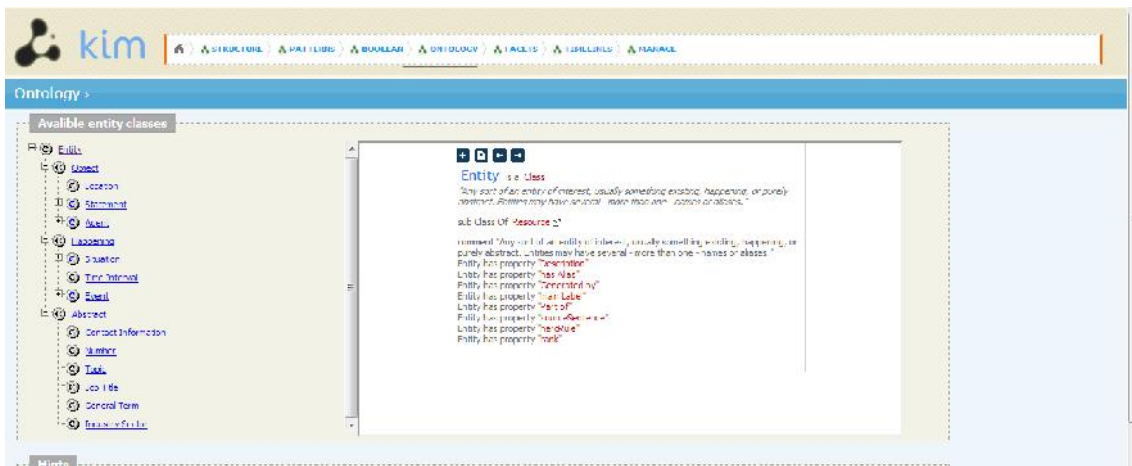


Gráfico 34 Ontología de la plataforma kim

Anexo 5

Desarrollo de los Servicios web

Este ha sido un paso esencial para la construcción del agente, ya que se ha elaborado los servicios web que serán consumidos por el cliente realizado en php en el bloque de recomendación.

Para la construcción de estos servicios se ha utilizado el entorno de desarrollo Net Beans IDE y como contenedor de Servlets se lo ha instalado su componente Tomcat, se tiene dos servicios web el uno se encarga de realizar las anotaciones semánticas y el otro se encarga en si de recomendar los recursos.

Creación del Web Service WSanotador

Este servicio web tiene la funcionalidad de anotar las descripciones de las tareas pendientes y los recursos educativos abiertos y subirlos en los repositorios de KIM, todo esto se logra con la utilización de los APIs de KIM, que tienen diferentes métodos tanto para guardar las anotaciones semánticas como para manejar estas anotaciones, en si este servicio posee tres métodos principales:

- Conectar: en este paso se conecta con la plataforma kim para utilizar todos sus funcionalidades, el resultado de este método es tener un objeto del tipo kim para realizar el anotado semántico
- Procesar Reas: dentro de este método el servicio web lo que hace es procesar los recursos educativos abiertos y los envía a anotar
- Anota: método sumamente importante ya que es el encargado de anotar las tareas y los recursos en los repositorios de kim.
- Recomendar: es un método parte del servicio web en el cual se manda a anotar las tareas.

El siguiente es el código de este servicio web con sus diferentes métodos, hay que recalcar que existen varias capas lógicas que son utilizadas por los métodos de este servicio el código es solo del servicio web y sus métodos principales

```
package logical;
import com.google.gson.Gson;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.Iterator;
```

```

import java.util.logging.Level;
import java.util.logging.Logger;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;

/**
 * @author Usuario
 */

@WebService(serviceName = "WSanotador")
public class WSanotador {
    public static kim rmi = null;
    lector lecto;

    /**
     * Web service operation
     */
    @WebMethod(operationName = "conectar")
    public String conectar(@WebParam(name = "server") String server) {
        //TODO write your implementation code here:

        rmi = new kim();
        return rmi.conectar( server );
    }

    /**
     * Web service operation
     */
    @WebMethod(operationName = "Procesar_reas")
    public String Procesar_reas(@WebParam(name = "datos") String datos) {
        //TODO write your implementation code here:
        rmi = new kim();
        rmi.conectar( "localhost" );
        Gson gson = new Gson();
        rea item = gson.fromJson(datos, rea.class);
        String msg = "";
        if( item.getTipo().equals("URL")){

```

```

        Url uri = new Url();
        lecto = uri.get_contenido( item.getUrl() );
        if( lecto.getError_url().isEmpty() ){//Error
            msg = anota( rmi, item, item.getUrl(),1 );
        }else{
            msg = lecto.getError_url();
        }

    }else{//Archivos
        String ocn = item.getResumen();
        msg = anota( rmi, item, ocn,2 );

    }
return msg;
}

```

```

public String anota( kim rmi, rea item, String contenido, int modo ) {

    String msg = "";
    String anotado = "";
    if(modo == 1){//URLS
        URL url;
        try {
            url = new URL(contenido);
            msg = rmi.set_contenido_url(url);
        } catch (MalformedURLException ex) {
            Logger.getLogger(WSanotador.class.getName()).log(Level.SEVERE, null,
ex);
            // msg = ex.getMessage();
            return ex.getMessage();
        }
    }else{//archivos
        msg = rmi.set_contenido_doc(contenido);

    }
}

```

```

rmi.set_titulo(item.getTitulo());
rmi.set_URL( item.getUrl() );
rmi.set_autor( item.getAutor());
rmi.set_fecha( item.getFecha() );
rmi.set_key_frases( item.getKeywords() );
rmi.set_key_frases( item.getResumen() );

if(msg.equals("ok")){
    anotado = rmi.guarda_doc();
}
return anotado;
}
/**
 * Web service operation
 */
@WebMethod(operationName = "recomendar")
public String recomendar(@WebParam(name = "tarea") String tarea) {
    //TODO write your implementation code here:
    String err = "nada";
    Iterator anote = rmi.anota_tarea(tarea);
    if( anote != null ){
    }else{
        err = "Se ha producido un error al buscar los reas";
    }
    return err;
}
}

```

Creación del Web Service WSagente

Este servicio web es el encargado de realizar en si las recomendaciones debido a que ya se tiene las tareas y los recursos educativos abiertos anotado mediante este método realizamos una búsqueda en primera instancia semántica y luego una sintáctica para encontrar los documentos que se relacionen con entidades de la descripción de la tarea y puede recuperar esos documentos:

Los métodos de este servicio web son los siguientes:

- Conectar: al igual que el otro servicio web este debe de conectarse para utilizar los servicios de Kim
- Recomendar: es un método principal del servicio web, en donde se recibe como parámetro una actividad que sería la descripción de la tarea y se realiza una búsqueda semántica de las anotaciones de las mismas para dar como resultado el o los documentos reas que satisfacen la consulta.
- Consulta Sintáctica: este método ha sido añadido para que si el servicio web no encuentra los documentos de forma semántica los encuentre de forma sintáctica.

El código de este servicio es el siguiente:

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package logical;

import com.google.gson.Gson;
import com.ontotext.kim.client.corpora.KIMAnnotationSet;
import com.ontotext.kim.client.query.SemanticQuery;
import java.util.ArrayList;
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import kim.kdocs;
import kim.kim_service;

/**
 *
 * @author Usuario
 */
@WebService(serviceName = "appAgenteWS")
public class appAgenteWS {

    public static kim_service kim ;
```

```

Gson data;
private String resp;
/*
 * Web service operation
 */
@WebMethod(operationName = "Recomendar")
public String Recomendar(@WebParam(name = "actividad") String actividad) {
    //TODO write your implementation code here:
    data = new Gson();

    kim = new kim_service("", "", "", "", "");
    //kim.conectar( "localhost" );

    KIMAnnotationSet anotations = kim.recomendar(actividad);

    if( kim.getE_annotado_sem().isEmpty() && !anotations.isEmpty() ){//SI hay error al
anotar no se puede realizar la consulta semantica

        //SemanticQuery seq = kim.construye_query_semantica(anotations);
        SemanticQuery seq = kim.construye_query_semantica2(anotations);

        if( kim.getE_make_seq().isEmpty() ){//todo va bien

            resp = consulta_sintactica( anotations );

            ArrayList kbdocs = kim.ejecuta_consulta_semanticamente(seq); //Empieza
query semantico

            if( kim.getE_exe_seq().isEmpty() && kbdocs.size() > 0 ){

                resp = "[";
                for( int i=0; i < kbdocs.size(); i++){

                    kdocs docs = (kdocs) kbdocs.get(i);
                    if( i == kbdocs.size() -1 ){//ultimo elemento no va con coma al final
                        resp += data.toJson(docs);

```

```

        }else{
            resp += data.toJson(docs)+" ";
        }
    }
    resp += "]";
}
else{
    resp = consulta_sintactica( anotations );
}
//end else */ //termina semantica

}
else{//consulta sintacticamente
    resp = consulta_sintactica( anotations );
}
//end else*/

}
else{//se manda el json vacio, los errores se revisan en los logs
    resp = "[]";
}

return resp ;

}
//Recomendar

/**
 * Web service operation
 */
@WebMethod(operationName = "Conectar")
public String Conectar(@WebParam(name = "host") String host) {
    //TODO write your implementation code here:

    kim = new kim_service("", "", "", "", "");

    return kim.conectar( host );
}

```

```

public String consulta_sintactica( KIMAnnotationSet anotations ){

String qs = kim.construye_query_sintactica(anotations);
String resp = "";
ArrayList kbdocs = kim.ejecuta_consulta_sintacticamente( qs );

if( kim.getE_exe_sint().isEmpty() ){

    resp = "[";
        for( int i=0; i < kbdocs.size(); i++){

            kdocs docs = (kdocs) kbdocs.get(i);
            if( i == kbdocs.size() -1 ){//ultimo elemento no va con coma al final
                resp += data.toJson(docs);
            }else{
                resp += data.toJson(docs)+",";
            }
        }
        resp += "]" ;

    }else{
        //resp = kim.getE_exe_sint();
        resp = "[]";
    }

    return resp;

} //end consulta_sintactica
} //end class

```

Anexo 6

Desarrollo del Bloque final como cliente que consume los servicios web

En esta parte se ha desarrollado el código de bloque final denominado “recomendador” el cual incluye la parte del código donde se filtran las tareas pendientes realizado en un inicio.

La carpeta del bloque consta de lo siguiente:

- el archivo del bloque
- un archivo process.php el cual contiene el cliente del bloque que consume los servicios web
- Carpetas del jquery para la interfaz gráfica en el EVA

Para la comunicación del cliente con los servicios web se ha bajado nusoap y se lo ha instalado en el EVA.

Instalación de NUSOAP

Nusoap es un kit de herramientas para el desarrollo de web service en PHP, con el cual se pueden realizar clientes para que consuman servicios web o servidores los que proveen servicios web, en el caso del proyecto de tesis se tiene que crear un cliente para que se consuma los servicios web.

Es una herramienta madura por eso se la ha escogido y además su instalación es bastante sencilla solo basta con descargar el archivo comprimido de la página oficial y luego se lo descomprime en el proyecto donde se lo requiere, en este caso la carpeta es C:\xampp\htdocs\eva

Con esta instalación se puede consumir ya los servicios web que ya están elaborados para realizar las recomendaciones en el EVA que se tienen instalado localmente.

La codificación del cliente es la siguiente y es el archivo Process.php ubicada en la carpeta del bloque.

```
<?php
```

```
/*
```

```
* To change this template, choose Tools | Templates
```

```
* and open the template in the editor.
```

```

*/
header("Content-Type: text/html; charset=UTF-8");
global $USER, $CFG,$COURSE;
require_once('db.php');
require_once(".././config.php");
require_once(".././nusoap/lib/nusoap.php");

$userid = $_POST['usuario'];
$curso = $_POST['curso'];
$modo = $_POST['modu'];
$conect_wx = null;
$COURSE = get_record("course", "id", $curso);
$recomendaciones = "";

if( $modo == "recomendar"){

    $conex_rmi= conectar_Kim( 'localhost' );

    if( $conex_rmi->status == "ok"){//Conexio Exitosa
        $conex_rmi->status;
        $conect_wx = $conex_rmi->cliente;

        $recomendaciones = /*<div id="tabs-1">' .*/ get_rec_tareas( $userid,
$COURSE ) /*.</div>'*/;
        $recomendaciones1 = /*<div id="tabs-2">' .*/ get_rec_foros( $userid,
$COURSE->id ) /*.</div>'*/;

        if( $recomendaciones == "" || $recomendaciones == null ||
$recomendaciones1 == "" || $recomendaciones1 == null ){
            echo "<h1>No existen tareas pendientes</h1>";
        }else{

            echo str_replace("|", "", $recomendaciones) . "|" . str_replace("|", "",
$recomendaciones1);

        }
    }
}

```

```

        }else{//Muestra el error
            echo $conex_rmi->status;
        }

    }else{
        $items = get_reas( $COURSE->id);

        if( $items ){
            $conex_rmi= conectar_Kim( 'localhost' );
            if( $conex_rmi->status == 'ok' ){//si conecta con el servicio e inicia el KIM se
realizará estas funciones

                anotar_reas($items);

            }else{
                echo $conex_rmi->status;
            }

        }else{

            echo "No hay items que anotar";

        }

    }

/*
* FUNCIONES
*/
function get_rec_foros( $userid, $cursoid ){

    $forums = get_records('forum', 'course', $cursoid);

    foreach ($forums as $forum) {
        if(!record_exists("forum_discussions", "userid", $userid, "forum",
$forum->id)){

```

```

        $no =0;
        $kdocs = get_recomendacion( strip_tags( $forum->intro ) );
        $recom = "";
        if($kdocs){
            foreach( $kdocs as $kdoc){
                if( $kdoc->error_consulta == "" ){
                    $recom .= '<p><a target="_blank" href=".'.$kdoc->url.'>.'.$kdoc->titulo.'</a></p>';
                    $no++;
                }
            }
        }
        }else{
            $recom = "<div><p>Sin Recomendaciones por el momento</p></div>";
        }

        $tasker .= "<h3>". /*substr(* $forum->name/* , 0, 15)*/ ." ($no)</h3>";
        $tasker .= "<div>".$recom."</div>";
    //}
} //end foreach
return $tasker;
} //finaliza la funcion de extraer los foros pendientes

function get_rec_tareas( $userid, $curso ){
    $modinfo = get_fast_modinfo( $curso );
    $tasker = "";
    foreach ( $modinfo->instances['assignment'] as $cm ) {
        if(!record_exists("assignment_submissions", "userid", $userid, "assignment", $cm->instance)){
            //$tblPendientes .= "<b>Tarea: ".$cm->name."</b><br>";
            $no =0;
            $tarea = get_record("assignment", "id",$cm->instance );
            if( $tarea ){

```

```

        $kdocs = get_recomendacion( strip_tags($starea->description) );
    }//end if

    $recom = "";
    if($kdocs){

        foreach( $kdocs as $kdoc){

            if( $kdoc->error_consulta == "" ){
                $recom .= '<p><a target="_blank" href=".'.$kdoc->url.'">'.$kdoc-
>titulo."</a></p>";
                $no++;
            }

        }

    }else{
        $recom = "<div><p>Sin Recomendaciones por el momento</p></div>";
    }

    $tasker .= "<h3>". substr($starea->name,0,50) ." ($no)</h3>";
    $tasker .= "<div>".$recom."</div>";
} //end if
} //end foreach

return $tasker ;

} //finaliza la función de extraer las tareas pendientes

function get_recomendacion( $starea ){
    global $conect_wx;

    try{
        $res = $conect_wx->Recomendar(array( 'actividad'=>$starea));
        $reas1 = json_decode($res->return);
        $reas = depura_json($reas1);
    }catch (SoapFault $exception) {

```

```

//echo $exception;
$reas = "";

}

return $reas;
} //end function

function depura_json($reas){
    $data = "";
    $arr = array();
    foreach( $reas as $kdoc){

        if( $kdoc->error_consulta == "" && !estas($data, /*str_replace("|", "", */$kdoc->titulo*/) ) ){
            $arr[] = $kdoc;
            $data.= /*str_replace("|", "", */$kdoc->titulo*/) . "|";
        }

    }

    return $arr;
}

function estas($data, $dato){
    $ban = false;
    $vars = explode("|", $data);

    foreach($vars as $var){

        if($var == $dato){

            $ban = true;

        }

    }

} //edn foreach

```

```
return $ban;
} //edn function
```

```
function get_reas( $curso){

    $codigo=get_record('syllabus_utpl', 'courseid', $curso);

    if ($codigo){

        $cursos_reas1=get_records('syllabus_utpl', 'componenteCode', $codigo->componenteCode);
        $cursos_reas2=get_records('course_utpl', 'componenteCode', $codigo->componenteCode);

        $reas_ant="";
        foreach ($cursos_reas1 as $recu1) {
            $reas_ant=$reas_ant.$recu1->courseid.'';
        }

        $reas_ant = substr($reas_ant, 0, -1);
set_time_limit(0);
        $sql = "select r.*,ra.ranking from mdl_rea_historico_recurso a ,
mdl_rea_recurso r,mdl_rea ra where r.id = a.recurso and a.recurso=ra.rea_id and
r.borrado = 0 and a.curso IN ($reas_ant) and r.titulo not like '%clave%' and r.titulo not
like '%bimestre%' and r.titulo not like '%supletorio%' ORDER BY ra.ranking DESC, r.id
ASC ";//limit 6 GROUP BY r.titulo

    }else{
        $sql = "select r.*,ra.ranking from mdl_rea_historico_recurso a ,
mdl_rea_recurso r,mdl_rea ra where r.id = a.recurso and a.recurso=ra.rea_id and
r.borrado = 0 and a.curso IN ($curso) and r.titulo not like '%clave%' and r.titulo not like
'%bimestre%' and r.titulo not like '%supletorio%' ORDER BY ra.ranking DESC, r.id
ASC ";//limit 6 GROUP BY r.titulo
```

```

}

$items = get_records_sql($sql);

if( !$items ){

    $items = null;

}

return $items;

} //end get_reas
function conectar_Kim( $server_kim ){
    //$client = null;
    $obj = new stdClass();

    try{
        $client = new SoapClient("http://localhost:8084/app_agente/appAgenteWS?wsdl");

        $res = $client->conectar(array( 'host'=>$server_kim));

        $obj->status = $res->return;
        $obj->cliente = $client;

    }catch (SoapFault $exception) {

        $obj->status = $exception;

    }
}

```



```

return $obj;
} //end conectar_Kim

function anotar_reas( $items ){

    foreach($items as $item){

        $recurso = new object();
        $recurso->item = $item->id;
        $recurso->titulo = $item->titulo;
        $recurso->autor = get_autores_rea($item->id);
        $recurso->fecha = time();
        $recurso->curso = $item->course_id;
        $recurso->resumen = "$item->resumen. $item->descripcion ";
        $recurso->keywords = get_keyword_rea($item->id);

        $var = anota_urls($recurso);
        if( $item->resumen != "" && $item->descripcion != "" && $item->resumen !=
null && $item->descripcion != null ){
            if ( strlen($item->resumen) > 20 || strlen($item->descripcion) > 20){
                $var = anota_files($recurso);
            }
        }

    } //end foreach

    return $var;

} //end anotar_reas

function get_autores_rea( $item ){

    $autor = 'UTPL';

```

```

$infos = get_records('rea_item_autor', 'item', $item);

if($infos){

    foreach($infos as $info){
        $autor = "$info->nombre $info->apellido";
        //break;
    }

}

return $autor;

} //get_info_base

function get_keyword_rea( $item ){

    $keys = "";

    $infos = get_records('rea_keyword_recurso', 'id_item', $item);

    if($infos){

        foreach($infos as $info){

            $keys .= str_replace("|", ", ", utf8_encode($info->keyword)). '|';

        }

    }

    }else{
        $keys = 'UTPL';
    }

}

```

```

return $keys;

} //get_info_base

function anota_urls( $recurso ){
    $urls = get_records('rea_items_url', 'item', $recurso->item);

    if( $urls ){
        $recurso->tipo = 'URL';
        foreach( $urls as $url ){
            if( !esta_annotado( $recurso->item, $url->id ) ){
                $recurso->url = $url->uri;
                $recurso->sud_id = $url->id;

                $var = anota_recurso($recurso);
                //break;
            }

        } //end foreach

    } //end if
    return $var;
} //end funtion

function anota_files( $recurso ){
    global $dataroot,$wwwroot;
    $archivos = get_records('rea_itemsyarchivos', 'item', $recurso->item);

    if( $archivos ){
        $recurso->tipo = 'file';
        foreach( $archivos as $fill ){
            if( !esta_annotado( $recurso->item, $fill->id ) ){
                // print_object($recurso, "desde aca");
                //if(file_exists( "$dataroot\\$recurso->curso\\$fill->name_archivo")){

```

```

        $recurso->url      =      "$wwwroot/file.php/$recurso->curso/$fill-
>name_archivo";
        $recurso->sud_id = $fill->id;
        /*$recurso->name_archivo = $fill->name_archivo;
        $recurso->peso = $fill->size_bytes;
        $recurso->formato = $fill->size_bytes;*/

        $var = anota_recurso($recurso);
    //}
}

```

```

    //end foreach

```

```

} //end if
return $var;

```

```

} //end funtion

```

```

function anota_files0( $recurso ){
    global $dataroot,$wwwroot;
    $archivos = get_records('rea_itemsyarchivos', 'item', $recurso->item);

    if( $archivos ){
        $recurso->tipo = 'file';
        foreach( $archivos as $fill ){
            if( !esta_annotado( $recurso->item, $fill->id ) ){

                if(file_exists( "$dataroot\\$recurso->curso\\$fill->name_archivo")){
                    $recurso->url      =      "$wwwroot/file.php/$recurso->curso/$fill-
>name_archivo";

                    $recurso->sud_id = $fill->id;
                    $recurso->name_archivo = $fill->name_archivo;
                    $recurso->peso = $fill->size_bytes;
                    $recurso->formato = $fill->size_bytes;

                    $re = get_dir_url_temp($recurso);
                }
            }
        }
    }
}

```

```

        if($re){
            $recurso->uri_prc = $re;

            anota_recurso($recurso);
            //print_object( $recurso );
        }else{

            registra_evento($recurso, "Error al leer archivo", 0);
        }

        //break;
    }
}

} //end foreach

} //end if
} //end funtion

function get_dir_url_temp( $recurso ){
    global $dataroot;

    $path
    "C:\\Users\\Usuario\\Documents\\Chauchas\\Tesis_Berru\\archivos\\$recurso-
    >curso\\$recurso->item\\$recurso->sud_id\\";
    if(!file_exists( $path. $recurso->name_archivo )){

        if(!mkdir( $path , 0700, true)){
            return null;
        }
        $ruta_ini = "$dataroot\\$recurso->curso\\$recurso->name_archivo";
        if(!copy( $ruta_ini , "$path\\$recurso->name_archivo")){

            return null;
        }
    }
}

```

```

        return "$ruta\\$recurso->name_archivo";

    }else{
        return $path;
    }

} //get_dir_url_temp

function anota_recurso( $recurso ){

    try{
        $client = new SoapClient("http://localhost:8084/appanotador/WSanotador?wsdl");
        $res = $client->Procesar_reas(array( 'datos'=>json_encode( $recurso ) ));
        $va = $res->return;
        if( $res->return == "ok"){
            registra_evento( $recurso, $res->return, '1');
        }else{
            registra_evento( $recurso, $res->return, '0');
        }
    }

    }catch (SoapFault $exception) {

        $va = $exception;
        registra_evento( $recurso, $exception, '0');
    }
    return $va;
} //end anota_recurso

function registra_evento($recurso, $msg, $estado){ //Registra en la tabla
rea_annotado el recurso para no volverlo anotar

    $dato = new stdClass();
    //$dato->id = null;
    $dato->item = $recurso->item;

```

```

$dato->sud_id = $recurso->sud_id;
$dato->tipo = $recurso->tipo;
$dato->curso = $recurso->curso;
$dato->fecha = $recurso->fecha;
$dato->anotado = $estado;
$dato->obs = $msg;

$id = insert_record("rea_annotado", $dato, true);
} //end registra_evento

/*
 * Verifica si el archivo o url ha sido anotado
 * Se verifica todos los archivos o urls xq el usuario puede asociar
 * mas de estos recursos cuando modifica el rea
 */
function esta_annotado( $item, $sud_id){

    $esta = false;

    // $aux = get_records('rea_annotado','anotado','1','item', $item, 'sud_id', $sud_id);
    $aux = get_records_sql("select * from mdl_rea_annotado where item = $item and
sud_id = $sud_id and anotado = 1 ");
    if( $aux ){
        foreach($aux as $au){

            if( $au->id != null){
                $esta = true;
            }

        }

    }

    return $esta;

} //end check_item
function get_filio( $items ){

```

```

        $stable = "";
        if( $items ){
            foreach($items as $item){
                $stable .= '<tr>
                    <td>' . $item->item . '</td>
                    <td>' . $item->name_archivo . '</td>
                    <td>' . $item->formato . '</td>
                </tr>';
            }
        }

        return $stable;
    } //end funtion
?>

```

Al final se tiene el código del bloque en el cual se presenta las tareas pendientes y cada documento recuperado en el archivo `block_recomendador` de la carpeta del bloque:

```

<?PHP // $Id: block_messages.php,v 1.13.4.4 2008/03/03 11:41:03 moodler Exp $
class block_messages extends block_base {
    function init() {
        $this->title = "Recomendador";
        $this->version = 2007101509;
    }

    function get_content() {
        global $USER, $CFG,$COURSE;

        if (!$CFG->messaging) {
            return "";
        }

        if ($this->content !== NULL) {
            return $this->content;
        }

        if ($COURSE->id == $this->instance->pageid) {
            $course = $COURSE;

```



```

    } else {
        $course = get_record('course', 'id', $this->instance->pageid);
    }
    $this->content = new stdClass;
    $this->content->text = "";
    $this->content->footer = "";

    if (empty($this->instance) or empty($USER->id) or isguest() or empty($CFG->
    >messaging)) {
        return $this->content;
    }

    // $this->content->footer = '<a href=".'.$CFG->wwwroot.'/message/index.php"
    onclick="this.target=\'message\'; return openpopup(\'/message/index.php\', \'message\',
    \'menubar=0,location=0,scrollbars,status,resizable,width=400,height=500\',
    0);">'.get_string('messages', 'message').'/>...';

    $this->content->text .= '<link type="text/css" href=".'.$CFG->
    >wwwroot.'/blocks/messages/css/redmond/jquery-ui-1.8.16.custom.css"
    rel="stylesheet" />';
    $this->content->text .= '<script type="text/javascript" src=".'.$CFG->
    >wwwroot.'/blocks/messages/js/jquery-1.6.2.min.js"></script>';
    $this->content->text .= '<script type="text/javascript" src=".'.$CFG->
    >wwwroot.'/blocks/messages/js/jquery-ui-1.8.16.custom.min.js"></script>';
    $this->content->text .= '<script type="text/javascript" src=".'.$CFG->
    >wwwroot.'/blocks/messages/js/acciones.js"></script>';
    $this->content->text .= '<style>
        .ui-progressbar {
            position: relative;
        }
        .progress-label {
            position: absolute;
            left: 50%;
            top: 4px;
            font-weight: bold;
            text-shadow: 1px 1px 0 #fff;
        }
    </style>';

    $this->content->text .= '<input type="hidden" value=".'.$CFG->wwwroot.'" id="cfg"
    />
        <input type="hidden" value=".'.$COURSE->id.'" id="cursoid" />
        <input type="hidden" value=".'.$USER->id.'" id="userid" />

        <div class="tabs"><i>Estimados estudiantes estas son las
    recomendaciones que tiene para sus actividades pendientes:</i><br><h1
    class="titulos">Tareas</h1><br><div class="tarea">';
    /* $this->content->text .= '<div class="ui-state-highlight ui-corner-all"
    style="margin-top: 20px; padding: 0 .7em;">

```

```

        <p><span class="ui-icon ui-icon-clock" style="float: left;
margin-right: .3em;"></span>
        <strong>Procesando Reas</strong></p></div><br>'; */

    $this->content->text .= '<div id="progressbar"><div class="progress-
label">Procesando...</div></div>';

    $this->content->text .= '</div><br><h1 class="titulos">Foros</h1><br><div
class="foro"></div> </div> <br>';

    return $this->content;
}

} //end class

?>

```

Anexo 6

Manual del programador

El objetivo de este manual es dar a conocer al lector el funcionamiento del bloque de recomendaciones, los archivos que han sido necesarios para su construcción y como ha sido implementado en el EVA.

El proyecto se ha creado en net Beans con un enfoque basado en servicios web, por lo que hay que utilizar algunos métodos de la plataforma kim. A continuación se describen algunas carpetas que son parte del proyecto, luego se menciona los archivos y su función en la aplicación. Al final de este manual se explica cómo integrarlo al EVA.

Carpetas importantes del proyecto

Las carpetas que son necesarias para el funcionamiento del bloque recomendador son las siguientes:

- Appagente_recomendador
- Appanotador
- kim-platform-3.7
- recomendador
- Nusoap

Appagente_recomendador: esta carpeta posee todos los archivos referentes al servicio web agente_recomendador y está en la ubicación en donde net Beans guarda todos los proyectos.

Appanotador: esta carpeta es del proyecto del otro servicio web anotador, que se encarga de hacer las anotaciones semánticas de las tareas y recursos educativos abiertos, esta en la misma ubicación donde net Beans guarda los proyectos

kim-platform-3.7: esta carpeta se la puede descargar desde la pagina de la plataforma kim, esta posee todas las herramientas de kim archivos de instalación y configuración.

Recomendador: es la carpeta en el que se encuentran los diferentes archivos del bloque que se instalan en el entorno.

Nusoap: esta carpeta contiene los archivos de configuración e instalación de la librería nusoap para establecer comunicación entre clientes y servicios web

Requisitos Previos

Para que el bloque de recomendación funcione se debe considerar los siguientes requisitos:

- Tener instalado Xampp, debido a que interactúa con algunas tablas del EVA
- Tener instalado java el development kit en su versión 1.6 y el runtime la más actual que exista hasta la fecha.
- Tener instalado kim platform
- Tener instalado el EVA
- Tener instalado net Beans ide con el servidor Tomcat
- Que no estén ocupados los puertos 1099, 8084, 80, 3606

Archivos importantes

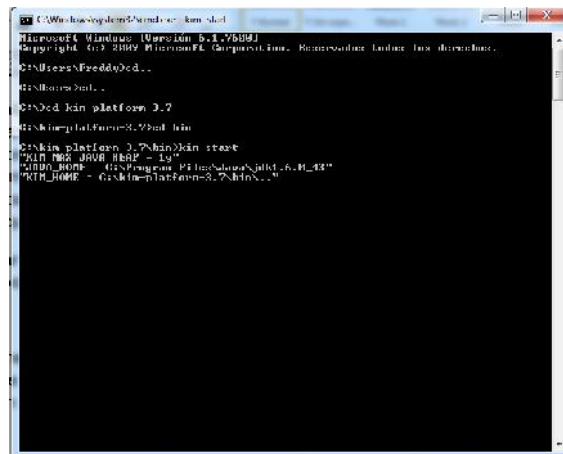
Por el enfoque del bloque de recomendación basado en servicios web se tiene que el archivo más importante es el del cliente que consume estos servicios siendo este archivo crucial para el funcionamiento del mismo.

El archivo se encuentra dentro de la carpeta del bloque recomendador y se denomina process.php en este archivo se encuentra el código para comunicarse y consumir los servicios web además se encuentra incluida la librería nusoap.

Instalación del bloque recomendador

Para instalar el bloque, en primer lugar tenemos que instalar la plataforma kim, para esto copiamos la carpeta en el disco de la pc, luego configuramos las variables de entorno como también el archivo config de la carpeta de bin ubicada en la carpeta de la plataforma kim, en este archivo se selecciona donde está ubicado el java en nuestra máquina para que kim lo pueda utilizar, además se puede configurar la cantidad de memoria que utilizara la plataforma.

Para iniciar kim hay que ir a la consola de comando cambiar al directorio de la carpeta de kim y poner el comando kim start lo cual dara inicio a la plataforma.

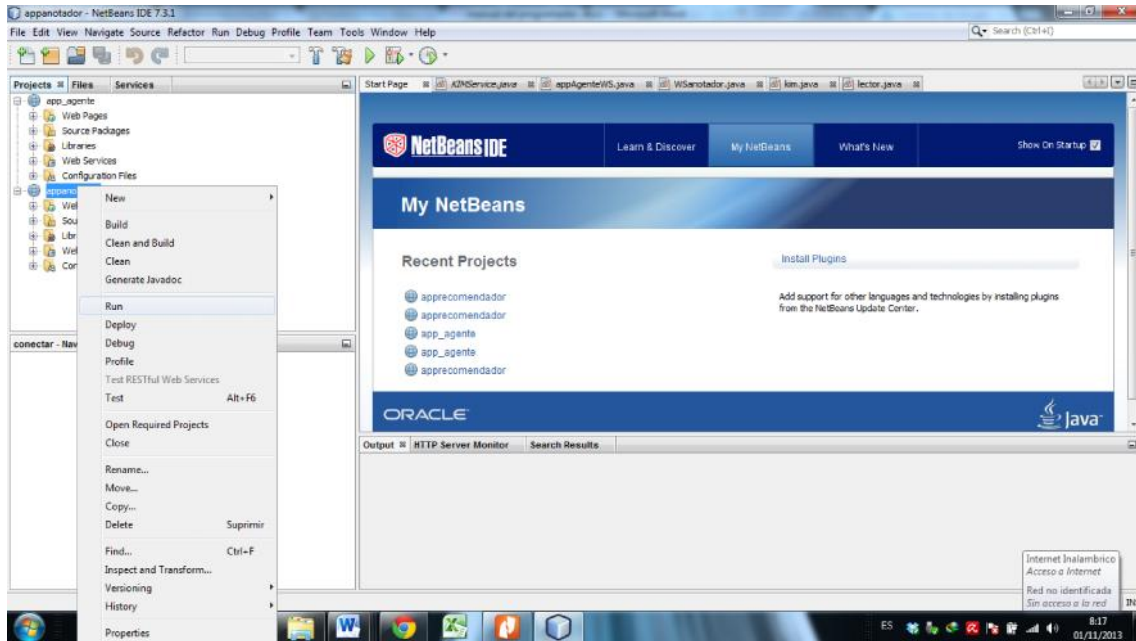


```
C:\Windows\system32\cmd.exe - kim start
Microsoft Windows [Versi3n 5.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

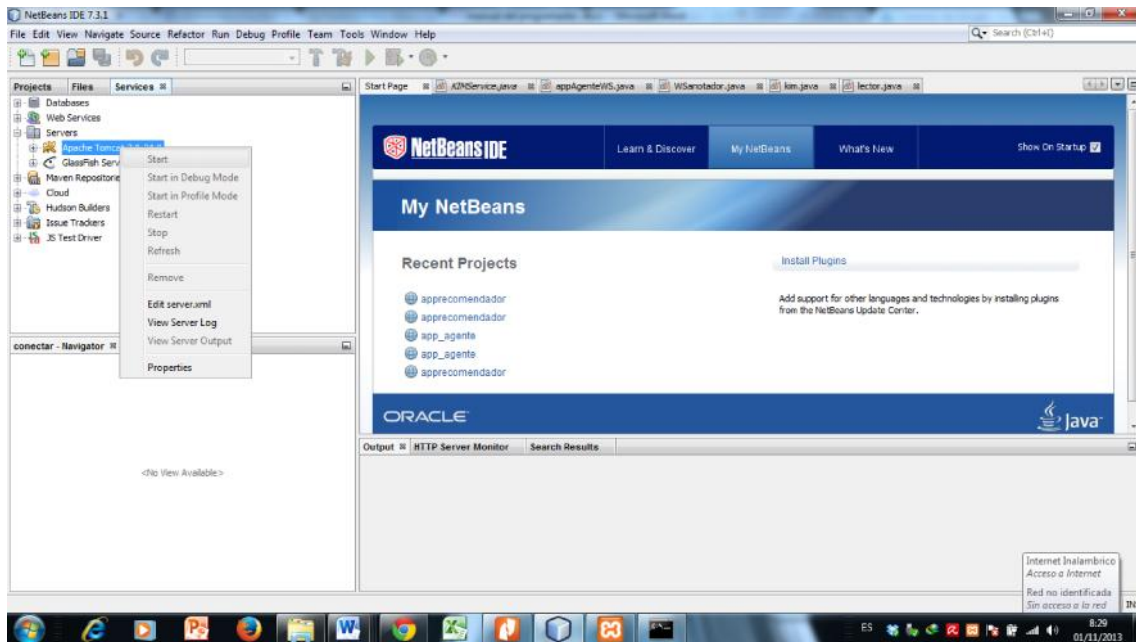
C:\Users\Freddy>cd ..
C:\Users>cd ..
C:\>cd kim platform 3.7
C:\kim platform 3.7>cd bin
C:\kim platform 3.7\bin>kim start
"JDK_HOME" = "C:\Program Files\Java\jdk1.6.0_45"
"JRE_HOME" = "C:\kim platform 3.7\bin\.."
```

Adicionalmente para observar los repositorios de kim se puede copiar el archivo kim.war de la carpeta de la plataforma a la carpeta del servidor Tomcat y correr este servicio en el puerto 8084

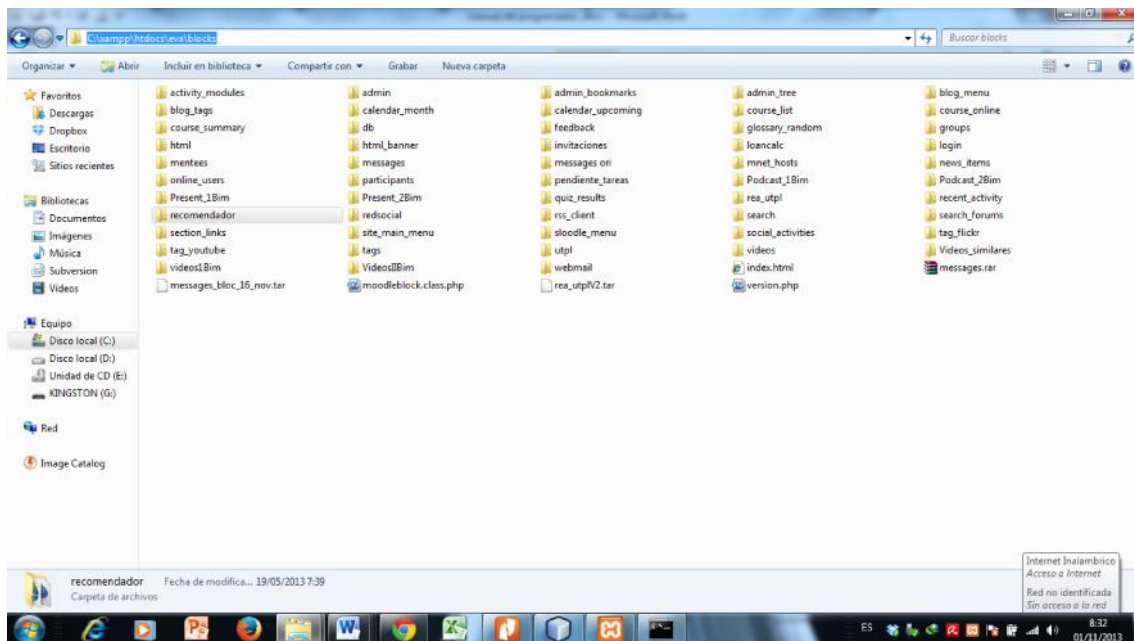
Una vez instalada e iniciada la plataforma kim se tiene que compilar los dos servicios web de proyecto.



Al final se tiene que iniciar el servicio Tomcat.



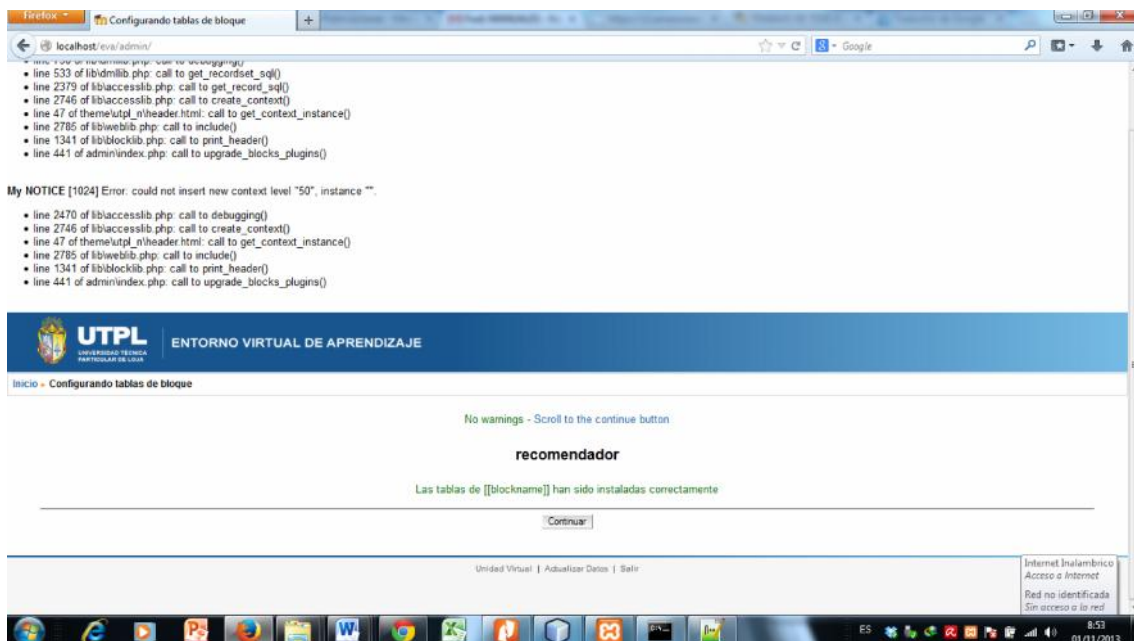
Cuando tenemos todo listo, vamos a la instalación local del EVA e instalamos el bloque recomendador copiando la carpeta recomendador en la carpeta blocks del EVA



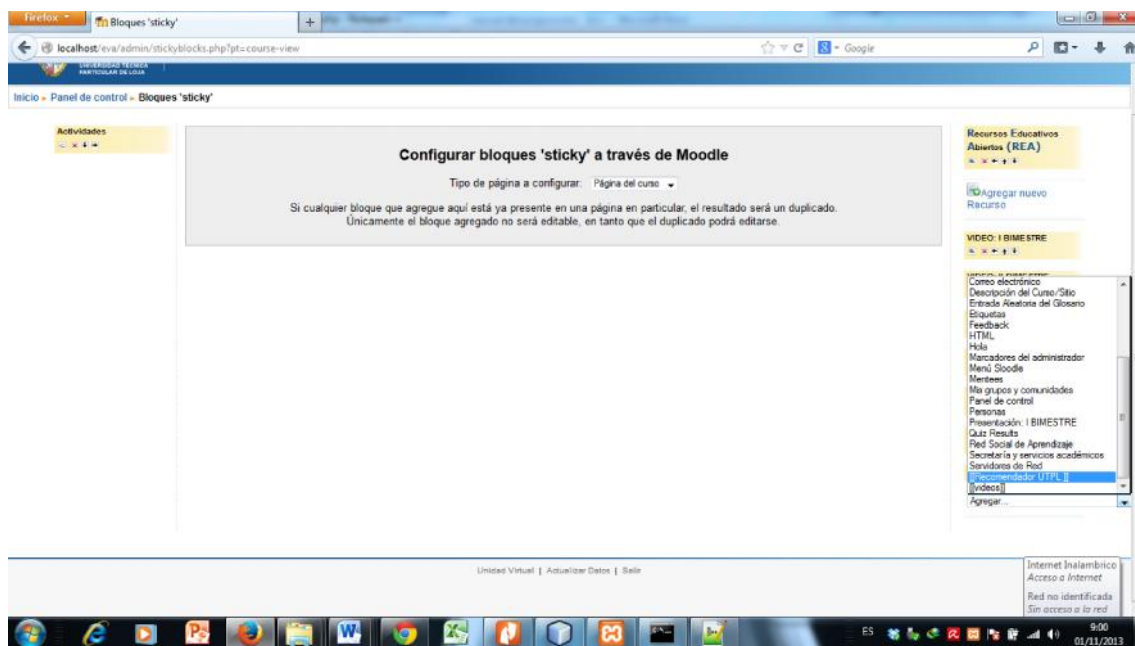
La carpeta recomendador tiene los siguientes archivos

- `block_recomendador.php`: es el bloque de recomendación el cual presenta las actividades pendientes y los recursos recomendados a esas actividades.
- `process.php`: este archivo enlaza el entorno con los servicios web y lo consume a través de sus peticiones.
- Carpetas de estilos: están carpetas se han incluido en el bloque para darle mejor presentación en el entorno.

Una vez que se ha copiado la carpeta del bloque vamos al admin del eva para instalar el bloque en el entorno



Para finalizar con la instalación se debe de activar la edición en el EVA e ir a la opción bloques stycks del panel de administración y seleccionar página de curso para que el bloque se muestre en cada página de curso.



Adicionalmente se ha creado una tabla en el entorno la cual se denomina re_a notado y la codificación se encuentra en el archivo process.php

Al final el bloque quedara instalado y aparecerá en cada curso del entorno.

Agente Recomendador para el soporte de actividades del estudiante en el EVA de la UTPL

Berrú Ramírez Freddy Eduardo
Universidad Técnica Particular de Loja
UTPL
Loja, Ecuador
feberru@utpl.edu.ec

Valdiviezo Díaz Prisila Marisela
Universidad Técnica Particular de Loja
UTPL
Loja, Ecuador
pmvaldiviezo@utpl.edu.ec

Resumen— El presente trabajo se lleva a cabo en el entorno virtual de aprendizaje (EVA) de la UTPL, en donde se ha visto la necesidad de dar soporte a los estudiantes en sus actividades de aprendizaje. Para este propósito se requiere del diseño y la construcción de un agente, que emita recomendaciones de recursos relacionados a las actividades que el estudiante tiene que realizar.

Se analiza el funcionamiento del EVA y el bloque de recursos educativos abiertos REA implementado en el mismo, se han identificado las actividades pendientes del estudiante, información que el agente utiliza y mediante el uso de la técnica de recomendación basada en atributos procesa esta información y da como resultado las recomendaciones al estudiante.

Para integrar esta técnica al agente recomendador se ha utilizado la herramienta web semántica kim, la cual ayuda en el proceso de anotaciones semánticas de los atributos y recuperación de los documentos que van a ser ofertados como recomendación.

En la parte final se ha acoplado el agente recomendador en el entorno mediante la utilización de servicios web y un cliente que consume estos servicios desde el EVA.

Keywords—LMS; Moodle; KIM; IRIS;STI;SR

INTRODUCCIÓN

En la UTPL se viene trabajando con un entorno virtual de aprendizaje EVA, el cual ha potenciado el proceso de aprendizaje de los estudiantes. Todos los cursos se llevan a cabo en el entorno y se desarrollan en cada periodo, una parte importante de estos cursos en línea son las actividades de aprendizaje ejecutadas en ellos, ya que estas actividades no solo contribuyen a la culminación y aprobación de los cursos sino también a otorgar conocimiento a los estudiantes, de ahí la importancia de dar soporte a estas actividades.

En la actualidad también se tiene implementado en la UTPL un bloque en el EVA, para el manejo de recursos educativos abiertos (REA), en el cual el profesor sube sus recursos para ayudar al estudiante.

Estas herramientas generan sobrecarga de información, por la gran cantidad de recursos que se suben en cada curso y en cada periodo universitario.

Esto hace que el estudiante no pueda elegir los recursos adecuados para resolver sus actividades en un determinado curso, ya sea porque a veces no tiene relación o se encuentran distintos recursos que pueden ayudarlo pero no distingue cual es el mejor.

Con el avance de las tecnologías del e-learning estos problemas aumentaran debido a que la información crecerá exponencialmente en estos entornos, por lo que se hace necesario la implementación de sistemas recomendadores los cuales según el trabajo de (Wang & Wu, Application of context-aware and personalized recommendation to implement an adaptive ubiquitous learning system, 2001) aportan con asistencia personalizada y en tiempo real, al proporcionar recursos en base a los requerimientos del estudiante en un entorno educativo y así puede ejecutar sus actividades de aprendizaje.

Para el diseño de un sistema recomendador se tiene que enfocar en las metas que tiene dicho sistema (Setten, 2005), y según estas metas profundizar en lo que se quiere realizar, para el proyecto la meta principal es dar soporte al estudiante en sus actividades de aprendizaje.

Cuando se tiene establecido el objetivo para el cual es creado un sistema recomendador se tiene que pensar en que técnicas serán las adecuadas para su desarrollo, revisando la literatura de los sistemas recomendadores existen una gran variedad de técnicas descritas en (Burke, 2007), (Adomavicius, 2005), (Itmazi, 2005).

Tomando en cuenta que el agente tiene que trabajar en un campo educativo, hay que escoger las técnicas que mejor se integren a este campo, ya que no es lo mismo integrar un sistema recomendador en un campo comercial que en uno educativo.

En el trabajo de (Olga C Santos) se tiene presente que en un ambiente educativo es mejor la utilización de sistemas recomendadores basados en conocimiento, por lo cual se ha orientado el trabajo por esa técnica de recomendación.

La importancia que tiene este proyecto para la institución y para la sociedad en general es bastante significativa ya que es un paso inicial al uso de tecnologías de recomendación, a brindar una mejor adaptabilidad de los sistemas de gestión de aprendizaje (LMSs) a las necesidades que tiene el usuario, promover la investigación de sistemas de recomendación y tecnologías web semántica que en la actualidad están en auge y cada vez son más utilizadas por sus amplias capacidades.

Durante el desarrollo de esta investigación también han surgido problemas referentes a integrar el agente en el EVA, ya que las tecnologías que se han utilizado son muy diferentes por una parte el agente ha sido construido enteramente en java y el entorno utiliza en cambio el lenguaje php, se ha tenido que encontrar la forma de comunicar estas dos tecnologías.

Otro problema que se ha presentado ha sido el gasto computacional que implica la utilización de herramientas web semánticas integradas a la parte semántica del agente, esto ha derivado en utilizar un computador con mejores prestaciones, también se tiene el problema de poder incorporar nuevas ontologías que permitan ampliar el contexto sobre el cual trabaja el agente por cuestiones de licencia lo cual ha limitado el trabajo a utilizar ontologías propias de la plataforma KIM.

En el presente artículo se describe cada una de las partes que han sido necesarias para desarrollar el agente:

En un comienzo se ha realizado un estudio de lo que comprende el funcionamiento del EVA de la UTPL con un enfoque en las actividades que realiza el estudiante dentro del mismo, luego se analiza y describe la literatura de los sistemas recomendadores de que tratan y como es su funcionamiento, luego se revisa las diferentes técnicas de recomendación y si son adaptables al agente recomendador.

Al tener presente que el agente se construye para un entorno educativo se ha observado un modelo presente en (Olga C Santos) el cual posee los elementos para recomendaciones en estos sistemas de ahí se han acoplado algunos elementos que tendrá el agente. Al revisar toda esta documentación se ha establecido la identificación de las tareas pendientes del estudiante ya que esta información es manejada por el agente, se ha hecho necesaria por la técnica a utilizar emplear anotaciones semánticas para estructurar de mejor manera esta información.

En la parte final se realiza un esquema del agente recomendador el cual a su vez permite el diseño de la arquitectura para describir después su construcción y los elementos necesarios para integrarlo al EVA. Una vez construido se han realizado las respectivas pruebas tanto de precisión como de usabilidad para sacar las conclusiones del proyecto.

ESTRUCTURA DEL EVA DE LA UTPL

En la Universidad Técnica Particular de Loja se tiene implementado un entorno virtual de aprendizaje (EVA), en donde se desarrollan los diferentes cursos tanto en modalidad presencial como en la modalidad abierta.

De las diferentes plataformas que existen para crear entornos virtuales en la universidad se adoptado Moodle con una serie de adaptaciones ya que se lo acoplado a las necesidades de la universidad, pero cabe recalcar que su estructura interna es la misma.

Moodle por sus siglas en ingles quiere decir entorno virtual de aprendizaje orientado a objetos y modular, este sistema es open source se distribuye con la licencia GNU sirve para la creación y desarrollo de cursos a través de internet.

Posee características importantes para el desarrollo de actividades de aprendizaje de un curso ayudando al desarrollo de conocimiento en estudiantes, hoy en día la mayoría de instituciones tiene implementado este LMS ya que facilita el proceso de enseñanza – aprendizaje.

En la búsqueda por un esquema que permita describir la estructura del EVA, se ha encontrado una arquitectura en (Itmazi, 2005) en donde se muestra la arquitectura general de un sitio construido con Moodle vista desde la perspectiva de un profesor. Se ha usado esta grafica para elaborar una arquitectura pero del entorno de la universidad.

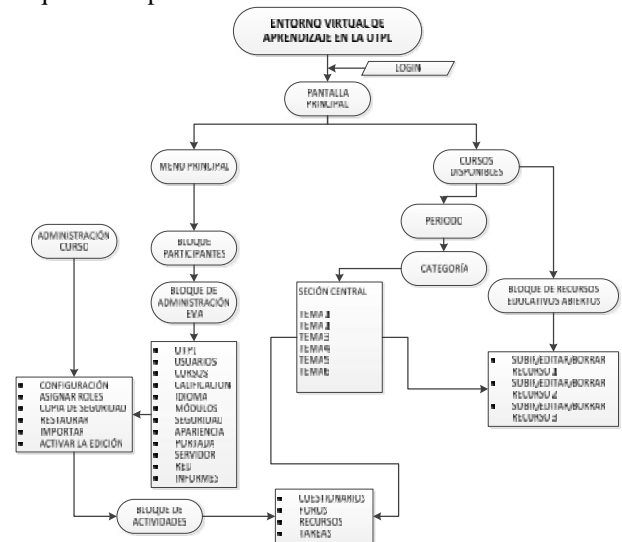


Grafico 1. Arquitectura General del EVA de la UTPL

En el grafico se puede observar como es el funcionamiento del sitio, de acuerdo a la universidad cada curso tiene una categoría que vendrían a ser las carreras, además de esto poseen un periodo al cual se rigen.

Se describe que en cada curso la parte central tiene los diferentes temas tratados en él y en el cual se pueden agregar actividades de aprendizaje, hay varias opciones de actividades pero en el entorno de la UTPL se tiene: cuestionarios, foros, recursos y tareas.

Cabe mencionar también que en el entorno se ha adaptado un bloque de recursos educativos abiertos denominado REAS.

A continuación se presenta una imagen de un curso real de la universidad.



Gráfico 35 Pantalla de un curso de la UTPL

En la gráfica anterior se tiene tres partes principales de un curso en el entorno.

Partes Laterales: a esta parte se la denomina bloques y sirven para mostrar información o crear alguna funcionalidad a ser utilizada por el estudiante o el profesor, en la imagen se muestra el bloque de actividades y el bloque REA.

Parte central: en esta parte se discuten los temas del curso y un profesor puede crear sus actividades de aprendizaje que el estudiante tiene que desarrollar.

De las actividades más utilizadas en el entorno se han seleccionado las siguientes con las que trabajará el agente:

- Tareas
- Foros

Tareas: este módulo de Moodle permite al profesor recoger tareas de todos sus estudiantes dentro del entorno, un estudiante no puede ver las tareas de otros a menos que se haya configurada en grupo, se pueden enviar en diferentes formatos.

Foros: este módulo permite la discusión de un tema en específico del curso, existen diferentes tipos de foros pero el más utilizado es el foro abierto donde el educador pone un tema y los estudiantes pueden opinar acerca del mismo y subir sus respuestas.

Una vez definidas las actividades que tienen mejor acogida en el EVA se tiene que continuar identificando que tareas realiza o no el estudiante y dar una solución para las actividades que no se realizan.

SISTEMAS RECOMENDADORES

Los sistemas recomendadores nacen de lo que se denomina algoritmos de predicción que hicieron uso de muchas tecnologías como son: ciencia cognitiva, recuperadores de información, teoría de la aproximación, teoría de pronósticos, ciencia administrativa y de consumo.

Según (Adomavicius, 2005), los sistemas de recomendación emergen como una ciencia independiente en los años 90, donde centran su atención en los problemas de recomendación que explícitamente se basan en una estructura de ratings (puntuaciones).

Este mismo trabajo define que un problema de recomendación común se formula dando ratings a diferentes ítems que no hayan sido vistos por determinados usuarios,

de ahí se escoge los mejores puntuados y se realiza la recomendación, esto en un escenario básico de recomendación.

Al describir formalmente un sistema de recomendación en este mismo trabajo se define: C como el conjunto de usuarios y S como el conjunto de todos los ítems a recomendar, cada uno de estos conjuntos puede ser inmenso, por ultimo tenemos una función de utilidad descrita por u que determina la usabilidad del ítem al usuario. $u: C \times S \rightarrow R$, R es un conjunto ordenado de enteros o reales no negativos que están en cierto rango)

Luego, se deduce que para cada usuario perteneciente a C , se escoge un ítem perteneciente a S que maximice la utilidad en el usuario.

$$\forall c \in C, \quad s'_c = \arg \max_{s \in S} u(c, s). \quad (1)$$

En un sistema recomendador la utilidad está definida por unas puntuaciones (ratings) que son la medida de preferencia del usuario por ese ítem.

El principal problema de un sistema recomendador radica en que la utilidad no se encuentra en todo el espacio $C \times S$, sino en un subconjunto de este, debido a que el usuario realiza puntuaciones de ciertos ítems en otros este valor es desconocido.

Por esta razón hay que extrapolar la función de utilidad a todo el espacio $C \times S$.

La extrapolación de ratings conocidos a desconocidos es realizada por medio de: 1) Especificar Heurísticas: que definen la función de utilidad y la validación empírica de su funcionamiento y 2) estimar la función de utilidad que optimiza cierto criterio de funcionalidad.

Una vez que los ratings desconocidos han sido estimados, recomendaciones actuales al usuario son realizadas seleccionando los ratings más altos de entre todos los ratings realizados por el usuario de acuerdo a la ecuación (1)

Los nuevos ratings de los ítems que no han sido puntuados (rated) pueden ser estimados de diferentes formas usando métodos como aprendizaje de máquina, teorías de aproximación y varias heurísticas.

En la investigación llevada a cabo en el presente artículo el conjunto C de usuarios está determinado por los estudiantes y el conjunto S son los ítems a recomendar es decir los recursos que dan soporte a sus actividades en el EVA.

La estimación de ratings es ejecutado de una manera diferente ya que el estudiante no tiene que puntuar (rated) los recursos, esta estimación se realiza de forma semántica y las puntuaciones se dan de acuerdo a la relación semántica del recurso con la actividad que el estudiante tiene que realizar en el EVA.

Según esta investigación (Adomavicius, 2005), los sistemas recomendadores se clasifican enfocados en estos métodos de estimación de ratings, según esto establecen sus técnicas de recomendación, en la construcción del agente

recomendador se tiene que analizar la técnica de recomendación que mejor se acople a la necesidad de soporte a las actividades del estudiante y que no utilice puntuaciones del estudiante. Más adelante se describe las diferentes técnicas de recomendación con sus ventajas y desventajas.

I. MODELO DE RECOMENDACIÓN EN UN ENTORNO EDUCATIVO

La mayoría de investigaciones realizadas sobre sistemas recomendadores se han centrado en entornos comerciales donde las preferencias se acogen mediante encuestas en el mismo sistema, para luego utilizar estas puntuaciones y recomendar artículos a nuevos usuarios, un ejemplo de este sistema ha sido la página Amazon.

Las investigaciones dirigidas a entornos educativos son recientes, tenemos varios trabajos pero no son abundantes, debido a que trabajar en un dominio educativo tiene sesgadas diferencias que trabajar en un dominio comercial.

Según (Olga C Santos), estas diferencias son las siguientes:

- **Predisposición del Usuario:** está relacionado al interés que el usuario tiene en puntuar ítems en un entorno educativo, su nivel de interés es bajo en relación a cuando se encuentra en uno comercial.
- **Contexto Estructural:** en un entorno comercial es fácil recomendar ítem ya que se trata de música, video, u otros ítems de gusto popular. Mientras que en un entorno educativo la complejidad de los ítems aumenta debido a que se maneja conocimiento, actividades de aprendizaje en relación a los cursos de este entorno

Por esta razón en este mismo proyecto (Olga C Santos) se maneja un modelo para generar recomendaciones en un entorno educativo de los cuales se ha elegido los elementos que pueden integrarse al agente recomendador:

Categoría: se relaciona con el alcance del sistema recomendador, se tiene diferentes categorías que pueden ser configuradas acorde al contexto y los objetivos del sistema recomendador.

Técnica: la técnica es la parte esencial de un sistema recomendador ya que se utiliza para maximizar el valor de utilidad de un ítem al usuario, existen varias técnicas y clasificaciones pero se escogerán de entre las que sirven mejor en un entorno educativo.

Origen: el origen determina la fuente que originó la necesidad del sistema recomendador está ligado a la categoría y depende del contexto donde se utiliza el sistema recomendador, por lo que también es configurable a las necesidades del contexto.

Condiciones Aplicables: Tiene relación a las características del entorno para emitir o no una recomendación, en este trabajo las condiciones son de la plataforma, las recomendaciones se dan el momento que

el estudiante utiliza la plataforma y revisa sus actividades en la misma.

II. TÉCNICAS DE RECOMENDACIÓN

La técnica de recomendación es el método y los algoritmos utilizados internamente para generar las recomendaciones se basan en varias tecnologías de inteligencia artificial, procesamiento de lenguaje natural como sistemas de recuperación de información y otras.

Existen diferentes clasificaciones de estas técnicas, de acuerdo a los orígenes donde la clasificación se enfoca al método utilizado para la estimación de ratings según (Adomavicius, 2005) existen las siguientes técnicas:

Sistemas Basados en Contenido: el ítem es recomendado al usuario de acuerdo a sus gustos de ítems similares en el pasado.

Sistemas de Filtrado colaborativo: es uno de los más extendidos y se basa en que si el ítem tiene una alta puntuación, se verifica el perfil de usuario y se recomienda ítems puntuados por usuarios con perfiles similares que puntuaron ese ítem en el pasado.

Sistemas Híbridos: combina ambas técnicas de filtrado colaborativo y basada en contenido.

Esta es la clasificación de acuerdo a como se realizan las predicciones, pero hay otra clasificación que se basa en el algoritmo que utilizan para realizar la predicción esta clasificación es la siguiente:

Basada en Modelo: las basadas en modelo usan información de acuerdo al usuario y establecen un modelo con el cual generar las recomendaciones.

Basadas en memoria: esta técnica utiliza toda la información para generar la recomendación

Tabla 11 técnicas de recomendación basadas en modelo y memoria

Basadas en modelo	Basadas en Memoria
<ul style="list-style-type: none"> • opera una base de datos del usuario y establecen un modelo • Aprenden lentamente y dan las recomendaciones rápidamente • Necesitan un gran cuerpo de datos generalmente 10000 para poder ofertar recomendaciones. 	<ul style="list-style-type: none"> • opera sobre la base de datos entera del usuario • Aprenden rápidamente y dan recomendaciones lentamente. • Se utilizan en entornos dinámicos, donde los datos cambian continuamente

De acuerdo a esta clasificación hay que seleccionar la que mejor se acople a un entorno educativo, con la revisión de varias investigaciones orientadas al campo educativo como (Drachler, Hummel, & Koper, 2007), (Burke, 2007), (Itmazi, 2005), en donde se recomienda que para un entorno educativo son adecuadas las técnicas de recomendación basadas en memoria se realiza un análisis de las mismas y su adaptabilidad a este proyecto.

Se propone los siguientes tablas en donde se especifica las debilidades y potencialidades de cada una de las técnicas y además se concluye si es que es la adecuada a los requerimientos del agente recomendador

Tabla 12 Filtrado colaborativo basado en usuarios

Ventajas	Desventajas	Utilidad en Agente	Adaptable
<ul style="list-style-type: none"> No se realiza análisis de contenido. Es independiente del dominio. Calidad en recomendaciones Tiene un enfoque ascendente Serendipity: ofertar recomendaciones por casualidad 	<ul style="list-style-type: none"> Problema de Nuevo Usuario Problema de Nuevo Ítem Gustos Populares Escalabilidad Escases Problema de arranque en frío 	<p>Según el enfoque del agente esta técnica no sería muy aconsejable, por lo que es necesaria información explícita, es decir que los usuarios hayan calificado varios ítems para generar una recomendación.</p> <p>Problemas de nuevos usuarios o ítems, además el usuario tiene que atenerse a recomendaciones de gustos populares.</p> <p>Tal vez combinada con otras técnicas y en una fase inicial</p>	No

Tabla 13 Filtrado colaborativo basado en ítems

Ventajas	Desventajas	Utilidad en Agente	Adaptable
<ul style="list-style-type: none"> No se realiza análisis de contenido. Es independiente del dominio. Calidad en recomendaciones Tiene un enfoque ascendente Serendipity: ofertar recomendaciones por casualidad 	<ul style="list-style-type: none"> Problema de Nuevo Ítem Gustos Populares Escases Problema de arranque en frío 	<p>Al tener que calificar los ítems padece de los mismo problemas que el filtrado basado en usuario como es el de un nuevo ítem, atenerse a gustos populares y el problema de arranque en frío que significa si no hay suficientes datos calificados no es posible emitir recomendaciones</p>	No

Tabla 14 Técnica de filtrado colaborativo demográfico

Ventajas	Desventajas	Utilidad en Agente	Adaptable
<ul style="list-style-type: none"> No hay problema de arranque en frío Independiente del Dominio Casualidad 	<ul style="list-style-type: none"> Obtener información Información insuficiente Solo Gustos populares Obtener información (Metadata) Mantenimiento de ontología 	<p>Se tiene problemas al obtener información de cierta forma delicada del usuario, además el enfoque del agente no está relacionado con la información demográfica que pueda tener un estudiante.</p>	No

Tabla 15 Técnica de razonamiento basado en casos

Ventajas	Desventajas	Utilidad en Agente	Adaptable
<ul style="list-style-type: none"> No hay análisis de contenido Independiente 	<ul style="list-style-type: none"> Problema de nuevo usuario Superespecialización 	<p>Tiene el problema de arranque en</p>	No

<ul style="list-style-type: none"> del Dominio Mejor Calidad 	<ul style="list-style-type: none"> Escases Problema de arranque en frío 	<p>frío es decir que se necesita de bastante información calificada para realizar la recomendación</p>	
--	---	--	--

Tabla 16 Técnica basada en atributos

Ventajas	Desventajas	Utilidad en Agente	Adaptable
<ul style="list-style-type: none"> No hay problema de arranque en frío No hay problema de usuario ni ítem nuevo Sensible a cambio de preferencias Puede incluir características relacionadas con el ítem Puede mapear las necesidades del usuario con el ítem 	<ul style="list-style-type: none"> No aprende Solo trabaja con categorías Es requerido un modelamiento y mantenimiento de ontologías Superespecialización 	<p>Según el enfoque del agente esta es una de las técnicas más aconsejables de utilizar junto al subtipo de esta técnica como la basada en conocimiento puede prevenir algunos problemas que padece esta técnica</p>	SI

Al observar cada una de estas tablas se concluye que cada una de estas técnicas tiene sus ventajas y desventajas, por lo que generalmente se la utiliza una técnica de recomendación híbrida que combinen cada una de ellas y de esta manera minimizar los errores que tienen.

Otra opción es extender las capacidades que tiene cada una de estas técnicas y crear subtipos como se lo realiza en esta investigación donde interviene otra clasificación dada por (Burke, 2007), en su trabajo se menciona la técnica basada en conocimiento que es tomada como un subtipo de la técnica basada en atributo que es conveniente para el agente.

IDENTIFICACIÓN DE TAREAS PENDIENTES EN EL EVA

Para identificar esta información del EVA se ha requerido de un proceso detallado a continuación.

Para el proceso de identificación de tareas pendientes se ha requerido instalar el moodle de forma local y trabajar con un herramienta de administración de bases de datos, en el caso del proyecto se ha trabajado con TOAD, una vez instalado la plataforma moodle esta trabaja con una base de datos Mysql que se encarga de guardar todo lo que sucede en el entorno.

Se ha creído conveniente instalar Moodle porque su estructura es similar a la del EVA, solo que en este último se han realizado varias adaptaciones pero eso no afecta el proceso de identificar las tareas.

Se han creado dos cursos y se han inscrito varios estudiantes con el fin de monitorear sus acciones dentro de la plataforma, debido a que sería un trabajo tedioso verificar cambios en todas las tablas que son un total de 215, solo se

han escogido las que se relacionan a los estudiantes, curso y actividades.

Según el diagrama de entidad relación, las entidades que intervienen en esta interacción son: usuarios, cursos, tareas y foros.

Mediante esto han sido seleccionadas las tablas para ver los cambios en los campos y poder identificar las tareas pendientes.

Entonces se ha procedido a llenar con actividades los cursos como son: foros y tareas, luego se experimenta de la siguiente manera: con una parte de los estudiantes se han completado todas las actividades mientras que con la otra parte no se han completado.

Con la herramienta Toad se ha visto los cambios en los registros y se ha obtenido como resultado que las siguientes tablas han tenido cambios en sus registros:

- mdl_forum_discussions
- mdl_assignment_submissions

Ambas tablas pertenecen a las entidades de foros y tareas.

Analizando ya el EVA en una instalación local se ha visto que estas tablas tienen relación con la tabla de usuario donde está el estudiante, con esto se deduce que estas tablas registran la participación de los estudiantes tanto en foros como en tareas.

La siguiente imagen sacada de una reingeniería a la base de datos de un EVA instalado localmente comprueba lo antes dicho.

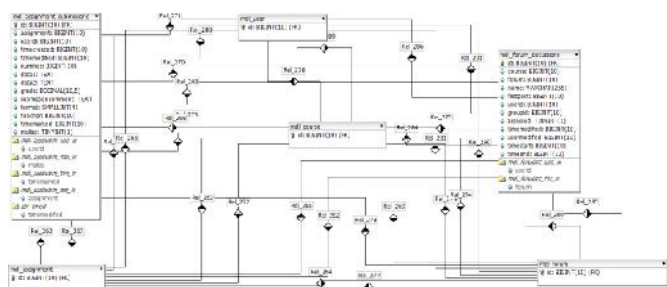


Gráfico 36 Modelo entidad relación del EVA

Se llega a la conclusión que si un estudiante no realiza sus tareas y foros, estas no se registran en las tablas mdl_forum_discussions y mdl_assignment_submissions y quedan identificadas como tareas pendientes de dicho estudiante.

Cuando se han identificado las tareas pendientes, ha sido necesaria la creación de un bloque en el EVA que muestre estas actividades al estudiante.

La siguiente imagen muestra este bloque en un curso de “Estructura de Datos” del EVA.

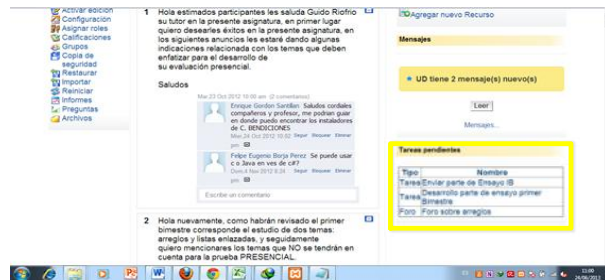


Gráfico 37 Bloque tareas pendientes en el EVA

El gráfico se muestra que el estudiante tiene dos tareas y un foro pendientes de realizar en la materia de Estructura de datos.

Sobre estas actividades se puede brindar soporte al estudiante a través de la recomendación de REAS de tal forma que contribuyan al desarrollo de las mismas, por esta razón se requiere de alguna técnica de recomendación que se base en estos atributos (actividades), además dar un enfoque semántico debido a que los recursos posiblemente recomendados tiene que tener relación a las actividades.

ANOTACIONES SEMANTICAS Y SUS HERRAMIENTAS

Una vez que se han identificado las tareas pendientes surge la necesidad de implementar tecnologías web semántica en el proyecto debido a dos razones: la una es porque un agente tiene que trabajar con información organizada que pueda ser procesada por una máquina y la otra es por la técnica seleccionada basada en atributos (tareas pendientes).

Para la realización de esta parte del agente se ha investigado las anotaciones semánticas.

De la investigación realizada hay diferentes herramientas de la web semántica que nos permiten hacer anotaciones semánticas y se ha tomado en consideración sus características como que sean de licencia libre y automáticas.

Además que se acoplen a la arquitectura del agente, siendo una de las elegidas la plataforma KIM, no solamente por realizar anotaciones semánticas sino por ser un conjunto completo de herramientas de administración de conocimiento, y que se acopla a la arquitectura de la forma correcta ya que también se puede realizar búsquedas semánticas, nombramiento de entidades, guardar documentos y recuperarlos mediante sus respectivas APIs. Además la documentación de Kim que se encuentra en su página web oficial, se puede obtener información de sus interfaces para acceder a las anotaciones y todo lo referente a su arquitectura.

5.1. Plataforma KIM

Según su documentación (2013) Kim es una plataforma de administración del conocimiento que permite realizar lo siguiente:

- Crear enlaces semánticos entre documentos, datos, modelos del dominio y linked data.

Esta arquitectura demuestra como el agente interactúa con la plataforma tanto del EVA como la de KIM para realizar la parte semántica.

El EVA le provee de la información de las tareas pendientes del estudiante y los REAS.

Por su parte el agente realiza la parte de anotación semántica y recomendación lo hace mediante la plataforma kim y dos servicios web.

En la parte de anotación el agente anota tanto las tareas pendientes y los REAS para guardarlos en el repositorio de kim, guarda tanto las entidades como los documentos, el servicio web encargado de esta parte se llama Anotador.

En la parte de recomendación interviene en servicio web agente recomendador, el cual una vez realizadas las anotaciones semánticas, se encarga de realizar una búsqueda semántica de las entidades en los REAS y los recupera para ofertarlos como recomendación en cada actividad del estudiante.

Esta arquitectura consta de los siguientes elementos:

Bloque de actividades pendientes

Se han identificado las actividades pendientes, por lo que se ha elaborado un bloque en donde se muestran estas actividades al estudiante, este bloque ha sido agregado para que forme parte de un bloque de recomendación general en donde también se muestran las recomendaciones por cada actividad pendiente.

APIs de la plataforma Kim

Para poder llevar a cabo las funciones de anotación semántica, recuperación de información y demás de la parte semántica del agente. Este tiene que acceder a varias de sus APIs por medio de RMI (remote method invocation) , por medio de sus dos servicios web.

Las siguientes son las APIs que tiene la plataforma kim y que se acceden a la mayoría de ellas:

- SemanticAnnotationAPI service
- DocumentRepositoryAPI service
- SemanticRepositoryAPI service
- EntityAPI service
- QueryAPI service

SemanticAnnotationAPI service: Esta API es muy importante y está integrada al agente recomendador ya que sirve para invocar procesos de anotación semántica. En una rutina estándar de población, la anotación debe ser llamada después de que el cuerpo del documento y las características son guardados en un documento tipo KIM. Se distinguen dos métodos principales:

DocumentRepositoryAPI service: se encarga de guardar todos los documentos anotados para ser recuperados en lo posterior con métodos como loadDocument o getDocument.

SemanticRepositoryAPIservice: se encarga de grabar la información semántica extraída como RDF. El servicio también permite navegar y modificar el repositorio.

EntityAPIService: provee de una abstracción sencilla en lo alto de la SemanticRepositoryAPI. Permite la carga de toda la información semántica de una entidad dada con una sola llamada.

QueryAPIService: mediante este servicio se pueden formalizar consultas complejas tanto al repositorio semántico como al repositorio de documentos. Combina funcionalidades de consulta del DocumentRepositoryAPI y del SemanticRepositoryAPI, se pueden realizar consultas semánticas y sintácticas. Las consultas deseadas son entradas a través de objetos tipo SemanticQuery y da como resultado una compleja estructura de objeto tipo SemanticQueryResult.

Servicio web Anotador

Para describir los servicios web se ha creado una tabla con sus datos de entrada y salida y sus respectivos métodos.

Tabla 17 servicio web anotador

Servicio Web Anotador		
Datos de entrada	Método	Salida
server (información del servidor)	Conectar	rmi Objeto tipo kim
datos (información de los recursos)	Procesar Reas	msg, este método envía los recursos al método anota
rmi, ítem, contenido y modo información de las tareas y recursos	Anota	anotado, este método se encarga de utilizar las funciones de kim para anotar y guardar documentos.
Tarea	Recomendar	Este método se encarga de enviar las tareas al método anota_tarea del kim

Este servicio actúa como un pequeño agente, tiene la funcionalidad de realizar las anotaciones semánticas de las tareas pendientes y de los recursos educativos abiertos y guardar tanto entidades como documentos en el repositorio de kim.

Servicio Web Agente_recomendador

Tabla 18 Servicio web agente_recomendador

Servicio Web Agente_recomendador		
Datos de entrada	Método	Salida
server (información del servidor)	Conectar	rmi Objeto tipo kim
actividad (información de las actividades en el EVA)	Recomendar	resp, en este método se realiza la búsqueda semántica de documentos relacionados con las entidades anotadas de las tareas y las recupera en un arreglo de datos. Para transformar los objetos de un tipo a otro en php se utiliza json Si no se obtiene resultado se

		envía a consulta sintáctica
anotations	Consulta Sintáctica	resp. en este método se realiza una consulta sintáctica y se devuelve un arreglo con los documentos recuperados

En si este servicio cumple con la funcionalidad para la cual fue creado, por esta razón se lo denominó agente. Es el encargado de obtener las actividades de donde se obtuvo las anotaciones semánticas del entorno y realizar una búsqueda en primera instancia semántica de documentos en donde se nombren estas anotaciones, si es que no se obtiene un resultado pasa a realizar una búsqueda sintáctica.

Como resultado este servicio web envía los documentos recuperados y relacionados con cada actividad pendiente del estudiante.

Estos son todos los elementos de la arquitectura del agente que se han integrado en la plataforma de aprendizaje de la UTPL.

INTEGRACIÓN EN EL EVA DEL AGENTE

Una vez que se han realizado cada una de las partes de la arquitectura del agente recomendador se ha integrado en el EVA de pruebas de la UTPL.

Esta tarea no ha resultado muy fácil ya que hay que comunicar dos tecnologías diferentes: por una parte está el EVA que basa su desarrollo en PHP y por otra parte está el agente que basa su desarrollo en Java.

Para poder realizar esta comunicación, se ha utilizado el protocolo de comunicación SOAP, protocolo de acceso simple a objetos, se lo ha logrado al instalar una herramienta NUSOAP en PHP, la cual permite enviar mensajes a través del protocolo.

Algo adicional que se ha creado ha sido una tabla en el EVA donde se guarda los recursos ya anotados, esto con el fin de que no se repitan las recomendaciones a los estudiantes.

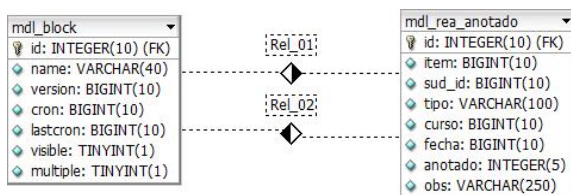


Gráfico 39 Tabla donde se guardan los reas anotados

Entonces para consumir los dos servicios web se ha creado un cliente en el entorno, el cual se encuentra en la carpeta del bloque de recomendación el archivo se denomina process.php y se conecta a los servicios web a través de estas líneas de código:

```
$client = new
SoapClient("http://localhost:8084/appanotador/WSanotador?wsdl");
```

```
$client = new
SoapClient("http://localhost:8084/app_agente/appAgenteWS?wsdl");
```

Al integrar el agente al EVA se tiene la siguiente imagen de bloque de recomendación:

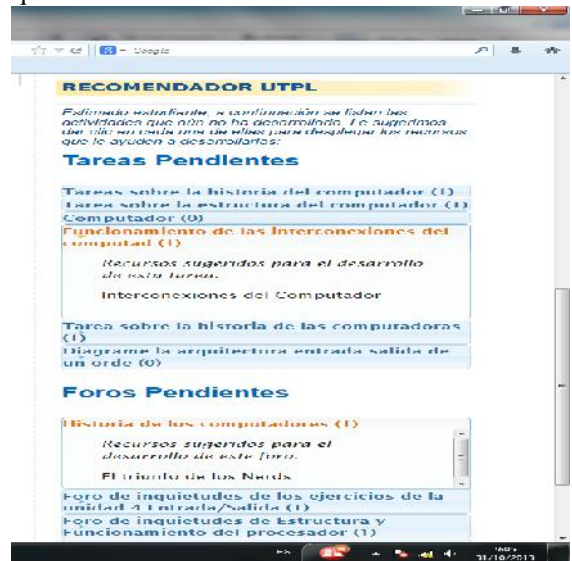


Gráfico 40 Agente recomendador integrado al EVA

En el gráfico se describe las actividades pendientes que tiene un estudiante en la materia de “Arquitectura de Computadores”, con un mensaje introductorio en donde se sugiere al estudiante realizar un clic sobre cada actividad tanto foros como tareas con la finalidad que se le desplieguen los recursos recuperados para cada actividad. Como se puede observar en la imagen hay una tarea y un foro que poseen un recurso respectivamente para su desarrollo.

Luego de la integración se han realizado algunas pruebas con el agente, para verificar su precisión a la hora de recuperar documentos relevantes a las actividades de los estudiantes.

Debido a que el agente utiliza una técnica basada en atributos, por lo que se menciona en (Setten, 2005), este tipo de técnicas utiliza sistemas de recuperación de información se ha centrado la atención en las pruebas más utilizadas en estos sistemas que según la investigación de (Lidó, 2002), las más utilizadas son las de precisión.

PRUEBAS DEL AGENTE RECOMENDADOR.

Pruebas de Precisión

Las pruebas de precisión que son una de las más utilizadas en sistemas de recuperación de información se han aplicado a este agente para verificar cuan preciso es al recomendar documentos adecuados a las actividades del estudiante. El valor de precisión viene dado por la siguiente ecuación:

$$\text{Precisión} = \frac{\text{Documentos Relevantes Recuperados}}{\text{Documentos Recuperados}} \quad (2)$$

En el EVA se han seleccionado tres cursos para realizar esta prueba, adicionalmente se han creado tareas y foros propuestos por el docente así como también se han creado varios REAS en cada curso.

Para un mejor funcionamiento del agente se ha tenido que describir cada actividad de una manera muy específica de tal forma que se puedan anotar entidades de ellas.

La siguiente tabla muestra los cursos y los documentos que se han recuperado.

Curso	Actividades	Documentos Relevantes Recuperados	Documentos Recuperados
Arquitectura de Computadores	10	7	8
Planeación estratégica	10	3	5
Fundamentos de Informática	10	14	16

Cabe recalcar que se considera documento relevante al recurso recomendado en el agente que tiene relación con la actividad pendiente y que puede ayudar a resolverla.

En algunos cursos se tiene más documentos recuperados, esto se debe a que para cada actividad el agente recuperó más de un documento, tal es el caso de “Fundamentos Informáticos” en donde se recuperan 16 documentos.

Resultados de las pruebas de Precisión

En las pruebas de precisión se han obtenido diferentes valores en cada una de las materias seleccionadas para estas pruebas, según la ecuación que describe esta prueba, estos son los resultados:

- Arquitectura de computadores: se han obtenido 7 documentos relevantes de 8 documentos recuperados dando un porcentaje del 88% de precisión
- Planeación estratégica: en este curso se obtuvo 3 de 5, dando un porcentaje del 60%
- Fundamentos Informáticos: en este curso se han recuperado 14 documentos relevantes de 16 recuperados, dando como resultado un porcentaje del 87%.

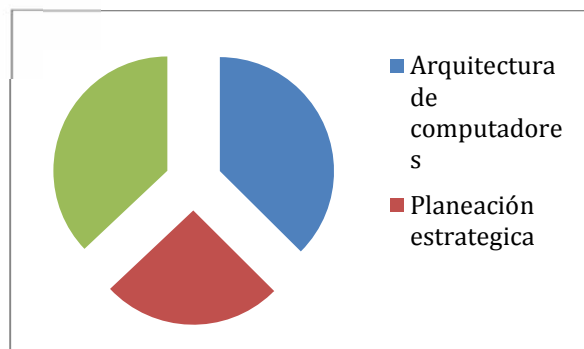


Gráfico 41 precisión en cada materia

Cuando de todos los documentos recuperados por el agente se tiene que todos son relevantes a la actividad, se dice que el agente es completamente preciso, pero esto no se puede lograr debido a problemas de falsos positivos y ruido documental, que es cuando se recupera documentos que parecieran importantes pero llegado a la realidad no lo son, también se da el silencio informativo donde un documento que es relevante no se lo recomienda.

De los resultados de la diferentes materias se ha realizado un promedio para establecerlo como el valor de precisión general que tiene el agente recomendador en la UTPL, dando como resultado una precisión del 78,6%, que según (Tolosa, 2008) está en un rango permitido.

Pruebas de Usabilidad

Para estas pruebas en cambio se pretende determinar el nivel de usabilidad del módulo de recomendación, por lo que fue necesaria la participación de un grupo de estudiantes y un docente. Los estudiantes interactuaron con el agente y posteriormente llenaron una encuesta que contiene preguntas orientadas a dar conocer la opinión de los estudiantes acerca del uso y navegación por la herramienta. Para el diseño de las pruebas de usabilidad se consideró el ejemplo propuesto en (Pressman, 2007).

Resultados de las pruebas de usabilidad.

De la encuesta realizada a un grupo de estudiantes y a un docente de la universidad que en total conforman una muestra de 10 personas se han obtenido los siguientes resultados:

Acerca de que si el bloque de recomendación es intuitivo, 8 usuarios consideran que es muy intuitivo y 2 consideran que es poco intuitivo.

De los 10 usuarios que utilizaron el bloque, 5 consideran que se entiende el objetivo del mismo a primera vista y 5 consideran que se entiende poco.

De los 10 usuarios que interactuaron con el bloque, todos consideran que se sintieron cómodos al utilizarlo.

De los 10 usuarios que utilizaron el bloque, todos consideran que es muy fácil de utilizar.

De los 10 usuarios que utilizaron el bloque, 7 consideran que los recursos son muy relevantes y 3 consideran que son poco relevantes.

De la muestra, 8 usuarios consideran que los recursos están muy relacionados con la actividad, mientras 2 consideran que es poca esta relación.

De la muestra, 8 consideran que los recursos sugeridos ayudan mucho a resolver sus actividades pendientes, mientras que 2 consideran que les ayuda poco.

Los resultados de las pruebas de usabilidad han ayudado a mejorar la forma de presentación del bloque de recomendación y del funcionamiento del agente recomendador, haciendo que la interfaz sea más amigable al usuario, debido a que al realizar las pruebas el grupo de estudiantes sugirió algunos cambios.

Un cambio ha sido en los documentos recuperados, en la materia de “Fundamentos Informáticos” se recomendaba algunos recursos que poco tenían que ver con la actividad, esto se solucionó mejorando las descripciones de cada actividad.

Otra sugerencia fue que se mejore la interfaz gráfica, debido a que en una primera impresión no lograron captar la funcionalidad del agente por lo que se realizó un trabajo de carpintería a la aplicación en el cual se especifica de mejor manera el bloque de recomendación

Esta prueba ayudó a mejorar la aplicación y según las preguntas realizadas el agente tiene un nivel de aceptación del 75,25%, lo que se debe a la falta de información acerca del agente y lo que realiza, por último un estudiante hizo la sugerencia de realizar un folleto informativo para el usuario en el que se detalle las características del agente

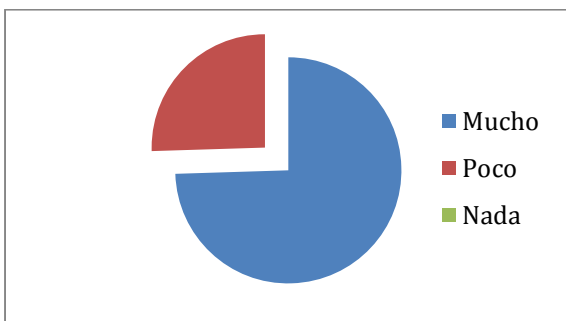


Gráfico 42 Nivel de aceptación

- Mediante los resultados se ha podido comprobar que el agente puede dar soporte al estudiante en sus actividades en el EVA, con la recuperación de un 78,6% de recurso ideales para realizar sus tareas, que en conclusión es un rango aceptable para sistemas de recuperación de información y puede sentar un precedente hacia la construcción de agentes en sistemas de gestión de aprendizaje
- Se concluye que existen diferentes técnicas de recomendación, por lo que en algunos casos hay que utilizarlas en conjunto o extender algún subtipo de las mismas para minimizar los problemas que tienen individualmente.
- Debido a los diferentes problemas que presenta cada una de las técnicas de recomendación se puede utilizar herramientas de la web semántica para solucionar algunos de estos problemas como el “arranque en frío”, que se relaciona con la falta de información puntuada por el usuario, así como también problemas de nuevo usuario o nuevo ítem.
- El agente recomendador obtuvo una precisión del 78,6% y un error del 21,4% esto se debe a los problemas que existen al recuperar información como falsos positivos, ruido documental y silencio informativo. Los problemas de falsos positivos y ruido documental se relacionan a cuando se obtiene un documento que pareciera relevante pero en la realidad no lo es y el problema del silencio informativo es cuando existen documentos relevantes que el agente recomendador no los recomienda.
- Se concluye que un requerimiento indispensable para el funcionamiento del agente es que el entorno le brinde información importante ya que así podrá realizar las inferencias necesarias y poder recuperar los documentos.
- Las tecnologías web semántica como los anotadores semánticos deben ser utilizados por agentes para explotar sus funcionalidades, de otra forma su uso se ve muy limitado.
- Los sistemas recomendadores se basan en algoritmos predictivos que buscan satisfacer necesidades de los usuarios para la toma de decisiones y solucionar la sobrecarga de información, en conclusión este proyecto es un paso inicial a solventar esos problemas en el entorno virtual de aprendizaje.

- Por el crecimiento de información en internet se han dado problemas de sobrecarga de información, por lo que se concluye que se necesitan herramientas que ayuden con estos problemas, hay diferentes opciones y hay que apoyar de alguna manera con proyectos encaminados a sobrellevar los problemas que causa toda esa información en internet.
- Se puede concluir que los sistemas recomendadores y principalmente los basados en conocimiento representan una alternativa muy potente para empresas e instituciones educativas que deseen manejar su información de manera organizada aumentando así la interactividad con sus bases de información.
- En entornos donde se incluye una gran cantidad de información, se debe tener la capacidad de seleccionar la mejor y de una forma rápida, es por esto que se han enfocado proyectos de sistemas de recomendación que filtren esta información y el proceso se pueda dar de una forma rápida. En el transcurso del desarrollo de este tema de tesis se ha podido concluir que las técnicas de recomendación más adecuadas a un entorno educativo son las basadas en memoria, a las cuales se les puede extender con la web semántica y minimizar sus falencias.
- Si bien tiene un funcionamiento aceptable en el entorno de la universidad, se puede mejorar con trabajos a futuro, que pueden comenzar tomando en consideración este proyecto o integrarlo con otro.

III. TRABAJOS FUTUROS

Al realizar este proyecto de tesis se ha visto las diferentes capacidades que tienen los sistemas recomendadores al agregar herramientas de gestión del conocimiento.

Una de las restricciones del proyecto ha sido utilizar la ontología de Kim, que se basa en entidades nombradas como personas, locaciones, fechas importantes, valores, nombres de empresas, entre otras. Al tener la posibilidad de crear ontologías propias adecuadas a los conceptos manejados en un curso y no solamente a entidades nombradas, se puede mejorar en un gran nivel el bloque de recomendación, por ejemplo se puede estar en un curso de Biología y crear una ontología solo de este campo, al momento de enviar las tareas si en la descripción algún concepto está relacionado con la ontología se puede recomendar mejores documentos con un alto nivel de relación semántica con la actividad.

Se puede construir otro tipo de agentes que interactúen con el realizado en esta investigación para realizar un monitoreo de estudiantes que no realizan sus

actividades de aprendizaje y sacar estadísticas de que carreras presenta estudiantes con bajo nivel académico o baja predisposición a realizar sus actividades.

IV. BIBLIOGRAFIA

- (2013). Obtenido de <http://www.ontotext.com/kim>
- Adomavicius, G. (2005). Toward the next generation of recommender system a survey of state of art and posible extensions .
- Álvarez, A. I. (2010). Magadi, una propuesta de sistema adaptativo multi-agente de apoyo al aprendizaje en un marco combinado. Donostia, España.
- Arana, A. I. (2010). Magadi, una propuesta de sistema adaptativo multi-agente de apoyo al aprendizaje en un marco combinado. Donostia, España.
- Burke, R. (2007). Hybrid Web Recommender System . Illinois , Chicago , USA .
- Carrillo, A. F., Llovet, J. M., & Rodríguez, I. (2010). Sistema Recomendador Orientado a la Educación Terapeutica del Paciente Diabetico. Madrid, España.
- Drachslar, H., Hummel, H. G., & Koper, R. (2007). Personal Recommender System for learners in lifelong learning networks: the requirements, techniques and model. The Netherlands.
- García, R. (2008). TFC: XML y web semantica Estudio del impacto de las aplicaciones comerciales basadas en tecnologías de la web semantica.
- Gerling, V. B. (2009). Un Sistema Inteligente para Asistir la Búsqueda Personalizada de Objetos de Aprendizaje.
- Guzmán, C. A. (2009). Sistema de Recomendación y Planificación Turística de la Ciudad de Valencia. Valencia, España.
- Itmazi, J. A. (2005). Sistema Flexible de Gestion del e-learning para soportar el aprendizaje en las universidades tradicionales y abiertas. Granada.
- Llidó, D. M. (2002). Extracción y Recuperación de Información Temporal. Castellón.

- Moodle. (2013). *Moodle Docs* . Obtenido de http://docs.moodle.org/dev/images_dev/5/5a/Moodle2erd.png
- Moodle. (s.f.). *Bloque de Actividades*. Recuperado el 13 de 03 de 2013, de http://docs.moodle.org/24/en/Activities_block
- Moodle. (s.f.). *Moodle*. Recuperado el 13 de 03 de 2013, de Acerca de Moodle: http://docs.moodle.org/24/en/About_Moodle
- Nwana, H. S. (1990). *Intelligent Tutoring Systems: an overview* . Liverpool .
- Olga C Santos, J. G. (s.f.). User´s expirience with a recomender system in an open source standar-based learning management system . Madrid , España .
- Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D., & Goranov, M. (2010). *KIM – Semantic Annotation Platform*. Bulgaria.
- Pressman, R. (2007). *Ingenieria de Software* . ROMA Project. (2007). *Personal Recomender System to Advice Learning Task in Moodle*. Netherlands.
- Rusell , S., & Norving , P. (2004). *Inteligencia Artificial Un Enfoque Moderno* . Madrid : Pearson .
- Santos, O. C., & Boticario, J. G. (2008). User´s expirience with a recomender system in an open source standar-based learning management system. Madrid, España.
- Setten, M. V. (2005). *Supporting People In Finding Information: Hybrid Recommender Systems and Goal-Based Structuring*. Enschede, Netherlands.
- Sirma Group . (2013). Obtenido de <http://www.ontotext.com/kim>
- Suarez, S. (2011). *Manual de Programación del bloque REA de la UTPL*.
- Tolosa, G. (2008). *Introducción a la recuperacion de información* . Buenos Aires , Argentina .
- Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas, M., Motta, E., y otros. (2005). *Semantic annotation for knowledge management: Requirements and a survey of the state of the art*.
- Wang, S.-L., & Wu, C.-Y. (2001). *Application of context-aware and personalized recommendation to implement an adaptive ubiquitous learning system*.
- Wang, S.-L., & Wu, C.-Y. (2011). *Application of context-aware and personalized recommendation to implement an adaptive ubiquitous learning system*. Taiwan .

Anexo 8

Encuesta de pruebas de usabilidad del agente recomendador UTPL

La siguiente encuesta tiene como objetivo determinar la usabilidad y navegabilidad del módulo de recomendación implementado en el EVA de la UTPL

Docente: () Estudiante: ()

USABILIDAD			
Indicador	Nivel de aceptación		
	Mucho	Poco	Nada
1. El módulo de recomendación es intuitivo			
2. Se entiende a primera vista el objetivo de este modulo			
3. Se siente cómodo utilizando el módulo de recomendación			
4. El modulo es fácil de utilizar			
5. Los recursos recomendados son relevantes			
6. El recurso está relacionado con la actividad			
7. Cree que los recursos sugeridos ayudan a resolver las actividades pendientes.			

Al momento de navegar a través de los ítems sugeridos en el módulo de recomendación presentó errores de navegación:

Si () No ()

Si su respuesta es Si, en que opciones se dieron los errores

La navegación por los recursos recomendados es:

Rápida ()

Lenta ()

Adecuada ()

Otra ()

GRACIAS