

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

Programa de Maestría en Computación

Agente Recomendador Híbrido de Objetos de Aprendizaje

Tesis sometida a consideración del Departamento de Computación, para optar por el grado de Magíster Scientiae en Computación, con énfasis en Ciencias de la Computación

Jacqueline Solís Céspedes

**Profesor Asesor:
M.Sc. Mario Chacón Rivas**

Junio, 2014

Agente Recomendador Híbrido de Objetos de Aprendizaje

Resumen

En esta tesis se presentará una investigación para diseñar e implementar un prototipo de Agente Recomendador Híbrido que sugiera recursos de aprendizaje para un curso a partir de la aplicación de técnicas de minería de datos.

Este recomendador toma como insumos el Diseño instruccional y la Teoría de Estilos de Aprendizaje para que a partir de estos elementos se realicen las fases de búsqueda, selección y recomendación de los Objetos de Aprendizaje en un repositorio en la nube.

Abstract

In this thesis research we present the model and prototype of a Hybrid Recommender Agent to suggest learning resources for a course from applying data mining techniques. This recommender takes as inputs the Instructional Design and Learning Styles Theory to perform the search, selection and recommendation of learning objects on a repository in the cloud.

APROBACIÓN DE LA TESIS

“Agente Recomendador Híbrido de Objetos de Aprendizaje”

TRIBUNAL EXAMINADOR



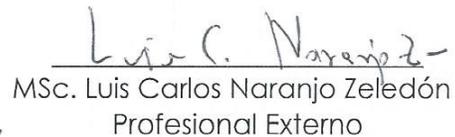
M.Sc. Mario Chacón Rivas
Profesor Asesor



MSc. Isaac Alpizar Chacón
Profesor Lector



M.Sc. Yendry Rojas Hernández
Profesional Externo



MSc. Luis Carlos Naranjo Zeledón
Profesional Externo



Dr. Roberto Cortés Morales
Coordinador Programa de Maestría

Junio, 2014

Dedicatoria

*Dedicado a
mi familia y amigos
más cercanos.*

Agradecimientos

El presente trabajo de tesis primeramente me gustaría agradecerle a mi tutor y jefe, el Ing. Mario Chacón Rivas, M.Sc por su apoyo durante este proceso y por darme la oportunidad de desarrollar y llevar a término este proyecto de investigación que representa una meta más que deseaba profundamente cumplir. Muchas gracias por todo su esfuerzo y dedicación, ya que gracias a sus conocimientos y guía ha logrado que pueda terminar mis estudios con éxito.

También debo agradecer de manera muy especial al equipo del TEC Digital por su ayuda durante el desarrollo de esta tesis, todos han puesto su granito de arena para darle forma al proyecto y siempre han estado a mi lado dándome ánimos o ayudando en cuanto han podido.

A don Oscar Borrell Morant, del Departamento de Educación Comparada e Historia de la Educación de la Universitat de València por su colaboración en la clasificación de los formatos de los recursos de aprendizaje desde el punto de vista pedagógico.

Índice general

Índice de figuras	IX
Índice de tablas	XII
1. Introducción	1
1.1. Organización del documento	2
1.2. Definición del problema	4
1.3. Justificación	5
1.3.1. Innovación	5
1.3.2. Impacto	6
1.3.3. Profundidad	7
1.4. Objetivos	7
1.4.1. Objetivo general	7
1.4.2. Objetivos específicos	7
1.5. Alcance	8
2. Contexto teórico	9
2.1. Aprendizaje permanente	9
2.2. Estilos de aprendizaje	11
2.3. Modelo de aprendizaje basado en competencias	12
2.4. Diseño instruccional	12
2.5. Objetos de aprendizaje en la educación	13
2.5.1. Tipos de Repositorios de Objetos de Aprendizaje	15
3. Generalidades de los sistemas Recomendadores	17
3.1. Sistemas adaptativos al usuario	20

<i>ÍNDICE GENERAL</i>	VII
3.2. Sistemas Recomendadores en el ámbito educativo	21
3.3. Limitaciones de los algoritmos de recomendación	25
3.4. Técnicas de hibridación	27
3.5. Herramientas de recomendación	28
4. Diseño del agente	31
4.1. Metodología	32
4.2. Arquitectura general	33
4.2.1. Tipos de datos	37
4.2.1.1. DI y MoGa	37
4.2.1.2. TEA	38
4.2.1.3. Equivalencias entre cursos	39
4.2.1.4. Consultas al ROA	41
4.2.1.5. Historiales de búsquedas	42
4.2.2. Identificación de las tareas de adaptación	42
4.2.3. Adquisición de datos del curso	43
4.2.4. Generación de la recomendación	47
4.2.4.1. Relevancia de términos	48
4.2.4.2. Recomendación por descriptores	50
4.2.4.3. Aporte del historial de selección de OA	52
4.2.4.4. Recomendación por estilos de aprendizaje	53
4.2.5. Arquitectura detallada	55
4.3. Problemas asociados al diseño	57
5. Implementación y resultados	60
5.1. Consideraciones técnicas	60
5.1.1. Extensiones de PostgreSQL para manipulación de datos	60
5.2. Adquisición del Modelo del Curso	61
5.3. Conexión con ROA	76

<i>ÍNDICE GENERAL</i>	VIII
5.3.1. Integración del OAI-PMH	80
5.3.2. Integración de SQL	82
5.3.3. Tendencias en la disponibilidad de metadatos	84
5.3.4. Preselección de los OA	89
5.4. Generación de la recomendación	90
5.4.1. Hibridación en cascada	91
5.4.1.1. Recomendación basada en contenidos	91
5.4.1.2. Recomendación colaborativa	95
5.4.2. Hibridación mixta	97
5.4.2.1. Clasificación por estilos de aprendizaje	97
5.4.3. Recomendación por Competencias	100
5.5. Interfaz gráfica de usuario	102
6. Conclusiones	105
6.1. Resumen del cumplimiento de los objetivos	110
6.2. Resumen de la contribución al estado del arte	111
6.3. Trabajos futuros	112
6.4. Futuras investigaciones	114
A. Anexos	117
B. Glosario	132
Bibliografía	135
Referencias	135

Índice de figuras

3.1. Clasificación de los sistemas de recomendaciones según el modelo y la heurística utilizada. Tomado de (Adomavicius y Tuzhilin, 2005)	18
3.2. Esquema de arquitectura de AHA. Tomado de Stash y De Bra, 2004.	25
4.1. Tareas relacionadas con el enfoque del tema de investigación	32
4.2. Relación entre conceptos de DI, MoGa, TEA y OA	34
4.3. Diagrama de Interacción de los orígenes de datos	36
4.4. Fragmento de la planificación semanal en un DI para un curso bimodal de programación en TCL.	37
4.5. Sistema de semáforos para la GUI de DI	38
4.6. Representación de cursos equivalente con grafo de expansión generado mediante algoritmo Yifan-Hu Multinivel con 3 capas.	40
4.7. Muestra de datos con agregación de cursos equivalentes	41
4.8. Fragmento de representación de las relaciones de equivalencia mediante grafos dirigidos	41
4.9. Extracción de frases para un perfil de curso usando un enfoque de minería de datos.	44
4.10. Tipos de OA disponibles ARIADNE. Tomado de http://www.ariadne-eu.org/content/more-info	54
4.11. Fragmento de tipos de OA disponibles en el repositorio según AriadneFinder. Disponible en http://ariadne.cs.kuleuven.be/finder/ariadne/	54
4.12. Valores de aprovechamiento asociados a las categorías generales de objetos a partir del criterio experto.	55
4.13. Arquitectura detallada de la solución	56

ÍNDICE DE FIGURAS

X

5.1. Muestra de matriz normalizada	65
5.2. Cardinalidad de resultados de los métodos para extracción de tokens aplicados en el caso de estudio.	66
5.3. Diagrama simplificado de Preprocesamiento de datos	67
5.4. Diagrama del proceso de reducción de frases	68
5.5. Clasificación de los pares de frases aplicando la distancia de Levenshtein y Similarity.	71
5.6. Relación de longitud de los pares de frases con respecto a la distancia de Levenshtein y Similarity.	71
5.7. Resumen de valores extraídos para los pares de frases que tienen relación.	72
5.8. Diagrama del proceso de reducción de tokens	74
5.9. Ejemplo de XML con el perfil de los textos que describen a un curso	76
5.10. Diagrama simplificado de conexión con el ROA	77
5.11. Elementos y estructura del esquema conceptual de LOM.	79
5.12. Diagrama de interacción entre dos repositorios. Tomado de (Simon y cols., 2005).	82
5.13. Ejemplos de sintaxis de los distintos niveles de PLQL	83
5.14. Elementos más frecuentes al especificando la búsqueda sobre los metadatos de idioma utilizando el ISO 639-1 e ISO 639-2 como criterio de búsqueda	87
5.15. Diagrama de Venn con la representación de las consultas realizadas mediante SQL al repositorio.	88
5.16. Muestra de datos obtenida al realizar la búsqueda sin el filtro por idioma para la actividad 1.	90
5.17. Diagrama de secuencia para recomendación por contenidos	92
5.18. Muestra de resultados al calcular similitud de cosenos a los descriptores de OA.	94
5.19. Top-10 OA sugeridos para la actividad 1 del DI.	95
5.20. Aplicación de Hibridación en cascada	96

<i>ÍNDICE DE FIGURAS</i>	XI
5.21. Priorización de coincidencias entre perfiles con ajuste de textos.	97
5.22. Priorización del TF-IDF	97
5.23. Gráfica de utilidad de los recursos generado con WEKA.	98
5.24. Tabla de rendimiento de los clasificadores	99
5.25. Vista del menú para seleccionar actividades	103
5.26. Vista previa de OA que se le ofrecerán al usuario final	104

Índice de tablas

4.1. Perfil básico de curso esperado, sin la agregación de actividades	46
4.2. Perfil general esperado de un curso con agregación de términos de las actividades	46
4.3. Perfil de cursos esperado con los cursos vecinos.	50
A.1. Resumen de la cardinalidad de conjuntos recuperados al consultar por dialectos de Español	119
A.2. Ventajas y Desventajas de las técnicas de recomendación	120

1 Introducción

En una encuesta realizada en Estados Unidos, el 68 % de los estudiantes universitarios y de educación superior reciben una formación basada en demostraciones y lecturas; donde el docente es quién transmite el conocimiento y ellos repiten los procesos; sin embargo, esto no prepara al estudiante para los adaptarse a los cambios, tener pensamiento crítico y adquirir habilidades a largo plazo (Wang, 2007).

Por esta razón, el proceso de enseñanza-aprendizaje se ha replanteado desde el punto de vista pedagógico y de alfabetización en la información; donde el profesor ha pasado a ser un motivador del pensamiento y la incorporación de teorías de aprendizaje colaborativo y de aprendizaje sociocultural, buscan reforzar el pensamiento independiente y reflexivo en el estudiante (Wang, 2007).

Durante la planeación de un curso se analizan y definen los objetivos, habilidades y actitudes que se van a fomentar durante la lección. Para alcanzar los objetivos planteados, se debe guiar el proceso de aprendizaje del estudiante de forma tal que adquiera las habilidades progresivamente y en un contexto particular. De esta manera, definir el objetivo del curso, los mecanismos de evaluación y el plan detallado de las actividades, son sólo una pequeña parte del diseño de un curso y carecen de sentido si no se puede lograr el aprendizaje.

El proceso de planeación o diseño de un curso se basa en gran medida en la selección de las actividades que están sujetas al criterio o conocimiento del docente. Estas actividades, por lo general, son presentaciones o exposiciones magistrales en cursos tradicionalmente presenciales, mixtos o virtuales y se conforman en la mayoría de las veces de lecturas complementadas con actividades de discusión grupal como foros, blogs o wikis.

Además se presenta la característica, según criterio de expertos, que la cantidad de objetos disponibles para un tema específico solo puede restringirse mediante criterios brindados por el experto en la materia a enseñar, lo que hace de la práctica de selección o

recomendación de los objetos una actividad poco frecuentada por el docente.

Los sistemas recomendadores utilizan diversas técnicas al generar recomendaciones personalizadas para los usuarios o grupos de usuario, por ejemplo mediante técnicas de recomendación colaborativas, basadas en contenidos, demográficas, entre muchas otras. Aunque existen otras propuestas de clasificación donde se toman en consideración más que solo las técnicas utilizadas para el análisis de datos (Montaner, López, y De La Rosa, 2003; Adomavicius y Tuzhilin, 2005; Jameson, 2009).

La clasificación de este tipo de sistemas se ha centrado en la forma en que recopilan la información y los algoritmos utilizados para procesarla. Sin embargo, existen sistemas que utilizan dos o más técnicas de recomendación distintas y se denominan recomendadores híbridos. Los sistemas recomendadores híbridos buscan compensar las debilidades de una técnica con las fortalezas de otra al generar las recomendaciones y así mejorar la calidad de las recomendaciones (Burke, 2007; Ricci, Rokach, y Shapira, 2011).

El presente documento pretende exponer una investigación del estado del arte de los agentes recomendadores y un prototipo denominado *Agente Recomendador Híbrido de Objetos de Aprendizaje* (ARHOA en adelante) que tome como base el diseño instruccional de un curso, así como las competencias asociadas a las actividades de la lección. Esto con el fin de analizar los metadatos que describen a una serie de objetos de aprendizaje (OA en adelante) y determinar cuales de ellos podrían ser útiles como recursos en la lección descrita.

1.1. Organización del documento

El presente documento de tesis está organizado en seis capítulos que buscan presentar la propuesta de investigación, su desarrollo y las conclusiones adquiridas tras cumplir los objetivos de investigación planteados.

El capítulo 1 presenta la investigación en términos generales y el impacto esperado al desarrollarla como tesis. Además, acota los alcances y la profundidad de la investigación que se presentará en el documento.

Seguidamente, el capítulo 2 presenta el contexto de la investigación y los conceptos que interactúan para dar forma a este recomendador de objetos de aprendizaje. En este capítulo se expone la influencia de las TIC's en los entornos educativos; así como los diferentes tópicos que son la base para el diseño de la propuesta de investigación, como lo son los estilos de aprendizaje, modelos de atributos, diseño instruccional y los objetos de aprendizaje.

En el capítulo 3 se presentan las generalidades de los sistemas recomendadores y su relación con los entornos educativos. Se presentan las principales características de los sistemas recomendadores, así como los criterios que se utilizan para su clasificación e hibridación.

Después de haber definido los conceptos clave que darán forma a la propuesta y su relación; se presentará el diseño sugerido como mecanismo de resolución para el prototipo de agente recomendador. El capítulo 4 corresponde al diseño del agente y contiene las descripciones de dominio del trabajo, así como una serie de condiciones que fueron tomadas en consideración para satisfacer los objetivos iniciales.

Más adelante, en el capítulo 5 se describe el proceso seguido en la implementación del agente, las herramientas que fueron utilizadas para este fin y los resultados que se obtuvieron tras aplicar cada una de las consideraciones del diseño. También se exhiben algunos elementos clave para realizar el intercambio de información con algunos paquetes de la plataforma del TEC Digital y los repositorios de objetos de aprendizaje.

Y para finalizar, se incluyeron las principales conclusiones alcanzadas al realizar esta tesis y el trabajo futuro que fue identificado como posibles extensiones o aplicaciones del conocimiento adquirido como parte del capítulo 6.

Además, al final del documento se incluye una sección de anexos con información relevante para el diseño e implementación del agente y otra correspondiente a las referencias utilizadas a lo largo de la investigación.

1.2. Definición del problema

La incorporación progresiva de las Tecnologías de Información y Comunicación (TICs) en el proceso de enseñanza-aprendizaje se ha visto como una oportunidad de mejora y expansión al incentivar la participación de los estudiantes y profesores a través del uso de la tecnología (*Enfoques Estratégicos sobre las TICs en Educación en América Latina y el Caribe*, 2013).

El proceso de ubicación, selección y recomendación de los contenidos o evaluaciones asociados a las actividades de un curso es largo y que se basa una serie de criterios que son brindados por un profesor experto en la materia. Además, dada la cantidad de recursos digitales disponibles el principal problema ya no es encontrar materiales, sino, seleccionar aquellos que tienen alguna utilidad para la lección.

Es por este motivo que se propone un modelo para integrar técnicas de minería de datos, cosecha de metadatos mediante OAI-PMH¹ y SQL², diseño centrado en el usuario y procesamiento de lenguaje natural en este proceso de búsqueda y selección de recursos. Para esto, se realizará un análisis de los recursos disponibles en un ROA³, con el fin de identificar los factores críticos a considerar durante las fases de clasificación y recomendación de OA⁴; de modo que fuera posible identificar aquellos con mayor afinidad con respecto a las descripciones brindadas en los documentos facilitados por el profesor.

De este modo, mediante la aplicación de distintas técnicas de minería de datos y textual, se propone la utilización de recursos como el Diseño Instruccional (DI), el Modelo de Competencias o de Gestión de Atributos (MoGa) y la Teoría de Estilos de Aprendizaje (TEA) como elementos estructurales para la selección de recursos; ya que aportan una descripción detallada de los elementos que deben ser cubiertos durante la lección.

¹OAI-PMH, Acrónimo de *Open Archives Initiative Protocol for Metadata Harvesting* o Iniciativa de Protocolo de Archivos Abiertos para la Recolección de Metadatos.

²SQL, Acrónimo de Simple Query Interface y responde al identificador CWA 15454:2005 del Comité Europeo de Normalización(CEN).

³ROA, Acrónimo de *Repositorio de Objeto de Aprendizaje*

⁴OA, Acrónimo de *Objeto de Aprendizaje*

Tras concretar el proceso de investigación y avanzar con el desarrollo del prototipo, se encontró que el agente sólo podría utilizar el DI y la información disponible en TEA para catalogar los OA; y quedó pendiente la incorporación del MoGA en las fases de recomendación.

1.3. Justificación

El Instituto Tecnológico de Costa Rica (TEC) es una institución dedicada a la educación superior en distintas áreas de especialización. En este contexto, el TEC Digital representa la unidad dedicada a dar apoyo a la academia mediante la integración de las TICs en la docencia⁵.

En esta tesis se presentan las bases teóricas y un prototipo de herramienta integrada a la plataforma educativa TEC Digital, capaz de analizar y recomendar una serie de objetos de aprendizaje con distintos niveles de afinidad con respecto a las actividades definidas en un DI, MoGa y TEA.

A continuación, se describe su justificación en términos de innovación, impacto y profundidad.

1.3.1. Innovación

Al realizar un análisis de la literatura se encuentran referencias a trabajos previos relacionados con el desarrollo de agentes de búsqueda de objetos de aprendizaje, por ejemplo LORSE, que crea múltiples agentes que se conectan a distintos repositorios de OA (Baldiris, Bacca, Rojas, Guevara, y Fabregat, 2011). Por otra parte, DELPHOS se centra en características como la palabras clave, frecuencia de descarga, similaridad o evaluación previa del OA utilizando el descubrimiento mediante vecinos más cercano (Zapata-Gonzalez, Menendez, Prieto, y Romero, 2011) para establecer sus recomendaciones.

⁵Disponible en <http://www.tec-digital.itcr.ac.cr/register/sitio/quienessomos.html>

Sin embargo, ambos agentes están relacionados con búsquedas a partir de los descriptores introducidos manualmente por el usuario y al historial de consultas de un repositorio particular.

Con el Agente Recomendador Híbrido de Objetos de Aprendizaje, se propone utilizar la información disponible en los documentos del DI, MoGa y TEA para recomendar un banco de recursos para actividades previamente descritas por el profesor; de modo que la tarea de filtrar los recursos se simplifique tras automatizar la selección de descriptores para una actividad y aplicar una serie de criterios de preselección y calificación que van más allá de la habilidad de un motor de búsqueda.

1.3.2. Impacto

Con la creación de un Agente Recomendador Híbrido de Objetos de Aprendizaje para sugerir los OA asociados a las actividades descritas en un DI, se le puede delegar la recomendación de los OA al agente, de modo que elija aquellos que responda a las descripciones brindadas por el profesor. Además, se pueden definir criterios de recomendación que permitan filtrar desde distintos repositorios solo aquellos elementos que mejor se adapten a los requerimientos del LMS.

Este agente funcionará para cualquier curso para el que se defina el DI; sin embargo, los principales beneficiados serán aquellos facilitadores que se encuentren trabajando en procesos de migración de sus cursos presenciales a bimodales o virtuales, ya que podrán reducir el tiempo requerido para filtrar y seleccionar los recursos digitales que le podrían interesar para sus lecciones.

Los OA usados para una lección pueden ser documentos comunes en diferentes formatos, por ejemplo documentos de texto, presentaciones, gráficos, entre otros; o bien, pueden estar empaquetados en formato como IMS-LD, SCORM o Common Cartridge.

Los OA empaquetados pueden cargarse dentro de Unidades de Aprendizaje (UA). Donde se entienda UA como paquetes de materiales que pueden incluir actividades, evaluaciones, servicios, entre otros; que son ejecutables por un LMS y en su definición se pueden

incluir objetivos de aprendizaje, roles, actividades y recursos, entre otros; mientras que los OA son cualquier recurso que pueden ser utilizados o referenciado durante el aprendizaje (Chacón Rivas y Garita, 2013).

De este modo, se apoyará en el tiempo que los docentes deben dedicar para la construcción de materiales nuevos y les permitirá dedicarle más tiempo a las labores asociadas a transmitir conocimiento.

1.3.3. Profundidad

En términos generales la investigación se centrará en validar la hipótesis de que es posible recomendar un banco de recursos a partir de los documentos facilitados por el docente mediante la aplicación de técnicas de minería de datos y procedimientos de calificación que son utilizados en los sistemas recomendadores.

1.4. Objetivos

1.4.1. Objetivo general

Diseñar e implementar un prototipo de agente que sugiera recursos consumidos desde repositorios de objetos de aprendizaje de acuerdo al modelo planteado en un diseño instruccional y que se ajusten al modelo de competencias oficial establecido para un curso y los estilos de aprendizaje de los estudiantes.

1.4.2. Objetivos específicos

- Elaborar un estudio del estado del arte en el dominio de agentes recomendadores de objetos de aprendizaje y su relación con las plataformas de gestión del aprendizaje.
- Analizar los objetos de aprendizaje e implementar una herramienta que permita evaluar el contenido de sus elementos descriptores.

- Utilizar el DI de un curso como base para sugerir la recomendación de recursos de aprendizaje.
- Implementar un mecanismo que permita sugerir aquellos recursos disponibles en un repositorio de objetos de aprendizaje que fomenten las competencias de los estudiantes sobre la estructura definida para un curso.

1.5. Alcance

- No se considerará la evaluación y catalogación de calidad de OA como parte de los alcances de la tesis; ya que existen otras herramientas que se pueden encargar de este proceso y responden a un problema distinto al que se desea abordar.
- Se asume que existen esquemas para el manejo de las competencias y del DI con información precargada, antes de la definición del agente.
- La definición de las de competencias o atributos a utilizar queda fuera del alcance de la tesis.
- Se asume que la evaluación y la catalogación inicial de los OA es dada por un agente externo.
- Se encuentra fuera de los alcances de esta tesis la creación y/o configuración de un LORs.
- La presente propuesta se enfoca en términos de la evaluación de la calidad de los criterios de búsqueda con respecto a los metadatos, por lo tanto, la interpretación del lenguaje natural excede los alcances establecidos.

2 Contexto teórico

En este capítulo se presentarán una serie de conceptos relacionados con psicología, pedagogía y computación que son claves para comprender el contexto en el que se desarrolla la investigación de tesis y las características de la información con la que se va a trabajar.

El capítulo está dividido en cinco secciones, donde se describen en términos generales el concepto de aprendizaje permanente, la influencia del aprendizaje por competencias en la industria, la relación de los estilos de aprendizaje con las competencias y se introduce el concepto de OA.

2.1. Aprendizaje permanente

El concepto de aprendizaje permanente o *"lifelong learning"* fue tomado dentro del proceso de Bolonia para crear "una sociedad basada en el conocimiento donde, las estrategias de aprendizaje son necesarias para enfrentar los retos de la competitividad y el uso de las nuevas tecnologías, mejorar la cohesión social, la igualdad de oportunidades y la calidad de vida" (*Bologna Process*, 1999).

Este proceso, constituye una iniciativa que busca solventar los cambios en el proceso de enseñanza-aprendizaje que se dieron en los últimos años en el sistema de educación superior dentro del marco europeo y se enfoca en la adquisición de competencias, o resultados de aprendizaje que contribuyan al correcto desempeño de los profesionales en un área particular, del mismo modo que se ha intentado en América Latina para empezar a crear una sociedad que se adapte al constante cambio del mercado (*Propuestas y Acciones para la transformación de la Educación Superior en América Latina*, s.f.).

De este modo, se puede decir que los trabajos convergen en la necesidad de replantear los planes de estudio mediante herramientas pedagógicas que faciliten la adquisición

de conocimientos, competencias y aptitudes en las distintas disciplinas. Desarrollando un plan de acción global de educación permanente y homologación de competencias (*Declaración mundial sobre la educación superior en el siglo XXI: Visión y Acción*, 1998). La evolución de las tecnologías de información ha repercutido directamente en la educación, por lo que se han realizado esfuerzos que permitan aprovechar estas nuevas herramientas en favor del aprendizaje. Un ejemplo de estas herramientas son los LMS¹; sin embargo, estas herramientas son de poca utilidad si no se pueden gestionar los contenidos y adaptarlos a las necesidades de los estudiantes (Boticario y Santos, 2007). Las nuevas tendencias, enfocan al usuario como una unidad con necesidades propias y únicas en un entorno particular, por lo que en el ámbito educativo ha aparecido un nuevo tipo de entorno, denominado PLE² y buscan solventar las necesidades específicas de cada estudiante mediante agentes o minería de datos.

Algunos ejemplos de estas adaptaciones son:

- Sistema para Generación de Sugerencias de Rutas de Aprendizaje Adaptativas en Entornos de e-learning: demostró la utilidad de aprovechar las rutas adaptativas para reforzar el conocimiento sobre el contenido de un curso tras aplicar redes bayesianas para decidir si una actividad es recomendable para un estudiante dado su promedio (Gámez Suazo, 2012; Suazo, Rodríguez, y Rivas, 2012).
- LORSE: crea múltiples agentes que se conectan a distintos repositorios de OA y actúan como buscadores (Baldiris y cols., 2011).
- DELPHOS: es un sistema recomendador híbrido para el repositorio AGORA³ y basa sus sugerencias en palabras clave, frecuencia de descarga, similaridad o evaluación previa del OA utilizando el descubrimiento mediante vecinos más cercano (Zapata-Gonzalez y cols., 2011).

¹LMS: Acrónimo para *Learning Management Systems* o Sistema de Gestión de Aprendizaje por sus siglas en inglés.

²PLE: Acrónimo para *Personal Learning Environments*, o Entornos de Aprendizaje Personalizados por sus siglas en inglés.

³AGORA es el acrónimo para Ayuda para la Gestión de Objetos Reusables de Aprendizaje.

Un ejemplo pero en otra línea de diseño, propone una evolución de la arquitectura de los OA hacia agentes inteligentes denominados ILOs⁴, gobernados por un ILOR⁵ que es comandado desde el LMS. En esta propuesta cada agente de OA autogestiona sus atributos para interactuar con el administrador gracias a una cualidad de los OA en la que se puede separar el contenido de la descripción del entorno (Gomes, 2005).

2.2. Estilos de aprendizaje

Según (Sternberg y Grigorenko, 1997), el estudio de los estilos de aprendizaje constituye un vínculo entre dos ramas de la psicología, conocidas como cognición y personalidad; por lo que la forma en la que se razona y percibe la información puede variar con el tiempo (Witkin, Goodenough, y cols., 1981).

Para corregir el desfase del aprendizaje en la etapa educativa y la incursión en un trabajo, se ha propuesto una serie de enfoques que pretenden mejorar la calidad de la educación para ajustarla a las necesidades de la industria, por ejemplo, el proceso de Bolonia, que se enfoca en la obtención de competencias. Estas competencias son vistas como resultados de aprendizaje cuantificables y contribuyen al correcto desempeño de los profesionales en un área en particular (*Bologna Process*, 1999; *Síntesis de la Legislación de la UE*, s.f.) .

La categorización y utilización de técnicas de enseñanza que se pueden aplicar para los distintos estilos de aprendizaje busca entregarle al docente una herramienta que le permita ajustar su estilo de enseñanza al estilo de aprendizaje preferido por sus estudiantes y desarrollar una lección de manera que todos reciban la información de un modo accesible (Felder y Silverman, 1988; Felder, Felder, y Dietz, 2002).

⁴ILO: Acrónimo para *Intelligent Learning Object* o Objetos de Aprendizaje Inteligentes por sus siglas en inglés.

⁵ILOR: Acrónimo para *Intelligent Learning Object Repository* o Repositorio de Objetos de Aprendizaje Inteligentes por sus siglas en inglés.

2.3. Modelo de aprendizaje basado en competencias

La tendencia del mundo globalizado busca crear un profesional con un alto conocimiento técnico y con una formación integral que les permita adaptarse a los cambios del mercado para generar resultados. Producto de este nuevo esquema de necesidades, se han replanteando las metodologías de trabajo, buscando la generación de habilidades y capacidades en los estudiantes, más que enfocarse en la transmisión de conocimientos y es en este nuevo paradigma donde se introduce el concepto de modelo curricular basado en competencias (Mertens, 1998).

A nivel académico es frecuente referenciar los modelos basados en competencias como “modelos basados en atributos”. En (*Canadian Engineering Accreditation Board. Accreditation Criteria and Procedures*, 2010) se considera el uso del término “competencia” para el ámbito de la práctica laboral o en la industria, por lo que es frecuente encontrar en la literatura el uso de los términos competencias, atributos, habilidades blandas, resultados de aprendizaje, syllabus, entre otros, en forma indiferente (Gluga, Kay, y Lever, 2012).

2.4. Diseño instruccional

Según (Arjona y Blando, 2005), “El Diseño Instruccional es un proceso en donde se analizan, organizan y presentan objetivos, información, actividades, métodos, medios y el proceso de evaluación, que al conjugarse entre sí conforman el contenido de un curso con miras a generar experiencias satisfactorias de aprendizaje.”, o en otras palabras, es un proceso que el profesor lleva a cabo y que consiste en planificar el curso, tema por tema y actividad por actividad para guiar el proceso educativo del estudiante.

De esta forma, se denomina “diseño instruccional” tanto al proceso como al producto que se genera. Siendo el resultado, generalmente una tabla o documento con el detalle de las actividades a desarrollar durante las lecciones; en este detalle se incluyen los objetivos, contenidos, materiales a utilizar y las evaluaciones de los aprendizajes.

Este proceso estructura un curso o unidades de aprendizaje como una colección de con-

ceptos compuestos por características de los estudiantes, temas o contenidos, objetivos actitudinales y de conceptos, materiales o actividades y evaluaciones.

El proceso de planeación o diseño de un curso se basa en gran medida en la selección de las actividades que están sujetas al criterio o conocimiento del docente y de las fuentes de información disponibles en el momento. Estas actividades son presentaciones o exposiciones magistrales en los cursos tradicionalmente presenciales; mientras que en curso mixtos y virtuales se conforman la mayoría de las veces de lecturas complementadas con actividades de discusión grupal como foros, blogs o wikis.

2.5. Objetos de aprendizaje en la educación

Según la IEEE, un OA se define como “cualquier entidad, digital o no digital, que puede ser usada para el aprendizaje, educación o capacitación” (*IEEE-Learning Technology Standards Committee*, s.f.). Esto quiere decir, que cualquier elemento didáctico desde imágenes, documentos, herramientas de evaluación, etc, puede ser considerado como OA.

El estándar LOM-IEEE ⁶ ofrece un esquema conceptual para la estructura de las instancias de metadatos de los OA que permita la búsqueda, evaluación, adquisición y uso de los recursos por parte de los instructores, estudiantes o procesos de software automatizados (*Draft Standard for Learning Object Metadata*, 2002); este estándar está avalado por el LTSC ⁷ de la IEEE, quién es el encargado de producir especificaciones y estándares para las tecnologías de e-learning (*Draft Standard for Learning Object Metadata*, 2002).

Existen múltiples iniciativas para estandarizar los OA y convertirlos en elementos portables entre las plataformas y muchas de estas iniciativas se basan en el uso de metadatos para describir el contenido de cada ítem. Estos metadatos se utilizan como llaves dentro

⁶LOM-IEEE, Acrónimo de *Learning Object Model*, o Modelo de Objetos de Aprendizaje propuesto por la IEEE.

⁷LTSC, Acrónimo de *Learning Technology Standards Committee* o Comité de Estándares para las Tecnologías de Aprendizaje.

de un sistema de catálogos o repositorios en donde los ítems u OA pueden ser “cosechados” con base en sus descriptores.

Por ejemplo, la iniciativa de OpenArchives busca proveer un framework de interoperabilidad basado en cosecha de metadatos y que se denomina OAI-PMH; donde hay dos tipos principales de participantes del proceso: los proveedores de datos que son los administradores del sistema que ofrecen soporte para el OAI-PMH y administran los repositorios y los proveedores de servicios, quienes usan los metadatos recolectados vía OAI-PMH como una base para crear servicios con valor agregado (Lagoze, Van de Sompel, Nelson, y Warner, 2002).

Existen modelos para empaquetar OA con el fin de lograr su reutilización en diferentes LMS. Estos modelos son:

- SCORM⁸: fue propuesto por ADL⁹ y comprende “un conjunto de estándares técnicos que le permiten a los sistemas de aprendizaje basados en la Web encontrar, importar, compartir, reutilizar y exportar contenidos en una forma estandarizada” (*Advanced Distributed Learning Initiative: SCORM 2004 4th Edition*, s.f.).
- IMS-LD: propuesto por el IMS Global Consortium y permite “especificar distintas estrategias de aprendizaje y cómo las fuentes de aprendizaje encajan dentro de las mismas” (Vélez Ramos, Baldiris, Nassiff, y Fabregat Gesa, 2008). Este modelo describe cómo empaquetar y transferir contenido usando XML para organizar los OA, y permite definir distintas formas de recorrer un conjunto de mismo recursos de aprendizaje (*IMS Global Learning Consortium*, 2012).
- Common Cartridge: También fue propuesto por el IMS Global Consortium y fue definido para resolver dos problemas; el primero, proveer un formato estándar para representar materiales de cursos en los LMS y el segundo, permitir la publicación de

⁸SCORM, Acrónimo de *Shareable Content Object Reference Model* o Modelo de Referencia de Objetos de Contenido Compatible.

⁹ADL, Acrónimo de *Advanced Distributed Learning*.

materiales en forma “modular, distribuida, interactiva y personalizable” (*IMS Global Common Cartridge Profile: Implementation*, 2014).

En general, los diferentes enfoques coinciden que los objetos de aprendizaje deben cumplir al menos las siguientes características:

- **Accesibilidad:** Los OA pueden ser almacenados y referenciados desde la base de datos gracias a los metadatos con los que fueron etiquetados.
- **Reusabilidad:** Permite la reutilización de los recursos en distintos contextos.
- **Interoperabilidad:** los OA puede ser independientes de los sistemas de administración de conocimiento.

2.5.1. Tipos de Repositorios de Objetos de Aprendizaje

Los repositorios de OA nacen en respuesta a las necesidades de los entorno de e-learning para reutilizar material didáctico, con el fin de almacenar y compartir OA digitales. Estos elementos se presentan como módulos de contenidos y metadatos que puedan ser reutilizables por las comunidades o cursos y responden al paradigma de orientación a objetos (Bannan-Ritland, Dabbagh, y Murphy, 2000; Wiley, 2003).

En estas bibliotecas digitales se almacenan diversos tipos de recursos de aprendizaje y los descriptores que ayudan a categorizarlos, con lo que se facilita el flujo de contenidos y las búsquedas gracias a los mecanismos de indexación (López, 2005).

Además, estas bibliotecas constituyen una oportunidad ante la creciente demanda de educación en todos los países, la capacidad limitada de la educación tradicional, el esfuerzo y costo que implica la construcción de materiales de aprendizaje multimedia y aprovechan las posibilidades que ofrece Internet (*Globe Alliance*, 2012).

Las federaciones de OA buscan organizar los materiales alojados en diversos repositorios de modo que se centralice el acceso a la información contenida en un solo portal. Un ejemplo de estas organizaciones es la alianza GLOBE, la cuál ofrece un conjunto de servicios y herramientas en línea a disposición de sus miembros para el intercambio de

recursos de aprendizaje y se presenta como una comunidad mundial abierta guiada por una serie de principios que buscan fomentar el intercambio de OA y la participación de los miembros como colaboradores activos tanto en la producción de material, como en las mejoras a los estándares abiertos con base a sus experiencias de implementación(*Globe Alliance*, 2012).

Todos estos conceptos pertenecen a distintas ramas de conocimiento, pero es necesario comprender su utilidad para aprovechar la relación que existe entre ellos al modelar una solución que pretenda integrarlas. De este modo, los conceptos mencionados en este capítulo se estarán utilizando más adelante en los capítulos de diseño de la solución y resultados de la implementación.

3 Generalidades de los sistemas Recomendadores

Los sistemas recomendadores son “sistemas de información que sugieren ítems a usuarios basados en su comportamiento y/o otros tipos de datos” (Ricci y cols., 2011) y en su mayoría están enfocados a procesos de toma de decisiones donde el usuario no tiene suficiente conocimiento para evaluar las alternativas de ítems que podrían resultar potencialmente de interés.

Según (Ricci y cols., 2011), la función principal de los sistemas recomendadores depende de los objetivos que hayan sido planteados al momento de su definición e implementación; pero en términos generales suelen utilizar tres tipos de objetos:

- Ítems: Se le llama ítem a los elementos que se van a recomendar.
- Usuarios: Los modelos de usuario codifican las necesidades y preferencias de un usuario dentro del sistema.
- Transacciones: Son las interacciones entre el usuario y el sistema y pueden ser recolectados de forma implícita o explícita.

Una vez identificadas sus partes, los sistemas recomendadores tienen como función principal identificar los ítems que puede ser de utilidad para un usuario. De modo que debe ser capaz de predecir la utilidad esperada de cada ítem para sólo seleccionar los que sean de interés y clasificarlos con base a algún mecanismo de comparación (Ricci y cols., 2011) . Desde el año 2006, la ACM organiza una conferencia llamada “RecSys” (Recommender Systems) dedicada a la presentación de resultados de las nuevas investigaciones, sistemas y técnicas aplicadas a diversos campos de los sistema de recomendaciones (por ejemplo, algoritmos de escalabilidad, casos de estudio, grupos de recomendación, modelado de usuario, aprendizaje automático para recomendaciones, entre otras).

Según la ACM, se define recomendación como una forma particular de filtrado de información que explota los comportamientos anteriores de los usuarios y sus similitudes para generar una lista de información a la medida de las preferencias de un usuario final (ACM, 2012).

Existen numerosos enfoques para el desarrollo de sistemas recomendadores y algunos

Recommendation Approach	Recommendation Technique	
	Heuristic-based	Model-based
Content-based	Commonly used techniques: <ul style="list-style-type: none"> • TF-IDF (information retrieval) • Clustering Representative research examples: <ul style="list-style-type: none"> • Lang 1995 • Balabanovic & Shoham 1997 • Pazzani & Billsus 1997 	Commonly used techniques: <ul style="list-style-type: none"> • Bayesian classifiers • Clustering • Decision trees • Artificial neural networks Representative research examples: <ul style="list-style-type: none"> • Pazzani & Billsus 1997 • Mooney et al. 1998 • Mooney & Roy 1999 • Billsus & Pazzani 1999, 2000 • Zhang et al. 2002
Collaborative	Commonly used techniques: <ul style="list-style-type: none"> • Nearest neighbor (cosine, correlation) • Clustering • Graph theory Representative research examples: <ul style="list-style-type: none"> • Resnick et al. 1994 • Hill et al. 1995 • Shardanand & Maes 1995 • Breese et al. 1998 • Nakamura & Abe 1998 • Aggarwal et al. 1999 • Delgado & Ishii 1999 • Pennock & Horwitz 1999 • Sarwar et al. 2001 	Commonly used techniques: <ul style="list-style-type: none"> • Bayesian networks • Clustering • Artificial neural networks • Linear regression • Probabilistic models Representative research examples: <ul style="list-style-type: none"> • Billsus & Pazzani 1998 • Breese et al. 1998 • Ungar & Foster 1998 • Chien & George 1999 • Getoor & Sahami 1999 • Pennock & Horwitz 1999 • Goldberg et al. 2001 • Kumar et al. 2001 • Pavlov & Pennock 2002 • Shani et al. 2002 • Yu et al. 2002, 2004 • Hofmann 2003, 2004 • Marlin 2003 • Si & Jin 2003
Hybrid	Combining content-based and collaborative components using: <ul style="list-style-type: none"> • Linear combination of predicted ratings • Various voting schemes • Incorporating one component as a part of the heuristic for the other Representative research examples: <ul style="list-style-type: none"> • Balabanovic & Shoham 1997 • Claypool et al. 1999 • Good et al. 1999 • Pazzani 1999 • Billsus & Pazzani 2000 • Tran & Cohen 2000 • Melville et al. 2002 	Combining content-based and collaborative components by: <ul style="list-style-type: none"> • Incorporating one component as a part of the model for the other • Building one unifying model Representative research examples: <ul style="list-style-type: none"> • Basu et al. 1998 • Condliff et al. 1999 • Soboroff & Nicholas 1999 • Ansari et al. 2000 • Popescul et al. 2001 • Schein et al. 2002

Figura 3.1: Clasificación de los sistemas de recomendaciones según el modelo y la heurística utilizada. Tomado de (Adomavicius y Tuzhilin, 2005)

de ellos fueron investigados por Adomavicius y Tuzhilin. En su análisis, se sintetizaron las

características de varios sistemas y presentó un resumen de los resultados obtenidos en la tabla de la figura 3.1 que muestra las formas más comunes para generar las recomendaciones de acuerdo al tipo de implementación y a las heurísticas utilizadas (Adomavicius y Tuzhilin, 2005).

Por otro lado, Jameson propone una serie de clasificaciones alternativas donde se incluye una para la forma en que los sistemas adaptables al usuario le ayudan a recuperar información (Jameson, 2009). Según su propuesta, existen tres situaciones típicas en las que podemos identificar los enfoques de un sistema:

- Soporte a navegador: donde los usuarios pueden realizar búsquedas sobre ítems y usar las referencias cruzadas entre ellos, como en los sistemas hipermedia o de noticias.
- Soporte para búsqueda y filtrado basado en consultas: como los mecanismos de búsqueda por palabras clave, pero en este enfoque se mejoran los criterios de selección mediante el modelo construido por el usuario con base en otros recursos de información como documentos o criterios de orden.
- Proveer información de forma espontánea: se captura información mientras los usuarios realizan diversas tareas y con base en esto se le presenta información que puede ser útil para el usuario.

Se define agente como aquellas aplicaciones que toman decisiones por sí mismas con el fin de conseguir sus objetivos de una forma rápida, en un entorno cambiante y con una alta probabilidad de fallo (Weiß, 1999).

Según (Russell y Norvig, 2002) existen diversos tipos de agentes que pueden ser categorizados de acuerdo a las propiedades del entorno en las que se desenvuelven, pero el entorno no es la única variable en cuanto al comportamiento de los agentes. Para definir a un agente como inteligente, este debe ser capaz de realizar acciones flexibles y autónomas para alcanzar sus objetivos de diseño, donde flexibilidad significa tres cosas (Weiß, 1999) :

- **Reactividad:** El agente debe percibir su entorno y responder a los cambios que ocurren para satisfacer los objetivos de diseño.
- **Pro-actividad:** debe ser capaz de mostrar un comportamiento enfocado en alcanzar los resultados y tomar la iniciativa para satisfacer los objetivos.
- **Habilidad social:** debe ser capaz de interactuar con otros agentes y tal vez humanos, para satisfacer sus objetivos.

3.1. Sistemas adaptativos al usuario

En los entornos de e-learning existen numerosos trabajos sobre interfaces adaptativas y agentes que aprenden sobre el comportamiento de usuarios individuales y adaptan su comportamiento de formas no triviales. Según la clasificación de Jameson, los sistemas UAS¹ se pueden adaptar de acuerdo a su forma y la funcionalidad y una de sus principales aplicaciones prácticas está en los sistemas para recomendaciones de productos o de análisis de afinidad que se usan en los sitios web comerciales (Jameson, 2009).

Estos sistemas ofrecen tres enfoques principales para seleccionar los productos: el primero, con base en un único ranking definido por el usuario que utiliza el sistema, el segundo mediante filtros colaborativos y el tercero, mediante un ciclo iterativo en el que el sistema propone una solución y la reevalúa con el ranking que le da el usuario a su selección. La diferencia clave entre los sistemas adaptativos y otros tipos de sistemas interactivos está en la utilización de métodos para adquirir y explotar los modelos de usuarios.

Los sistemas de recomendaciones le agregan a los entornos en línea una guía sobre las tendencias del comportamiento de las personas y realizan las sugerencias basadas en esta información, por lo que pueden ser catalogados como sistemas que intentan actuar como humanos. Para lograr esto, se involucran modelos matemáticos para determinar la relación que pueden tener las acciones de una persona con respecto a las demás y se pueden interpretar como una asociación de comportamientos.

¹UAS, acrónimo de *User Adaptive Systems*, o Sistemas Adaptativos al Usuario.

Estos sistemas suelen categorizar sus recomendaciones en tres tipos principales: basados en contenidos, colaborativos y los enfoques de recomendaciones híbridos (Adomavicius y Tuzhilin, 2005).

Según esta última clasificación, se puede decir que el agente propuesto se puede ubicar dentro de la categoría de búsqueda y filtrado basado en consultas. De este modo, Jameson identifica tres principales problemas que debe enfrentar un sistema de recomendaciones con estas características:

- El usuario puede desconocer los aspectos del producto que le interesan o que pueden ser determinantes en su decisión.
- Si el usuario no está relacionado con los conceptos usados para describir el producto no podrá realizar búsquedas efectivas.
- Ordenar de forma adecuada la información que se recuperó para tomar una decisión y mostrársela al usuario.

A estos problemas se puede agregar la falta de información suficiente y relevante para adaptarse y que los procesos de recuperación de información puede ser complejos e influir en los procesos de decisión de un modo innecesario.

3.2. Sistemas Recomendadores en el ámbito educativo

En el contexto particular de la educación, los sistemas recomendadores están dando sus primeros pasos, por lo que es necesario realizar una revisión del estado del arte de la aplicación de dichas tecnologías actualmente disponibles en la academia.

Según (Manouselis, Drachsler, Vuorikari, Hummel, y Koper, 2011) en un análisis de diversos sistemas de recomendación utilizados para las tecnologías educativas, se plantea que la cantidad de recursos potencialmente útiles para la educación puede resultar abrumadora; y es por este motivo, que los sistemas recomendadores han tomado fuerza en el ámbito educativo. Su propuesta se centró en un análisis de los recursos disponibles en

las áreas de Hipermedia Adaptativa para la Educación y redes de aprendizaje que han colaborado en el desarrollo de los sistemas denominados TELs ², tanto en los sistemas educativos formales, como los informales.

En su recopilación, los autores aclaran que los objetivos para los sistemas recomendadores vinculados a procesos educativos difieren de los de sus contrapartes comerciales e incluso pueden agregar requerimientos nuevos a problemas sencillos. Es por este motivo, que en su contexto particular, existen modelos y teorías pedagógicas que tienen como misión enriquecer el aprendizaje y que deben tomarse en consideración antes de iniciar cualquier implementación.

Los primeros intentos de incorporar las funcionalidades de los sistemas recomendadores en procesos de aprendizaje se remontan aproximadamente al año 2000. Algunos de ellos que han sido mencionados en (Manouselis y cols., 2011), son los siguientes:

- Altered Vista: fue creado para propagar información sobre la calidad de los recursos de aprendizaje (Manouselis y cols., 2011).
- RACOFI³, el cuál fue presentado de manera informal el 12 de Agosto del 2003 y su objetivo fue realizar recomendaciones a usuarios en línea acerca de audios correspondientes a objetos de aprendizaje utilizando filtros basados en reglas y colaborativos (Anderson y cols., 2003; Manouselis y cols., 2011).
- QSIA ⁴: Es un sistema enfocado en compartir recursos que le permite al usuario determinar quienes van a ser parte de su insumo en el filtro colaborativo (Manouselis y cols., 2011).
- CYCLADES: Es un sistema que propone la evaluación de distintos recursos digitales disponibles en repositorios a través del OAI-PMH. Este sistema utilizó filtros colaborativos para determinar las posibles recomendaciones útiles a sus usuarios;

²TEL, Acrónimo de *Technology Enhanced Learning*

³RACOFI, Acrónimo de *Rule-Applying Collaborative Filtering*

⁴QSIA, Acrónimo de *Question Sharing and Interactive Assignments*

enfocándose en el intercambio de documentos. (Gross, 2003; Manouselis y cols., 2011).

- CoFind⁵: Es un prototipo capaz de aprovechar recursos libres disponibles en internet y centró su enfoque en el uso de etiquetas para exponer los metadatos del contexto educativo (Manouselis y cols., 2011).
- ISIS: Utiliza un enfoque híbrido para recomendar rutas de navegación sobre recursos de aprendizaje utilizando información social relacionada a otros usuarios y metadatos de los estudiantes y actividades de aprendizaje (Drachsler y cols., 2009; Manouselis y cols., 2011).
- TORMES⁶: Es un sistema recomendador semántico basado en conocimientos diseñando para reflejar las necesidades del estudiante al estar bajo distintos escenarios y al mismo tiempo, ofrecerle al educador control al seleccionar lo que se le va a estar presentando a sus alumnos (Santos y Boticario, 2010, 2011). Fue diseñado por el grupo de investigación aDeNu⁷ de la UNED España, quienes han dado grandes pasos en cuanto a la aplicación de sistemas recomendadores a distintas etapas del proceso educativo.
- Willow: Es un sistema de evaluaciones adaptativo asistido por computadora mediante la aplicación de técnicas de minería de datos (Pascual-Nieto, Santos, Perez-Marin, y Boticario, 2011). Este sistema ya ofrecía soporte para modelos de usuario y procesamiento de lenguaje natural como su antecesor Atenea, y se le creó una extensión para ofrecer las recomendaciones en los distintos escenarios enfocándose en el proceso de selección de las preguntas (Perez-Marin, Alfonseca, Rodriguez,

⁵CoFind, Acrónimo de *Collaborative Filter in N Dimensions*

⁶TORMES, Acrónimo de *Tutor-Oriented Recommendations Management for Educational Systems*, es un sistema diseñado para integrarse con la plataforma educativa mediante servicios web con el fin de recomendarle al educador recursos disponibles dentro de la misma plataforma

⁷aDeNu, Acrónimo de Adaptación Dinámica de sistemas de Educación oNline basada en el modelado de Usuario. Disponible en <https://adenu.ia.uned.es/web/>

y Pascual-Nieto, 2006; Pérez-Marín, Alfonseca, y Rodríguez, 2006).

- Entre otros, los cuales se encuentran en su mayoría en fase de diseño o prototipos y existen pocos registros sobre validaciones incluyendo interacción humana (Manouselis y cols., 2011).

El enfoque del modelo para SERS⁸ presentado en (Santos y Boticario, 2011), implica aprovechar los aspectos semánticos que dejan por fuera los sistemas recomendadores tradicionales, incluso los referentes a la usabilidad, accesibilidad e interoperabilidad de los recursos en distintos escenarios. Para esto los autores plantearon tres requerimientos mínimos al diseñar este tipo de sistemas: el uso de un modelo de recomendaciones, el uso de una arquitectura orientada a servicios abierta y una interfaz de usuario para desplegar los resultados.

Uno de los enfoques de los sistemas hipermedia, busca facilitar la generación de recursos web adaptativos mediante la utilización de cuatro componentes básicos para alcanzar una arquitectura abierta. Estos cuatro componentes corresponden al modelo del dominio, el modelo de usuario, el modelo de adaptación/dominio⁹ y el motor de adaptación¹⁰. Dicha arquitectura se denomina AHA¹¹ y fue desarrollado por la Universidad de Eindhoven (De Bra y cols., 2003; Stash y De Bra, 2004).

Otro modelo de adaptación para las recomendaciones es el utilizado en las redes de aprendizaje, el cuál, permite aprovechar los métodos y la infraestructura de los siste-

⁸SERS, Acrónimo de *Semantic Educational Recommender Systems*, son sistemas recomendadores enfocados en los LMS y se caracterizan por su capacidad para guiar al estudiante en la plataforma de aprendizaje en ambientes de educación formales (Santos y Boticario, 2011).

⁹Modelo de adaptación/dominio: Es un archivo XML en el que se describen las reglas de adaptación para el modelo de usuario. Según (De Bra y cols., 2003), es realizado con una herramienta de autoría con la que se accede a los conceptos y relaciones y define los prerrequisitos del sistema.

¹⁰Motor de adaptación: Es la parte del sistema encargada de realizar las actualizaciones a partir de las descripciones dadas por el modelo de adaptación.

¹¹AHA, Acrónimo de *Adaptive Hipermedia Architecture*. Fue desarrollado para sistemas web adaptativos e inspirado por el modelo AHAM, el cuál a su vez, es una extensión del modelo de referencia de hipertexto de Dexter (De Bra y cols., 2003).

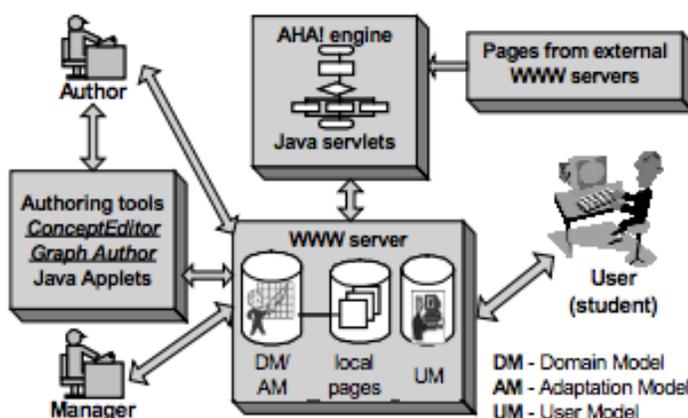


Figura 3.2: Esquema de arquitectura de AHA. Tomado de Stash y De Bra, 2004.

mas distribuidos para darle apoyo al desarrollo de las competencias de los estudiantes (Manouselis, Drachsler, Verbert, y Duval, 2012). Además, estas redes permiten monitorear el comportamiento del estudiante y ajustar el sistema a sus necesidades de aprendizaje.

3.3. Limitaciones de los algoritmos de recomendación

Una técnica común dentro de los sistemas recomendadores, consisten en la combinación de dos o más técnicas con el fin de mejorar el desempeño de la posible recomendación. De este modo, se busca compensar las deficiencias de una técnica, con las fortalezas de otra. Sin embargo, para decidir cuáles técnicas se pueden combinar, es necesario conocer las principales características de cada uno de los métodos utilizados para su construcción.

Se han realizado numerosos trabajos en los cuales se recopilan las características principales de las técnicas utilizadas para generar recomendaciones (Resnick y Varian, 1997; Burke, 2002; Adomavicius y Tuzhilin, 2005; Burke, 2007; Manouselis y Costopoulou, 2007; Ricci y cols., 2011; Manouselis y cols., 2011, 2012). Las principales técnicas para generar recomendaciones que mencionan estos autores son las siguientes:

- **Colaborativos:** Son aquellos sistemas que intentan generar las recomendaciones más acertadas basándose en las recomendaciones de otras personas que han tenido gustos o preferencias similares (Manouselis y cols., 2012). Un perfil de usuario de este tipo, consiste típicamente de un vector de items y un rating que aumenta continuamente con las interacciones de usuarios a través del tiempo (Burke, 2007).
- **Demográficos:** Se refiere a aquellos que usan la información demográfica de los atributos del perfil de un usuario para realizar su clasificación y asignarle las posibles recomendaciones de su interés (Burke, 2007).
- **Basados en utilidad:** Realiza las sugerencias tomando como base la utilidad de cada objeto para el usuario, por lo que el problema central consiste en cómo crear la función de utilidad para cada uno de ellos (Burke, 2002; Manouselis y Costopoulou, 2007).
- **Basados en conocimientos:** Realizan inferencias lógicas mediante reglas para determinar las preferencias del usuario. Es decir, posee información sobre cómo puede un item en específico, ajustarse a las necesidades de un usuario (Burke, 2002).
- **Basados en contenidos:** Utilizan una serie de descriptores iniciales tanto para los usuarios como para los ítemes que se van a recomendar; de modo que se puede usar esa información para predecir los intereses del usuario (Manouselis y Costopoulou, 2007), es decir, entrena el perfil del usuario basándose en las características de los items con los que ha interactuado (Burke, 2002).
- **Basados en preferencias:** Aquellos sistemas que intentan generar los posibles valores que darían los usuarios al evaluar aquellos ítemes que aún no han visto (Adomavicius y Tuzhilin, 2005)

3.4. Técnicas de hibridación

Las técnicas de hibridación se utilizan para compensar las debilidades de una técnica de recomendación con las fortalezas de otra; con el fin de mejorar la calidad de la recomendación que se va a presentar al usuario o solucionar algún problema como el “cold start”¹².

Existen numerosas técnicas, algunas de ellas sensibles al orden y otras que no lo son. Por ejemplo, las técnicas de pesos, mixtas, intercambio y mezcla de características no son sensibles al orden; por otro lado, las demás sí pueden verse afectadas por el orden de las entradas (Burke, 2002; Ricci y cols., 2011). Algunas de las técnicas comunes usadas para lograr la hibridación en dichos sistemas son:

- Pesos: Los resultados de varias recomendaciones se combinan en una sola respuesta.
- Intercambios: El sistema cambia de técnica de recomendación de acuerdo a la situación.
- Mixtos: Se dan los resultados de diferentes recomendadores al mismo tiempo.
- Combinación de características: Las recomendaciones de diferentes orígenes de datos se combinan en un único algoritmo de recomendación.
- Cascada: Un recomendador refina la salida de otro.
- Aumentar la características: La salida de un recomendador, se usa como característica de entrada de otro.
- Meta-nivel: El modelo aprendido por un recomendador, se usa como entrada para otro.

¹²“Cold Start”: Es un problema que se origina cuando ingresan nuevos usuarios al sistema y no se cuenta con información suficiente para generarle una recomendación.

La selección de la técnica apropiada para la hibridación de métodos, depende del tipo de información con el que vamos a trabajar y el objetivo final de realizar la combinación. Algunos autores, como R. Burke en (Burke, 2002), proponen que algunas combinaciones pueden resultar en un análisis redundante de los datos, o bien, pueden dar como resultado combinaciones que no son posibles o útiles para el usuario final. Es por este motivo que el análisis del contexto y los datos disponibles para el recomendador, juegan un papel crucial antes de iniciar cualquier diseño.

3.5. Herramientas de recomendación

Al igual que en otros ámbitos del desarrollo de software, la reutilización de componentes busca reducir el tiempo invertido en realizar implementaciones de algoritmos bien conocidos y estudiados; de modo que no se le reste tiempo a la implementación y refinamiento de las nuevas propuestas de algoritmos (Ekstrand, Ludwig, Konstan, y Riedl, 2011).

Existen múltiples librerías y APIs de recomendación que se encuentran implementadas bajo licencias OpenSource y están disponibles para trabajos académicos y comerciales. En estas implementaciones se pueden encontrar algoritmos asociados a recuperación de información, aprendizaje automático y generación de diversos tipos de recomendaciones. Algunos ejemplos de estas implementaciones son:

- The Lemur Project ¹³: Es un toolkit que ofrece múltiples herramientas para recuperación de la información con consultas, manipulación de estructuras en XML y mecanismos de sumarización, filtrado y categorización en las recomendaciones. Este proyecto fue desarrollado en conjunto por la Universidad de Massachusetts y Carnegie Mellon.

¹³Lemur Project, disponible en <http://www.lemurproject.org/>

- SLIM¹⁴: es una librería que ofrece métodos de recomendación del tipo top-n basados en modelos de dispersión lineal, para generar recomendaciones mediante filtros colaborativos y matrices de similitud.
- Apache Mahout¹⁵: es una librería para aprendizaje automático que ofrece una amplia gama de algoritmos como filtros colaborativos o de recomendación, medidas de similitud entre vectores, minería de patrones, regresiones, algoritmos evolutivos, entre otros.
- LensKit ¹⁶: Es una herramienta basada en Java que implementa algoritmos colaborativos y herramientas para medir su rendimiento. Además, permite administrar las configuraciones, modelar procesos y comparar nuevas implementaciones contra otras de algoritmos existentes mediante un API.
- myCBR¹⁷ : es una herramienta de recuperación basada en similitud y fue construida en conjunto por el centro de competencias CBR del DFKI en Alemania y la escuela de computación y tecnología de la Universidad del Oeste de Londres en Reino Unido.
- Duine¹⁸ : Es un framework que ofrece siete técnicas de predicción, entre ellas: razonamiento basado en casos, razonamiento basado en géneros, popularidad y filtros colaborativos
- PREA¹⁹ : es una aplicación en Java que ofrece comparaciones entre algoritmos para filtrado colaborativo; así como ofrece conjuntos de datos para experimentación.

¹⁴SLIM, disponible en <http://glaros.dtc.umn.edu/gkhome/slim/overview>

¹⁵Apache Mahout, disponible en <http://mahout.apache.org/>

¹⁶LensKit, disponible en <http://lenskit.grouplens.org/>

¹⁷MyCBR, disponible en <http://www.mycbr-project.net/>

¹⁸Duine, disponible en <http://www.duineframework.org/>

¹⁹PREA, acrónimo de "Personalized Recommendation Algorithms Toolkit", disponible en <http://prea.gatech.edu>

- WEKA²⁰: Es una colección de algoritmos de aprendizaje automático para tareas de minería de datos que incluye herramientas para realizar las tareas preprocesamiento de datos, clasificación, regresiones, clustering, reglas de asociación y visualización.
- Entre otros.

Estos API's y librerías son herramientas probadas que permiten a quienes las utilizan, reutilizar códigos funcionales y optimizados. De modo que agilizan los procesos de implementación de la tecnología en diferentes contextos de operación.

Para el ARHOA se utilizarán las definiciones de algoritmos provistos por estos API's tras una fase de preprocesamiento de los datos, específicamente el de Lenskit para aprovechar las adaptaciones para recomendación en entornos colaborativos. Mientras que para la construcción de los perfiles de curso, se utilizarán los algoritmos provistos por WEKA para manipulación de textos.

²⁰WEKA, acrónimo de "Waikato Environment for Knowledge Analysis "

4 Diseño del agente

En este capítulo se presentarán aspectos técnicos asociados al diseño del agente y su contexto de operación. Además, se discutirán aspectos como la arquitectura general del sistema, las necesidades especiales del diseño y finalmente, se presentará un modelo detallado del agente a construir.

El agente propuesto se ubicaría en la categoría de híbrido, ya que utilizará distintas técnicas de recomendación para generar sus resultados. Se emplea un recomendador basado en contenidos enlazado en cascada con un recomendador colaborativo; y finalmente se desplegará la información mediante una representación mixta con la recomendación por estilos de aprendizaje (Adomavicius y Tuzhilin, 2005; Burke, 2007).

Si se enfoca el diseño de la arquitectura según la estructuración típica de los sistemas recomendadores; se debería separar la línea de desarrollo en dos subproblemas: la definición del perfil del curso y generar la recomendación. Sin embargo, en el enfoque expuesto en SERS se añaden otros requerimientos para aprovechar otros aspectos semánticos de los entornos educativos, como el uso de una arquitectura orientada a servicios abierta y una interfaz de usuario para desplegar los resultados.

El uso de herramientas como diseño instruccional y los modelos de aprendizaje basados en atributos o competencias, ofrecen, mediante prosa, una descripción de las metas de aprendizaje que se esperan alcanzar con cada unidad. Sin embargo, actualmente en el TEC no existe un modelo de atributos estandarizado para las carreras, por lo que los esfuerzos para crearlo se centran en algunas escuelas que se encuentran en un plan piloto para implementarlas dentro de su curriculum.

Desde la perspectiva del agente, la información facilitada por el TEC Digital se estaría aprovechando como una base de datos de la que se pueden extraer textos del DI, MoGa y TEA; y además, se desea aprovechar la información histórica de búsquedas similares, para refinar los procesos de recomendación.

4.1. Metodología

La metodología de trabajo se puede abordar desde distintas perspectivas debido a las características de los datos y las tareas que se deben realizar.

Primero se puede ver como un problema de minería de datos (DM), debido a que se debe analizar y procesar altos volúmenes de datos semiestructurados y tomar decisiones a partir de esta información.



Figura 4.1: Tareas relacionadas con el enfoque del tema de investigación

Dado que es necesario identificar frases útiles para realizar búsquedas a partir de una serie de documentos textuales, se puede considerar un problema de procesamiento de lenguaje natural (NLP). Estos documentos contienen errores ortográficos y contenidos que no son significativos para describir cada actividad, de modo que hay que aplicar una serie de tareas propias de los NLP para elegir sólo aquellos textos de utilidad.

Si nos centramos en la forma de recuperar los ítems a evaluar, se deben realizar conexiones utilizando agentes HTTP para recuperar los OA que van a ser analizados. Estos agentes deben conectarse a repositorios de OA usando distintas interfases y recuperan-

do recursos en distintos formatos.

Finalmente, una vez recuperados los recursos y descritas las actividades, es necesario sugerir aquellos OA que mejor se adapten a las descripciones dadas para las actividades del curso; de modo que se puede ver como un problema relacionado con sistemas recomendadores (RecSys).

Al ser una investigación en la que se relacionan distintas áreas del conocimiento, elegir una metodología de trabajo única sería insuficiente o dejaría por fuera aspectos que podrían abordarse mejor siguiendo otro modelo; por lo que se propone tropicalizar las metodologías de trabajo para aproximar una mejor solución.

Se utilizará como base el modelo de los sistemas recomendadores al descomponer el problema principal en dos subproblemas: construcción del perfil y generar la recomendación. De este modo, al trabajar cada uno de estos subproblemas por separado, se aplicarán las tareas específicas relacionadas con otras áreas del conocimiento según sea necesario; de modo que durante el desarrollo de la investigación y del prototipo se encontrarán referencias y tareas que relacionadas con distintos enfoques de trabajo como se puede apreciar en la figura 4.1.

4.2. Arquitectura general

Los procesos involucrados en la adquisición, preprocesamiento y análisis de datos, son elementos fundamentales para alcanzar el éxito en una implementación que está relacionada con la identificación de patrones y/o necesidades de un usuario con base en la información disponible.

Existen trabajos que se han enfocado en la adquisición de modelos de usuario para plataformas de elearning con entornos adaptativos; de modo que se puedan ajustar los ambientes a las necesidades del usuario a partir de una serie de reglas de adaptación y un banco de recursos (Boticario y Santos, 2007); por lo que este trabajo se centrará en la recomendación de un posible banco de OA para las actividades de un curso en particular. Antes de definir una arquitectura para la aplicación, es necesario identificar la forma en

que se relacionan los conceptos que se van a ver involucrados en el diseño. Es por esta razón, que en la figura 4.2 se muestra a nivel macro la relación que existe entre los elementos definidos mediante el DI, MoGa y TEA.

Cómo se puede apreciar en este diagrama, cada una de las actividades del DI pueden tener uno o más objetivos definidos; los cuales, responden a diversas metas. Estas metas son las llamadas competencias o atributos que el estudiante debería adquirir al realizar dicha actividad.

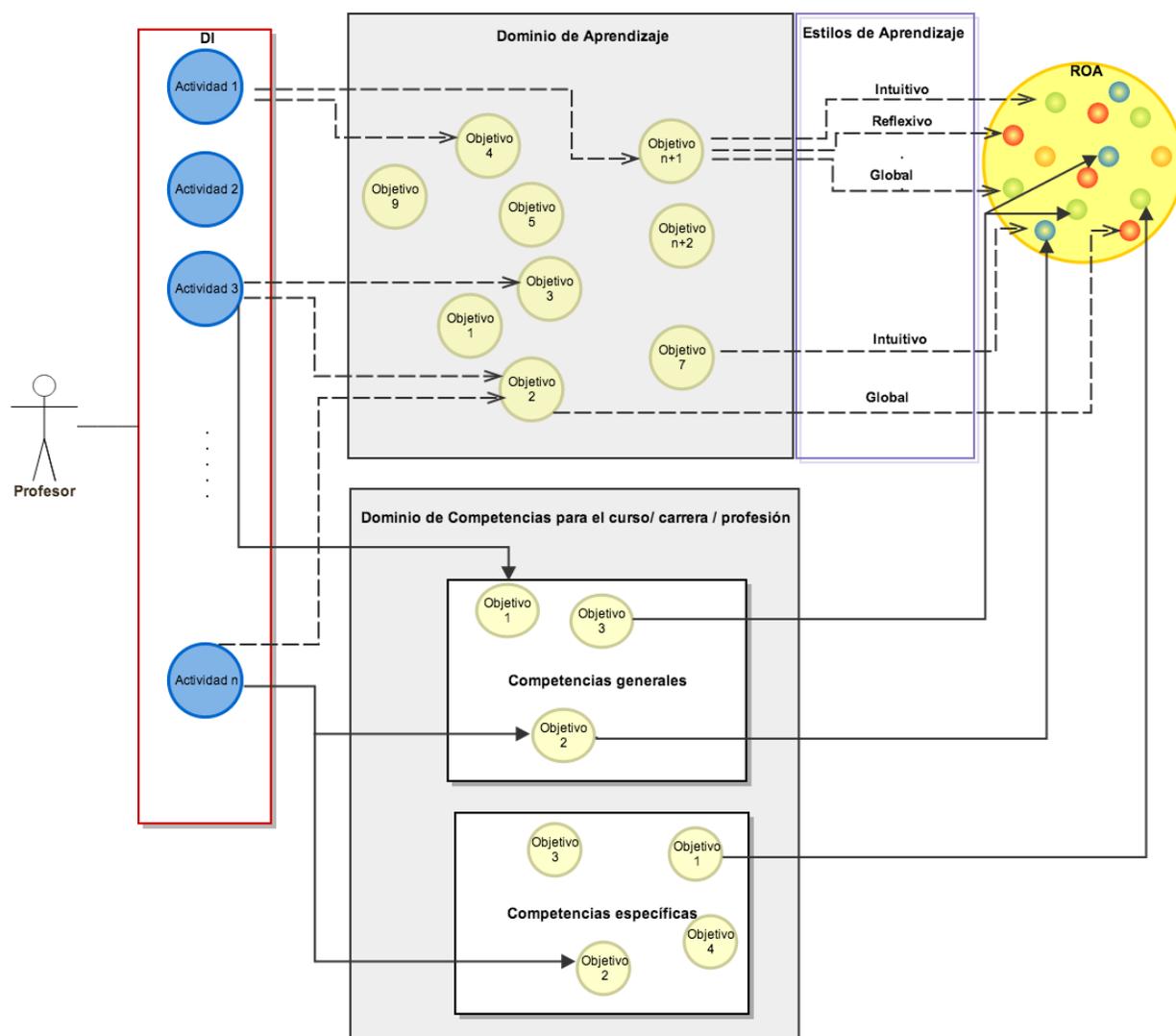


Figura 4.2: Relación entre conceptos de DI, MoGa, TEA y OA

Según las TEA, un estudiante puede tener inclinaciones hacia uno o más estilos de aprendizaje, los cuales varían a lo largo de los estudios. Por este motivo, existen múltiples caminos para alcanzar una misma meta; y por ende, la preferencia hacia un OA u otro con el fin de mejorar la experiencia del aprendizaje puede variar de acuerdo al estilo que se esté buscando satisfacer.

Con esta información, se espera que el agente sea capaz de analizar aquellos OA disponibles en un repositorio que tengan mayor grado de coincidencia con las descripciones brindadas en los documentos y es por esto, que en la sección 4.2.1 se presentará mayor detalle sobre las características de cada uno de ellos.

Estos conceptos representan la información que se podrá obtener desde la plataforma del TEC Digital, sin embargo, existe información adicional que es necesaria y no se encuentra solamente en la plataforma; por lo que la figura 4.3 muestra a nivel conceptual, donde se ubican los distintos orígenes de datos para el agente.

Cómo se puede apreciar en la figura, se tienen tres orígenes principales de datos:

- Servidores del Departamento de Admisión y Registro del TEC: Ofrecen una interfaz para consumir los servicios web del Departamento de Admisión y Registro (DAR); de modo que la plataforma del TEC Digital puede implementarlos para consultar información. De estos servicios web se espera obtener la lista de cursos que son equivalentes para un DI.
- Plataforma del TEC Digital: Ofrece la plataforma del LMS y posee las aplicaciones del MoGa y el DI. En esta plataforma funciona el paquete ws-servicios que consume la interfaz disponible de los servicios web del DAR y se espera que contenga la descripción que alimentará los estilos de aprendizaje.
- Servidores del ROA: Contiene los OA disponibles en un repositorio y se ofrecen mediante un API para consumo.

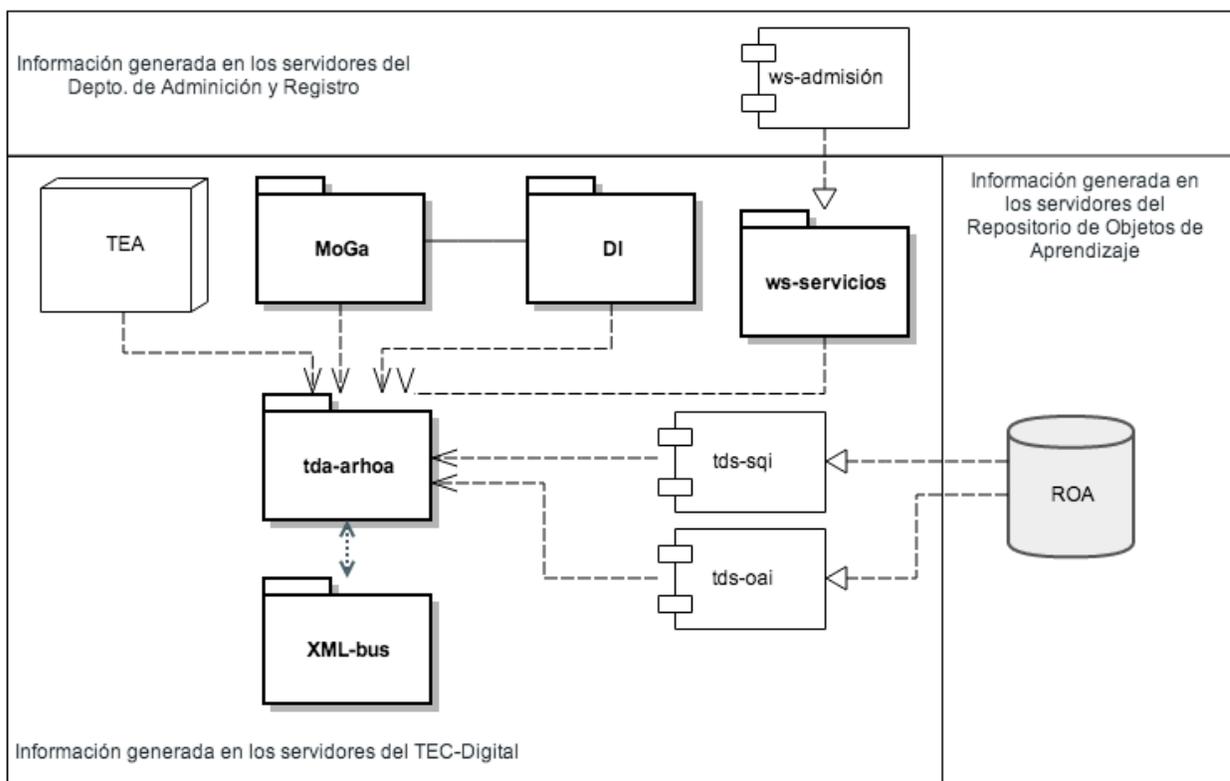


Figura 4.3: Diagrama de Interacción de los orígenes de datos

Cada uno de estos elementos retorna distintos tipos de información y en distintos formatos; por lo que a continuación se procederá a detallar los tipos de datos con los que se trabajará.

4.2.1. Tipos de datos

La mayoría de los datos con los que se trabajarán están dados mediante campos de texto en una base de datos o en formato XML; por lo que la fase de preprocesamiento de datos es crítica para lograr la interacción y reducir ambigüedades textuales.

4.2.1.1. DI y MoGa

Como se mencionó previamente en la sección 2.4 sobre el diseño instruccional; éste consiste en un planeamiento detallado de las lecciones que se impartirán. En este documento se incluyen elementos que van desde el propósito del curso, hasta los conocimientos previos mínimos que debe tener su población meta; por lo que es dado mediante lenguaje natural por el profesor. Un ejemplo de esto se puede apreciar en la figura 4.4; que muestra una fracción de la planificación semanal de un curso.

Semana	Título	Objetivos de aprendizaje	Contenidos de los módulos	Actividades de aprendizaje	Medios y materiales que se utilizarán	Evaluación de los aprendizajes
1	Módulo introductorio	<p>El participante:</p> <ul style="list-style-type: none"> Recibirá la descripción de la metodología de trabajo para el curso. Obtendrá un panorama general de las herramientas OpenSource disponibles para apoyo a la docencia. (Conceptual) Podrá describir el objetivo general del tecDigital. (Conceptual) 	<ul style="list-style-type: none"> -Descripción general del curso -Explicar la metodología de trabajos. -Breve introducción al concepto de plataformas de apoyo a la docencia / plataformas educativas (mencionar moodle, dotim, blackboard, sakai). -¿Qué es el tecDigital? ¿Qué hacemos? 	Trabajo individual: Lectura	<p>Documento de presentación del curso</p> <p>Se utilizará el tecDigital como plataforma para postear los trabajos de cada estudiante.</p> <p>Lectura opcional de material: "Usability in e-Learning Platforms: heuristics comparison between Moodle, Sakai and dotLRN" (http://www.litkora.com/ja/papers/Usability%20in%20e-Learning%20Platforms%20Heuristics%20comparison.pdf)</p>	Cuestionario sobre plataformas.
2	Introducción a la arquitectura de Openacs	<p>El participante:</p> <ul style="list-style-type: none"> Analizará la arquitectura de trabajo según Openacs. (Conceptual) Organizará a un alto nivel la distribución que deben seguir las aplicaciones diseñadas para la plataforma del tecDigital. (Conceptual) 	<ul style="list-style-type: none"> -Modelo de capas de Openacs -Concepto de paquete -Modos de visualización de la GUI. -Perfiles de usuario en openacs 	Lectura individual.	<p>Plataforma del tecDigital para postear las unidades de aprendizaje.</p> <p>Lectura de unidad de aprendizaje con información sobre openacs, roles de usuario, etc (Documento por crear)</p>	NA

Figura 4.4: Fragmento de la planificación semanal en un DI para un curso bimodal de programación en TCL.

La aplicación para DI desde la cuál se tomará la información, sigue un esquema de almacenamiento similar al del documento. En el DI se organizan los contenidos por sesión con la descripción de las actividades a desarrollar . Esta aplicación cuenta con un sistema visual de semáforos para indicar el avance en el desarrollo de la planeación del curso como se muestra en la figura 4.5, por lo que podría darse alguna situación en la que un usuario solicite una recomendación para un DI con información insuficiente.

Sesiones

Sesión	Título	Objetivos de aprendizaje	Módulos de contenidos	Actividades de aprendizaje	Medios - Materiales a utilizar	Evaluación de los aprendizajes
1	●●●● Título...	●●●○ Objetivos...	●●●○	●●●○	●○○	●○○
2	○○○	○○○	○○○	○○○	○○○	○○○
3	○○○	○○○	○○○	○○○	○○○	○○○
4	○○○	○○○	○○○	○○○	○○○	○○○
5	○○○	○○○	○○○	○○○	○○○	○○○
6	○○○	○○○	○○○	○○○	○○○	○○○
7	○○○	○○○	○○○	○○○	○○○	○○○
8	○○○	○○○	○○○	○○○	○○○	○○○
9	○○○	○○○	○○○	○○○	○○○	○○○
10	○○○	○○○	○○○	○○○	○○○	○○○
11	○○○	○○○	○○○	○○○	○○○	○○○
12	○○○	○○○	○○○	○○○	○○○	○○○

Figura 4.5: Sistema de semáforos para la GUI de DI

En el caso del MoGA, se conecta con el DI mediante una relación existente con las actividades de aprendizaje. Estas actividades se asocian a instrumentos que nos permiten seguir las evidencias de los estudiantes para una determinada competencia y también están descritas mediante lenguaje natural en un campo de tipo texto.

4.2.1.2. TEA

Los estilos de aprendizaje consistirán en una serie de descriptores para cada uno de los estilos descritos por Felder y Silverman para los estudios en ingeniería (Felder y Silverman, 1988). Además, para realizar la clasificación de los OA, se deberá contar con la definición de distintos niveles de afinidad de cada estilo con respecto a los distintos tipos de recursos que se podrían obtener del repositorio.

Estos criterios de afinidad se podrán brindar mediante etiquetas que describan los grados de pertenencia de los elementos con respecto a un subconjunto u otro.

4.2.1.3. Equivalencias entre cursos

Utilizar únicamente un documento del DI y su MoGa asociado para establecer las palabras clave relevantes para un curso, deja un grado de confiabilidad¹ y de soporte² muy bajo en las reglas de asociación³ que se pueden identificar dentro de un conjunto de datos.

Ahora bien, ¿qué significa que un curso posee un curso equivalente para el agente?; esto significa que puede existir otro DI y otro MoGA que se supone que cumple con el mismo conjunto de requisitos de aprendizaje que el curso que está en proceso de revisión. Por este motivo, sus descriptores se pueden utilizar para validar los elementos clave que se encontraron en los documentos evaluados al inicio y con esto, se pueden balancear mejor los porcentajes de confiabilidad y los grados de soporte de las reglas de asociación. En otras palabras, mientras más información existe, mejor va a ser la aproximación.

La información en la que se indica si un curso tiene otros equivalentes es administrada por el DAR y puede ser accedida a través de servicios web. De este modo, la información potencial que puede ser aportada por los cursos equivalentes en los distintos planes de estudio excede las 6866 equivalencias, reportadas para 3173 cursos.

Otro ejemplo de posible utilización de estos datos se aprecia en la figura 4.6, donde se está trazando un mapa de las relaciones entre los cursos equivalentes según la información que se puede recuperar en los servicios web del DAR. En esta figura se pueden apreciar grupos densos de nodos que se conectan y representan aquellos cursos que

¹Grado de confiabilidad: Porcentaje del total de transacciones o grupos de datos que cumplen una regla.

²Grado de soporte: Probabilidad de que dado un dato o evento X, se llegue a un dato o evento Y.

³Reglas de Asociación: Descubrir reglas de asociación o aprenderlas, consiste en el proceso de identificar reglas de dependencias entre distintos grupos para predecir la ocurrencia de un objeto o ítem, basándonos en la ocurrencia de otros (Gorunescu, 2011) .

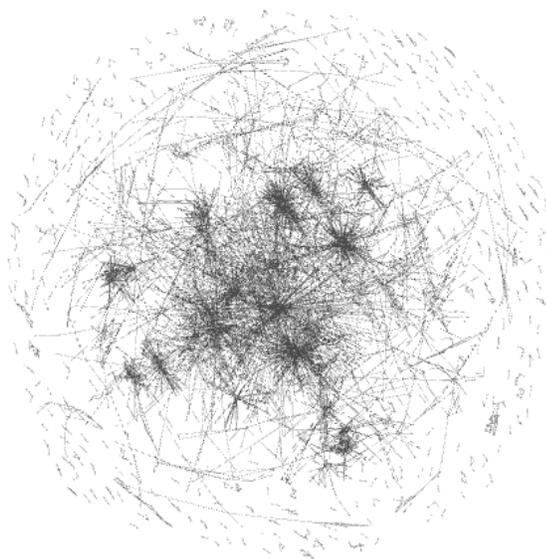


Figura 4.6: Representación de cursos equivalente con grafo de expansión generado mediante algoritmo Yifan-Hu Multinivel con 3 capas.

tienen un mayor número de cursos equivalentes identificados. Este comportamiento nos lleva a otro detalle que se debe tomar en cuenta al evaluar los cursos equivalentes según el DAR, que es la existencia de cursos que son comunes a más de a una carrera.

Algunos de los cursos que son considerados como equivalentes pertenecen a contextos muy distintos (por ejemplo, tomando en consideración los cursos de formación humanística y las consideradas como culturales y deportivas). Estos cursos comparten características mínimas, o casi nulas entre sí, y si se consideran con la misma prioridad que los documentos que describen el curso que vamos a analizar mediante el agente, podrían insertar ruido en la toma de decisiones.

En las figuras 4.7 y 4.8 se aprecia mejor el impacto que pueden tener sobre la toma de decisiones si se les aplica el mismo peso que a los documentos del DI y del MoGa que estaría evaluando el agente. Bajo las condiciones actuales de trabajo para el agente, esta lista no es significativa, pero la existencia de 228 cursos considerados como equivalentes

	materia character varying	cantidad de equivalentes bigint
1	FH1000	228
2	AM2400	64
3	AM2700	63
4	AM2600	63
5	AM2500	63
6	SE1400	56
7	IC5001	54
8	IC4003	53
9	IC4000	52
10	IC5000	52

Figura 4.7: Muestra de datos con agregación de cursos equivalentes

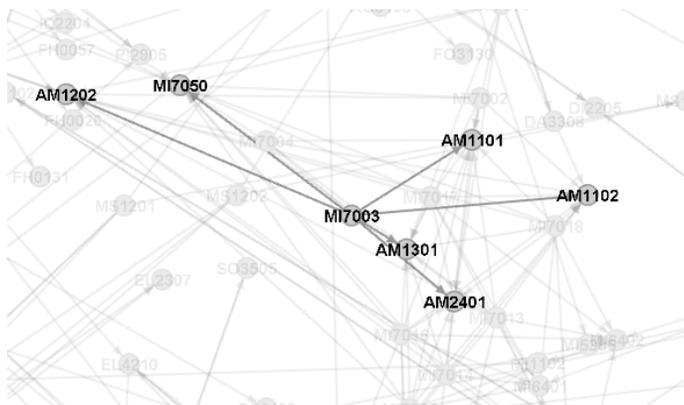


Figura 4.8: Fragmento de representación de las relaciones de equivalencia mediante grafos dirigidos

para formación humanística con sus respectivos documentos de DI y MoGa podrían causar un caos al intentar identificar un patrón entre ellos si no existe un contexto común que los esté delimitando.

4.2.1.4. Consultas al ROA

Como ROA fuente para realizar las consultas se eligió el repositorio Ariadne, el cuál, permite realizar búsquedas cruzadas con otros repositorios que pertenecen a GLOBE como LORNET, MERLOT y LACLO; además, cuenta con miles de OA disponibles que pueden descargarse en múltiples formatos.

Este repositorio da soporte principalmente a IEEE-LTSC LOM⁴, pero también soporta Dublin Core y el ISO/IEC MLR y puede realizar la conversión entre tipos de metadatos de forma automática⁵. Otra de sus ventajas, es que cuenta con una herramienta de búsqueda

⁴La especificación del borrador está en el sitio ltsc de la IEEE bajo la ruta http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf, en ese mismo archivo se presenta una tabla de equivalencias con respecto al estándar de Dublin Core

⁵Información disponible en <http://www.ariadne-eu.org/content/technologies>

da denominada "ARIADNE finder" que permite realizar búsquedas sobre el repositorio y esconde los protocolos y estándares de la vista del usuario.

Para dar soporte a los derechos de autor, este ROA incluye descriptores asociados a los costos de uso de los OA, tipos de documentos, formatos, usos pedagógicos del OA, contexto, nivel para el que fue construido y niveles de dificultad, duración, nivel de interactividad, entre otros (Duval y cols., 2001). Estos recursos disponibles en el repositorio vienen en forma de juegos, fotos, videos y textos que están dados por recursos como archivos .doc, .html, .pdf, .ppt, .txt y otros formatos multimedia.

4.2.1.5. Historiales de búsquedas

Dado que de momento no se pretende crear un ROA que se conecte con el consorcio; no se pretende crear una base de datos nueva con los OA descargados después de cada recomendación. Sin embargo, sí se pueden reutilizar los metadatos de los OA elegidos por el profesor después de cada búsqueda.

Los metadatos que describen los OA que el profesor ha elegido como útiles para su lección, pueden ampliar o refinar el conjunto de metadatos a partir de los cuales se realizarán otras búsquedas para cursos equivalentes.

Cuando se cuente con más información histórica, será posible crear perfiles descriptores por carreras y cursos, de modo, que se podrían crear alianzas estratégicas para fortalecer los nodos o cursos que conectan múltiples disciplinas. Así como buscar estrategias que creen nuevos vínculos en aquellas carreras donde aún no se haya explotado el trabajo interdisciplinario.

4.2.2. Identificación de las tareas de adaptación

La información para refinar los perfiles de búsqueda viene dada en forma de metadatos expresados mediante diferentes formatos de encapsulación para XML⁶, los cuales se pueden expandir mediante distintos métodos de codificación de orden (Tatarinov y cols.,

⁶XML, Acrónimo de *eXtensive Meta Language* por sus siglas en inglés

2002) para convertirlos en operaciones sobre una base de datos relacional. Esta información puede ser procesada mediante algoritmos que permita explorar la estructura de árbol con el fin de seleccionar de múltiples elementos mediante análisis de patrones como se sugiere en (Al-Khalifa y cols., 2002) y así obtener los textos que serán reducidos mediante algoritmos de minería de datos y recuperación de información para retornar un porcentaje de certeza en la recomendación.

Al trabajar con colecciones textuales reducidas para la generación del perfil de cada curso y enfocarnos en el diseño de una solución de propósito general para cursos de diferentes contextos, el uso de diccionarios de entrenamiento no está siendo contemplado como una opción viable en esta fase de investigación. Sin embargo, no se descarta la construcción o integración posterior de diccionarios enfocados en dominios de conocimiento específicos.

El proceso de selección de los recursos candidatos está relacionado directamente con la calidad y la precisión de la descripción de los contenidos de los OA y las actividades del DI; de modo que será necesaria una o más fases de extracción y adaptación de las dimensiones de los elementos involucrados con el fin de reducir la cardinalidad de las representaciones vectoriales y mejorar los tiempos de respuesta de los algoritmos aplicados.

Además, dado que los metadatos que describen a un OA proviene de repositorios externos y deben ser obtenidos mediante consultas a servicios web, se intentará minimizar la cantidad de consultas que se dispararán para recuperar información y de ser posible, se espera refinar las búsquedas con cada selección. Este comportamiento al refinar las búsquedas es típico de los sistemas recomendadores colaborativos, ya que la interacción de los usuarios con el sistema determinan el flujo de las adaptaciones del perfil.

4.2.3. Adquisición de datos del curso

Según (Webb, Pazzani, y Billsus, 2001), la forma en que se construye un modelo de usuario puede responder a distintos propósitos:

- Identificar los procesos cognitivos que ocultan las acciones del usuario
- determinar las diferencias entre las habilidades de un usuario normal y un experto
- determinar los patrones o preferencias ocultas en el comportamiento del usuario
- determinar las características del usuario.

En nuestro caso, la “adquisición del modelo del usuario” del agente estará asociada al curso para el que fue definido el DI, y se contruye con el fin de determinar las principales características y necesidades que se deben satisfacer.

Dado que los documentos de DI y del MoGa vienen dados en lenguaje natural, se debe

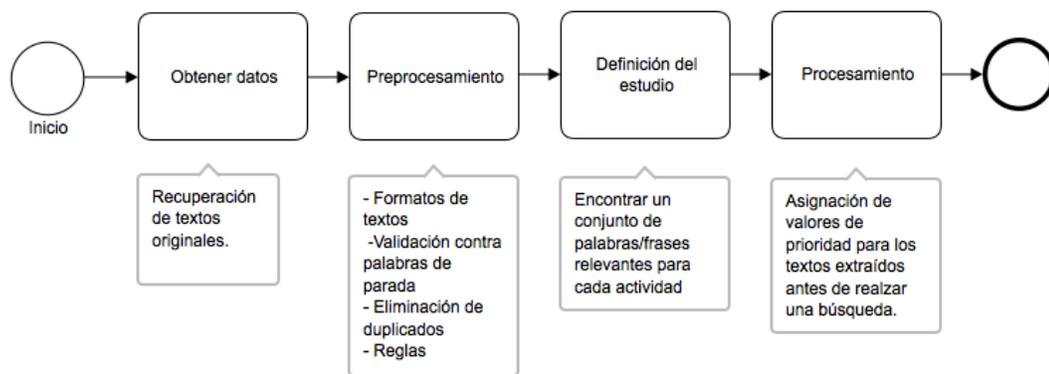


Figura 4.9: Extracción de frases para un perfil de curso usando un enfoque de minería de datos.

buscar un mecanismo que simplifique los procesos de búsqueda y procesamiento de textos. Por este motivo, se utilizarán técnicas para reducir los bloques de texto expresados en lenguaje natural a un subconjunto de palabras que sean significativas para describir el curso y más útiles a nivel computacional; específicamente mediante el uso de palabras de parada en español y reglas para reducir las expresiones.

Enfocando este problema desde el punto de vista de la minería de datos, como se puede apreciar en la figura 4.9, las fases para este proceso de construcción del perfil serán las siguientes:

- Obtención de datos: Se recuperan los textos disponibles para describir a las actividades de un DI.
- Preprocesamiento : Consiste en estandarizar el formato de los textos obtenidos como posibles palabras claves. Se validan los textos contra un diccionario de palabras de parada y se convierten a mayúsculas o minúsculas con el fin de descartar palabras duplicadas de la lista de candidatas antes de evaluarlas.
- Definición del estudio: Determinar los subconjuntos de palabras clave relevantes para cada una de las actividades definidas para un curso en particular
- Procesar la información: Una asignación de pesos le dará la relevancia a cada uno de los términos para elegir aquellos que ameriten ser incluidos dentro de las opciones de búsqueda del sistema recomendador.

Una vez generado y validado el conjunto de palabras clave se podrá usar esta información para construir un perfil básico del curso que se podría representar como se muestra en el ejemplo de la tabla 4.1, la cuál muestra la frecuencia de aparición de términos en la descripción textual de cada actividad. La matriz interior representa la cantidad de veces que aparece cada término y está agrupando los términos de cada actividad del DI en un único vector que la representa; por lo que posteriormente será sometida a un proceso de reducción para convertirla en una estructura de datos más representativa y con menor impacto computacional.

Visto de un modo simplificado, cada actividad es un modelo con múltiples atributos por satisfacer que están determinados por los contenidos textuales de los documentos que las describen. Para esto, es necesario evaluar la relevancia de cada uno de sus atributos y su impacto sobre otras actividades que pertenecen al mismo vecindario, o colección de actividades para un DI. La agregación de todos estos descriptores darán forma a un perfil general del curso, de modo que pueda compararse su descripción contra la de otros documentos equivalentes.

		Términos					
		t_1	t_2	t_3	...	t_{n-1}	t_n
Actividades	Act. 1	1	1	0	...	8	1
	Act. 2	4	1	1	...	1	1
	Act. 3	0	1	1	...	1	0
	Act. 4	2	0	0	...	2	1
	Act. 5	2	1	3	...	1	2
	Act. m

Tabla 4.1: Perfil básico de curso esperado, sin la agregación de actividades

	Términos						
	t_1	t_2	t_3	...	t_{n-2}	t_{n-1}	t_n
Curso 1	22	10	40	20	...	8	12

Tabla 4.2: Perfil general esperado de un curso con agregación de términos de las actividades

4.2.4. Generación de la recomendación

Determinar si un OA es relevante o no para una actividad específica, puede ser una decisión ambigua; ya que depende del contexto y del criterio de quien esté eligiendo el conjunto de actividades. Además, utilizar únicamente el soporte brindado por los documentos originales del DI y del MoGa del curso, sería restringir el conjunto de datos disponibles en otros contextos que pueden tener relación directa con las actividades; por lo que se reducirían los niveles de confiabilidad debido a la escasez de información. Como parte de la información obtenida de la base de datos, se pueden identificar al menos cinco categorías o grupos de pertenencia para las actividades recuperadas en el momento de intentar generar una recomendación para un curso:

- La actividad en evaluación (C1).
- Actividades pertenecientes al mismo curso (C2), pero que no son la que estamos evaluando en el momento.
- Actividades descritas en cursos equivalentes que tienen similitudes con la actividad en evaluación (C3).
- Actividades descritas en cursos equivalentes que tienen similitudes con otras actividades del DI, pero con poca similitud con la actividad que está en evaluación (C4).
- Pertenecientes a cursos equivalentes, pero que no tienen similitud con ninguno de los descriptores del DI (C5). En este caso, se consideraría que la actividad descrita es específica del dominio del curso equivalente; por lo que no se tomará en cuenta para evaluar los descriptores.

Para cada una de estas posibles categorías se desea determinar el grado de influencia que será utilizado al calcular la función de peso de cada una de las palabras clave.

4.2.4.1. Relevancia de términos

Según (Bramer, 2007), una forma común de calcular los pesos consiste en contar la cantidad de apariciones de un término en un documento; otro enfoque consiste en utilizar variables binarias para indicar la presencia o ausencia del mismo dentro de documento y con esta información, calcular sus distancias mediante modelos de espacio vectorial. El algoritmo denominado TF-IDF⁷ es una herramienta probada y ampliamente difundida para el análisis de textos en uno o múltiples documentos. Este algoritmo se basa en el principio de que los términos más comunes tendrán mayor frecuencia de aparición en la colección de documentos, por lo que es una medida que expresa cuán relevante es una palabra para un documento dentro de una colección.

La estimación por TF-IDF se da en término de dos variables: la frecuencia de apariciones de una palabra sobre un documento y la medida inversa de la frecuencia de aparición en los documentos disponibles.

La frecuencia relativa se calcula contra el total de apariciones de un término sobre el documento y existen distintos tipos de ajustes para calcularla junto con el IDF. Por ejemplo, en (Bramer, 2007), el valor de la frecuencia está dado por la cantidad de veces que aparece un término j en documento i . Mientras que el IDF, responde al $\log_2 \frac{n}{n_j}$, donde n_j es el número de documentos que contienen el término t_j y n es el total de documentos. Por otro lado, en (Witten, Frank, y Hall, 2011) se propone la utilización de un enfoque logarítmico para normalizar el vector a una misma distancia euclideana, como se muestra en la siguiente fórmula para el IDF :

$$IDF(t) = \begin{cases} 0 & \text{si } \text{frec}(t,d)=0 \\ -\log_2 \frac{|d|}{|d_t|} & \text{en otro caso} \end{cases}$$

Otro ejemplo de ajuste es el propuesto en (Han, Kamber, y Pei, 2006), en este ajuste, se aplican logaritmos para obtener la frecuencia de aparición relativa del término sobre

⁷TF-IDF, acrónimo de "term-frequency / inverse- document-frequency".

un documento (según los autores, esta fórmula está basada en el sistema SMART⁸ de la Universidad de Cornell). La fórmula utilizada para calcular el TF-IDF está dado por :

$$TF(t, d) = \begin{cases} 0 & \text{si frecuencia de apariciones} = 0 \\ 1 + \log(1 + \text{frec}(t, d)) & \text{en otro caso} \end{cases}$$

$$IDF(t) = \begin{cases} 0 & \text{si } \text{frec}(t, d) = 0 \\ \log \frac{1 + |d|}{|d_t|} & \text{en otro caso} \end{cases}$$

En este ejemplo, d_t es el número de documentos que contienen el término buscado y d representa al total de documentos disponibles. Además, $\text{frec}(t, d)$ se reemplaza por $\log t$, donde t es la cantidad de apariciones del término sobre el documento. En cualquier caso de ajuste deseado, la fórmula final para el TD-IDF estará dada por:

$$f_{td-idf}(t, d, D) = tf(t, d) \times idf(t, D)$$

Usando esta fórmula, podemos pasar del modelo de curso presentado en la figura 4.2 a una representación basada en la en cuantificadores de relevancia dentro de un contexto. Con esta información generada a partir del análisis del total de descriptores del curso, podemos establecer la importancia que tiene cada término con respecto a los demás de su dominio. Además, estos valores obtenidos para cada documento se pueden distribuir entre los descriptores de sus actividades; de modo que podemos ordenar la prioridad que tiene cada metadato específicamente para las actividades desde una perspectiva global de los documentos y manteniendo la prioridad que tiene definida dentro del curso, como se muestra en la figura 4.3.

⁸SMART; acrónimo de "System for the Mechanical Analysis and Retrieval of Text". Es un sistema de recuperación de información desarrollado por la universidad de Cornell, muchos de los conceptos de IR se desarrollaron durante su investigación, entre los que se puede incluir los modelos de espacio vectorial, retroalimentación de relevancia y las clasificaciones de Roccio.

		Términos												
		t_1	t_2	t_3	...	t_{n-1}	t_n	t_{n+1}	t_{n+2}	...	t_m	t_{m+1}	...	t_k
Act. del curso	Act. 1	0.0003	1.03	0	...	0.09	0.15	0	0	...	0	0	...	0
	Act. 2	4.000	0.893	1.4	...	0.3341	0.0004	0	0	...	0	0	...	0
	Act. n
Equivalente 1	Act. 1	0	0	0	...	3.42323	0.2342	0	3.5233	...	0.55	0	...	0
	Act. 2	0	0	2.3234	...	0.234	1.3234	0.663	0.2342	...	0.2342	0	...	0
	Act. 3	2.742	0	0	...	0	0	1.865	1.33	...	1.6344	0	...	0
	Act. m
Equivalente k	Act. 1	1.342	1.333	0	...	8	0.67	0	0	...	0.4289	0	...	0
	Act. 2	2.6634	1.40	0.3	...	0	0.342	1.12	0.53	...	0.34	0	...	0
	Act. k

Tabla 4.3: Perfil de cursos esperado con los cursos vecinos.

4.2.4.2. Recomendación por descriptores

Al intentar elegir los descriptores para una actividad específica, podemos buscar solo aquellos que tenemos registrados, o bien, podemos aprovechar los que nos podrían aportar otras actividades descritas en cursos equivalentes que tengan similitudes con las actividad que estamos evaluando.

Existen múltiples algoritmos utilizados para determinar la similitud entre documentos, por ejemplo la similitud por correspondencia de coseno, la correlación de Pearson, el coeficiente de Jaccard y el coeficiente de traslape, entre otros. Estas medidas para comparar documentos no solo se pueden aplicar a términos específicos, sino que pueden ajustarse a familias de palabras sobre las que se le han aplicado algoritmos de "stemming"⁹ con el fin de reducir la cantidad de términos que los representan (Bramer, 2007).

Para la comparación de dos documentos utilizando la correlación de coseno, se busca determinar el ángulo que representa la similitud entre un documento d_1 representado en v_1 y el documento d_2 , representado mediante v_2 . Para determinar la distancia de sus

⁹Stemming: son algoritmos que reducen las variables morfológicas de las palabras en el documento para agruparlas bajo una misma raíz que las represente. Su utilización puede ser muy efectiva para reducir el número de palabras que representan al documento, pero puede incurrir en errores semánticos como agrupar términos que no tengan relación (Bramer, 2007).

ángulos, se puede utilizar la fórmula presentada en (Han y cols., 2006), donde el producto de $v_1 \cdot v_2$ corresponde a la evaluación del producto vectorial de v_1 y v_2 ; mientras que $\|v_1\|$ y $\|v_2\|$ representan la magnitud de los vectores. De modo que:

$$v_1 \cdot v_2 = \sum_{i=1}^n v_{1i}v_{2i}$$

$$\|v_i\| = \sqrt{v_i \cdot v_i}$$

$$sim(v_1, v_2) = \frac{v_1 \cdot v_2}{\|v_1\| \|v_2\|}$$

$$sim(v_1, v_2) = \frac{\sum_{i=1}^n v_{1i}v_{2i}}{\sqrt{v_1 \cdot v_1} \sqrt{v_2 \cdot v_2}} = \frac{\sum_{i=1}^n v_{1i}v_{2i}}{\sqrt{\sum_{i=1}^n v_{1i}^2} \sqrt{\sum_{i=1}^n v_{2i}^2}}$$

Al utilizar este modelo de similitud, si se obtiene un coseno de cero significa que los documentos no tienen similitudes entre sus términos mientras que un valor de 1 indica que son idénticos, y dado que no hay valores negativos en los vectores, el ángulo que se construye no será mayor a 90 grados. Aplicando esta fórmula a un ejemplo concreto, sean $v_1 = [1, 2, 3]$ y $v_2 = [1, 2, 6]$, tenemos que:

$$\|v_i\| = \sqrt{v_i \cdot v_i}$$

$$\|v_1\| = \sqrt{v_1 \cdot v_1}$$

$$= \sqrt{\sum_{i=1}^n v_{1i}v_{1i}}$$

$$= \sqrt{\sum_{i=1}^n v_{1i}^2}$$

$$= \sqrt{\sum_{i=1}^3 v_{1i}^2}$$

$$= \sqrt{1^2 + 2^2 + 3^2}$$

$$= \sqrt{1 + 4 + 9}$$

$$= \sqrt{14}$$

$$\|v_2\| = \sqrt{1^2 + 2^2 + 6^2} = \sqrt{1 + 4 + 36} = \sqrt{41}$$

$$v_1 \cdot v_2 = \sum_{i=1}^n v_{2i}v_{1i} = 1 * 1 + 2 * 2 + 3 * 6 = 1 + 4 + 18 = 23$$

$$sim(v_1, v_2) = \frac{v_1 \cdot v_2}{\|v_1\| \|v_2\|} = \frac{23}{\sqrt{14} * \sqrt{41}} = 0,960001$$

Una vez determinados las actividades similares a cada una de las que están descritas en el curso en evaluación; podemos decir que se delimita un vecindario para cada una de ellas y es posible proceder a identificar aquellos metadatos adicionales que nos aportará cada una de las nuevas actividades descritas en el sub-dominio. Como se puede apreciar en la tabla 4.3, al utilizar los cursos equivalentes se multiplican los datos que tenemos para evaluar, por lo que es necesario aplicar algoritmos que mejoren los tiempos de respuesta y extraigan aquellos metadatos que no sean representativos para las búsquedas en cada actividad.

4.2.4.3. Aporte del historial de selección de OA

Con el historial de navegación sobre los OA recomendados para un curso es posible recuperar información implícita y explícita asociada al comportamiento de los usuarios finales del sistema.

La selección de múltiples OA para previsualizar sus metadatos antes de elegir alguno para una actividad, o no seleccionar del todo los OA presentados sugiere que existen desajustes sobre la función de recomendación, o bien, que los descriptores de la actividad estaban incompletos antes de generar la recomendación.

Cómo primer mecanismo de ajuste, se asumirá que si un usuario navega por cinco OA dentro de los top-N recomendados y no selecciona ninguno; se debe a un error en la fórmula aplicada sobre los OA para determinar su afinidad. En este caso, se buscarán los metadatos que compartan los OA visualizados que no coincidan con los descriptores base de la actividad (aquellos que sean comunes solo a los OA) y se bajarán de prioridad en la lista de recomendación todos los OA que coincidan con ellos.

Por otro lado, si un usuario selecciona un OA recomendado de alguna de las actividades que se encuentran en el grupo C3 descrito en la primer sección de 4.2.4 y clasificados en 4.2.4.2, se les subirá de prioridad a aquellos OA que compartan los términos únicos asociados a ese OA que respondió a las necesidades específicas del usuario después de la recomendación.

Para determinar el valor asociado a la prioridad que se incrementará o decrementará, se usará como aproximación la función dada por:

$$\frac{\text{cantidad de veces seleccionado}}{\text{cantidad de veces visto}} * \frac{\text{Total de metadatos únicos}}{\text{Total de metadatos del OA}}$$

Este ajuste se aplicará al valor asignado en la recomendación de cada OA según sea el caso. Para el caso de las actividades que sean parte del conjunto C4 y C5 no influirán en el ajuste.

4.2.4.4. Recomendación por estilos de aprendizaje

Los estilos de aprendizaje están asociados a las preferencias de un estudiante durante el proceso de enseñanza-aprendizaje, por lo que su definición implica que está asociado a un modelo de usuario más que a un modelo de curso.

Desde el punto de vista del recomendador, no se podría generar una sugerencia directa para los estudiantes sin contar con un perfil de usuario que los describa y actualmente no se cuenta con suficiente información para construirlo. Sin embargo se podría generar una clasificación de los recursos utilizando como referencia la categoría de documento a la que pertenece.

Las figuras 4.10 y 4.11 muestran dos clasificaciones de los recursos disponibles en ARIADNE. La figura 4.10 fue tomada de una de las presentaciones disponibles en sitio web oficial del repositorio y representa las categorías de recursos de aprendizaje que se encuentran en él. Por otro lado, la segunda imagen es un fragmento de los tipos de recurso que se pueden encontrar en el mismo repositorio usando el ARIADNE Finder.

Para definir la pertenencia de los OA a cada estilo, primero se categorizaron los diversos tipos de recursos disponibles (juegos, fotos, videos, textos en formato .doc, .html, .pdf, .ppt, .txt y otros formatos multimedia) en alguna de las categorías propuestas para organizar los recursos y seguidamente, se utilizó el criterio de un experto para definir la posible utilidad de las categorías con respecto a cada estilo de aprendizaje.

Como referencia se utilizaron las dicotomías de Felder y Silverman, De este modo, la

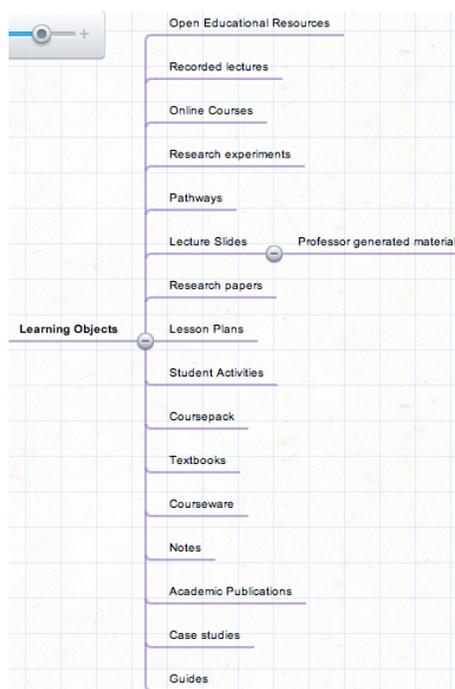


Figura 4.10: Tipos de OA disponibles ARIADNE. Tomado de <http://www.ariadne-eu.org/content/more-info>

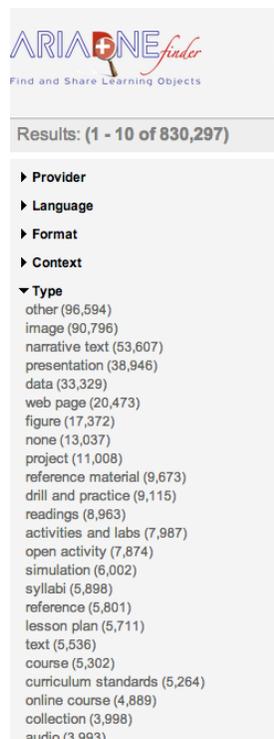


Figura 4.11: Fragmento de tipos de OA disponibles en el repositorio según AriadneFinder. Disponible en <http://ariadne.cs.kuleuven.be/finder/ariadne/>

matriz de utilidad esperada contiene las dicotomías definidas para los distintos estilos de aprendizaje como se puede ver en la figura 4.12.

	nbr_tipo_doc character varying(50)	auditivo character	visual character	secuencial character	global character	sensitivo character	intuitivo character	activo character	reflexivo character	inductivo character	deductivo character
1	Aplicaciones	Alto	Alto	Alto	Bajo	Alto	Bajo	Alto	Bajo	Alto	Bajo
2	Archivos Flash	Alto	Alto	Alto	Bajo	Alto	Bajo	Alto	Bajo	Alto	Bajo
3	Artículos de Investigaciones	Medio	Medio	Bajo	Alto	Bajo	Alto	Bajo	Alto	Bajo	Alto
4	Audio Libros	Alto	Bajo	Bajo	Alto	Medio	Bajo	Bajo	Alto	Medio	Medio
5	Casos de estudio	Medio	Medio	Alto	Bajo	Bajo	Alto	Medio	Medio	Alto	Bajo
6	Cursos online	Bajo	Alto	Medio	Medio	Alto	Bajo	Medio	Medio	Medio	Medio
7	Datos de muestra	Medio	Medio	Alto	Bajo	Medio	Medio	Bajo	Alto	Medio	Medio
8	Diagramas	Bajo	Alto	Alto	Bajo	Medio	Medio	Medio	Medio	Bajo	Alto
9	Fotografías	Bajo	Alto	Bajo	Alto	Alto	Bajo	Alto	Bajo	Bajo	Alto
10	Guías de estudio	Medio	Medio	Medio	Medio	Bajo	Alto	Medio	Medio	Medio	Medio
11	Herramientas de redes sociales	Medio	Medio	Medio	Medio	Alto	Bajo	Medio	Medio	Medio	Medio
12	Hojas de Cálculo	Medio	Medio	Alto	Medio	Bajo	Alto	Medio	Medio	Medio	Medio
13	Instrucciones para experimentos	Medio	Medio	Medio	Medio	Alto	Bajo	Medio	Medio	Medio	Medio
14	Juegos	Alto	Alto	Alto	Bajo	Alto	Bajo	Alto	Bajo	Alto	Bajo
15	Lecturas (PDF, doc, txt, notas)	Medio	Medio	Alto	Bajo	Bajo	Alto	Bajo	Medio	Alto	Medio
16	Libros de texto	Medio	Medio	Alto	Bajo	Bajo	Alto	Bajo	Medio	Alto	Medio
17	Modelos y simulaciones	Medio	Medio	Alto	Bajo	Alto	Bajo	Alto	Bajo	Medio	Medio
18	Presentaciones (Powerpoint, prezzy)	Alto	Alto	Alto	Bajo	Alto	Bajo	Alto	Bajo	Alto	Medio
19	Publicaciones académicas largas	Medio	Medio	Alto	Bajo	Bajo	Alto	Bajo	Alto	Medio	Alto
20	Quices y tests	Medio	Medio	Alto	Bajo	Bajo	Alto	Bajo	Alto	Medio	Alto
21	Tutoriales	Medio	Medio	Alto	Bajo	Medio	Medio	Alto	Bajo	Medio	Alto
22	Videos y animaciones	Alto	Alto	Alto	Bajo	Alto	Bajo	Alto	Bajo	Medio	Alto

Figura 4.12: Valores de aprovechamiento asociados a las categorías generales de objetos a partir del criterio experto.

Para alinear los metadatos de los OA con la utilidad esperada para un estilo de aprendizaje, basta con extraer el tipo de documento que contiene para determinar su categoría y de este modo conocer la utilidad que se espera para un estilo de aprendizaje en particular.

En el momento en que se cuente con un modelo de usuario para los estudiantes, este mecanismo de clasificación sería reemplazado por un algoritmo de recomendación que determine la utilidad de los OA del banco de recursos sugeridos y los organice de acuerdo al estilo de aprendizaje específico del usuario que lo desee utilizar.

4.2.5. Arquitectura detallada

Después de tener claras las relaciones de conceptos y los elementos que componen este agente, se puede presentar la integración de sus componentes en la figura 4.13.

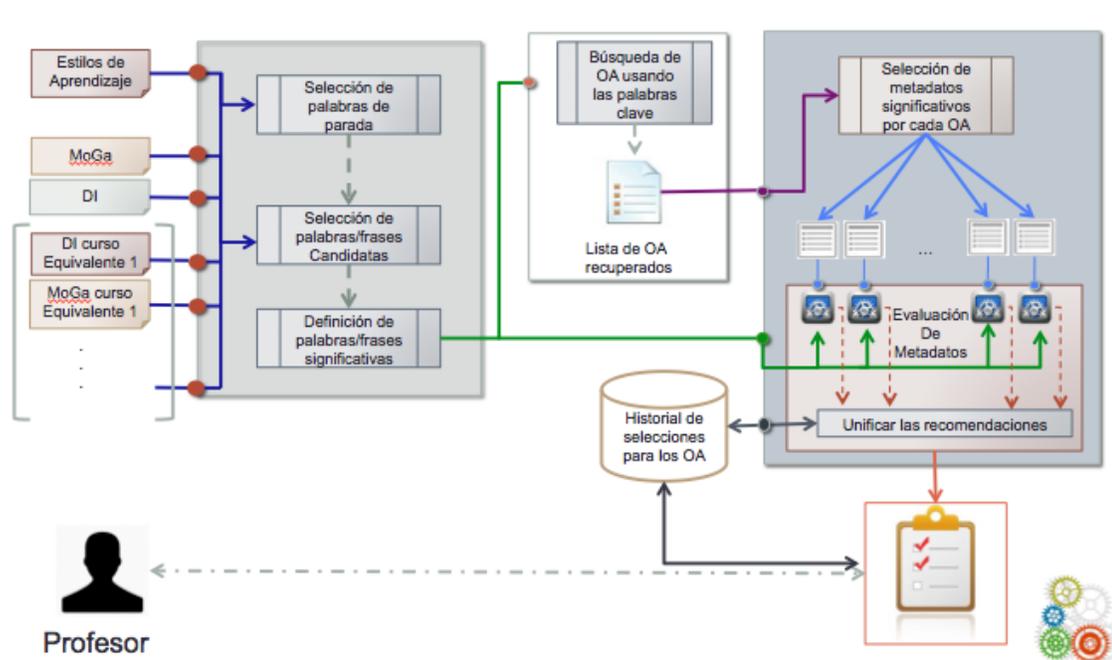


Figura 4.13: Arquitectura detallada de la solución

Esta figura refleja la secuencia en la que deberán interactuar cada una de las partes con el fin de generar una recomendación al usuario final.

Algunos de sus componentes son “cajas negras” que contienen algoritmos o conexiones con repositorios remotos y todas convergen en un mecanismo de recomendación simplificado que puede segmentar sus funcionalidades a través de casos de uso.

La similitud de cosenos se utilizará tras extraer los metadatos significativos de los OA recuperados mediante consultas. Una vez seleccionados los más afines, se refinará el proceso de búsqueda aplicando el TF-IDF con el fin de unificar la recomendación.

Según las descripciones brindadas en (Adomavicius y Tuzhilin, 2005; Burke, 2007), nuestro sistema recomendador se ubica en la categoría de híbrido por una serie de elementos que forman parte de su estructuración. La primera de ellos, es el uso de características propias de los sistemas colaborativos (aplicación de la similitud de coseno y balanceo de pesos con las evaluaciones de los usuarios), demográficos (delimitación de vecindarios de acuerdo a las características del curso) y de los basados en contenidos (aplicación del TF-IDF) combinadas en los procesos asociados al entrenamiento y generación de la re-

comendación. Además, la construcción de la hibridación al generar las recomendaciones se da mediante el uso de técnicas como el procesamiento en cascada y manejo de pesos (ajustes de peso mediante metadatos y aumento de características) y la utilización de representaciones mixtas (al incluir los estilos de aprendizaje en la clasificación del OA).

4.3. Problemas asociados al diseño

Existen otros posibles problemas que están relacionados al proceso de diseño y análisis de datos, los cuales se presentan a continuación:

Asociaciones léxicas

Las asociaciones léxicas son las palabras que están compuestas por dos o más términos; o mejor dicho, son una frase y podrían verse excluidas de las listas de palabras clave por el uso de diccionarios de datos al manipular los texto.

Es necesario encontrar un mecanismo que permita identificar estos casos para aprovecharlos en los procesos de búsqueda de recursos, por ejemplo, la medida de información mutua.

Cold Start

El problema de Cold Start se da cuando ingresan nuevos usuarios al sistema y existe poca información para generar una recomendación, o bien, se agrega un nuevo ítem a las colecciones y no se cuenta con parámetros suficientes para recomendarlo a los usuarios. La solución a este tipo de problemas suele encontrarse al combinar técnicas colaborativas y basadas en contenidos; de modo que el sistema ya no solo dependa de información explícita en la base de datos.

Algo similar ocurre con los algoritmos para “suavizado” que se utilizan cuando una recomendación retorna cero. En este caso, ninguno de los ítems disponibles coincide con las

preferencias de los usuarios; por lo que se deben modificar los criterios de recomendación para alcanzar una respuesta.

Evaluación del modelo

Como cualquier sistema, es necesario evaluar que la propuesta realizada es adecuada para lograr los objetivos propuestos. En nuestro caso la evaluación realizada tiene como objetivo comprobar la validez de las tareas de adaptación propuestas para recomendar OA pertinentes para un determinado curso.

Estas evaluaciones del rendimiento se basan en conteos de los registros de prueba correcta o incorrectamente predichos, por lo que se suelen clasificar en dos tipos principales: la evaluación explícita (que es brindada por el usuario, por ejemplo los sistemas de calificación mediante estrellas o puntos) o la implícita (deducida del comportamiento, ya que si una sugerencia es pertinente, es posible que sea tomada por el usuario).

En el contexto de los sistemas recomendadores, las evaluaciones solo pueden darse en términos numéricos ya que la aplicación de otro tipo de métricas se ve afectado por los factores subjetivos de la recomendación y los constantes cambios del entorno de pruebas.

Algunos de los algoritmos comunes para medir el rendimiento de un sistema son:

- MAE¹⁰, o Error Medio Absoluto. Consiste en determinar la magnitud media del error al realizar la clasificación.
- RMSE¹¹ es una medida utilizada para estimar la distancia entre un modelo y los valores observados. Junto con el MAE, ayudan a determinar la variación que se puede presentar en el margen de error.
- RAE¹², o Error Absoluto Relativo. Es el total absoluto del error y compara tanto los datos del modelo como los del conjunto de entrenamiento.

¹⁰MAE, Acrónimo de Mean Absolute error

¹¹RMSE, Acrónimo de Root Mean Squared Error

¹²RAE, Acrónimo de Relative Absolute Error

- Precisión: La probabilidad de que un elemento sea relevante.
- Pasadas sobre la base de datos
- Tasas de falsos positivos

Para la validación del modelo de la clasificación por estilos de aprendizaje se realizará una validación cruzada mediante clasificadores Bayesianos, de árbol y basados en reglas con el fin de aplicar sólo aquel con mejores resultados con respecto al porcentaje de aciertos, RMSE y MAE.

En cuanto a la validación de las recomendaciones del prototipo en general, será necesario esperar a un plan piloto para capturar información que permita realizar los análisis respectivos.

5 Implementación y resultados

En este capítulo se presentará el resumen del proceso de implementación del prototipo para verificar la validez de la propuesta de solución planteada y los resultados obtenidos durante el proceso.

5.1. Consideraciones técnicas

Existen distintos tipos de herramientas y repositorios de algoritmos que se pueden utilizar para facilitar la manipulación de textos e interpretar su contenido. Algunas están asociadas al análisis textual en concreto, mientras que otras están más relacionadas con el proceso de recomendación. Por lo que a continuación se presenta su aplicación a las distintas etapas de desarrollo del sistema.

5.1.1. Extensiones de PostgreSQL para manipulación de datos

El postgres-contrib provee características adicionales para la base de datos, algunas de las cuales se encuentran en pruebas antes de ser incluídas como parte del núcleo de PostgreSQL. La librería `pg_similarity`¹ ofrece una serie de implementaciones para calcular similitudes sobre datos y es configurable para trabajar sobre bases de datos específicas (la instalación no afecta a todas las bases de datos en el servidor); sin embargo, las pruebas realizadas para realizar la integración con postgresql 8.4 no brindaron los resultados esperados y la plataforma de dotlrn instalada producción funciona con esta versión de base de datos.

Por otro lado, la librería `smlr` sí es integrable con PostgreSQL 8.4 y permite realizar las estimaciones mediante un sistema que se denomina FTS²; el cuál, a partir de la aplica-

¹Los instaladores están disponibles en github.

²FTS, Acrónimo de Full Text Search

ción de trigramas sobre un texto; determina los valores de similitud entre dos cadenas completas de texto y la librería de “fuzzystring” ofrece la medida de Levenshtein, la cuál está asociada a la cantidad de transformaciones requeridas para convertir una cadena de caracteres en otra.

Sin embargo, este tipo de análisis no permite realizar la poda de las palabras clave, ya que el resultado se da sobre el total del documento; por lo que es necesario recurrir a otros mecanismos que nos permita la selección de las palabras claves.

La extensión de trigramas (pg_trgm) nos da acceso al manejo de márgenes de error en los textos y análisis por trigramas en las palabras o frases encontradas. Sin embargo, según el sitio web oficial de postgresql³, esta herramienta de soporte morfológico no incluye las características para análisis de palabras compuestas o frases. De modo que sólo ofrece soporte para diccionarios de MySpell y Hunspell; con las operaciones básicas de Hunspell para palabras compuestas.

Cómo alternativa para este inconveniente con el análisis textual, se buscó una solución mediante la utilización de WEKA⁴, que es una colección de algoritmos de aprendizaje desarrollado por la Universidad de Waikato, Australia y es ampliamente utilizada a nivel académico para análisis de datos.

5.2. Adquisición del Modelo del Curso

Para preprocesar los textos y generar los conjuntos de entrenamiento se utilizaron elementos propios de Postgresql y WEKA.

Se implementó un proyecto en Java donde se incluyó la versión 3.7.11 de WEKA para desarrolladores. En este proyecto se modificó el conector de WEKA para realizar la conexión con Postgres y se implementaron tareas adicionales para manipular los datos. Además, se incluyeron funciones específicas para el preprocesamiento, como la aplica-

³<http://www.postgresql.org/docs/9.1/static/textsearch-dictionaries.html>

⁴Acrónimo de Waikato Environment for Knowledge Analysis, y disponible en <http://www.cs.waikato.ac.nz/ml/weka/>

ción de reglas para reducir la cardinalidad de los conjuntos resultantes y la lectura de los verbos a excluir. De este modo, el intercambio de datos se manejó sin pasar por archivos en disco intermedios y una vez cargados, se procedió a preprocesarlos mediante filtros y clasificadores.

Para el caso de los datos provenientes de DI, se agregó un prefijo al identificador de cada actividad y se asignó como nombre de clase; así, al exportar los contenidos a formato .arff cada identificador representará a una clase y los texto contendrán la descripción de la actividad, como se muestra en el fragmento de archivo a continuación:

```
@relation 15051_di
@attribute num_sesion {actividad_7,actividad_8,actividad_1,...}
@attribute nom_actividad string

@data

actividad_7,'ER al Modelo Relacional. Presenta ...'
actividad_8,'Quiz ...'
...
```

El atributo denominado *num_sesion* contiene los identificadores de las actividades definidas, mientras que *nom_actividad* contiene el texto que describe a dicha actividad. Sin embargo, aún no se han aplicado los filtros para tratar los textos como vectores ni se ha normalizado la representación de su contenido; por lo que cada descripción representa a un elemento único e independiente de los demás.

Después de aplicar los filtros para convertir las descripciones de las actividades a cadenas de caracteres y convertir a minúsculas todos los textos, obtenemos una reducción importante de palabras, que se reflejarán posteriormente en la matriz que representa al modelo del curso.

El diseño de sistemas que trabajen con lenguaje natural en español se ha centrado en las características morfológicas, sintácticas y semánticas del idioma. Sin embargo, la aplicación de las técnicas tradicionales en recuperación de la información, como el stemming y la extracción de las palabras de parada, producen pérdidas importantes de información.

Por este motivo, algunas de las propuestas para mejorar la precisión de las evaluaciones de datos se han inclinado a aplicar técnicas mixtas o de mayor complejidad computacional (Atserias y cols., 2006; Bick, 2006).

El uso de diccionarios de propósito general y específico para dominios particulares es otra práctica común, sin embargo, requiere de una colección importante de documentos y de la validación por parte de expertos en el área. Además, existen reconocedores de nombres de entidades (NER, acrónimo de Name Entity Recognizer por sus siglas en inglés), etiquetadores de las partes de la oración (POS o POST, *Part-of-Speech Tagger*, por sus siglas en inglés) y extractores de frases (Chunkers) que son componentes de los sistemas de procesamiento de lenguaje natural y ayudan a mejorar la comprensión de las expresiones brindadas.

Se encontraron numerosas referencias de sistemas que utilizan estas herramientas como por ejemplo CLIPS⁵, FreeLing (Atserias y cols., 2006), OpenNLP⁶, Stanford Named Entity Recognizer⁷, Phora (Palomar y cols., 2001), Cavorite⁸, OpenCalais⁹, OpenNLP Pos Tagging Engine de Apache Stanbol, entre otras. Que utilizan desde bases de datos léxicas hasta modelos de análisis de texto probabilístico, perceptrones u ontologías para realizar las clasificaciones de las frases y ponerlas disponibles para los usuarios finales.

Wordnet¹⁰ es una base de datos léxica que originalmente fue definida para el idioma inglés, sin embargo, existen wordnets en diversos idiomas, entre ellos el español. La colección de la versión 3.0¹¹ se denomina MCR30 y está disponible bajo la licencia *GNU Lesser General Public License*, de modo que puede ser compartido y utilizado por cualquier usuario, pero no modificado. Esta colección ofrece 36681 palabras en español con

⁵CLIPS, acrónimo de Computational Linguistics & Psycholinguistics, de la Universidad de Antwerp. Es un centro de investigación cuyo sitio web está disponible en <http://www.clips.ua.ac.be/pages/using-wikicorpus-nltk-to-build-a-spanish-part-of-speech-tagger>

⁶OpenNLP, disponible en <http://opennlp.apache.org/>

⁷Stanford NER, disponible en <http://nlp.stanford.edu/software/CRF-NER.shtml>

⁸Cavorite, <http://cavorite.com/labs/nlp/opennlp-models-es/>

⁹OpenCalais, <http://www.opencalais.com>

¹⁰Disponible en wordnet.princeton.edu

¹¹Disponible para descarga en compling.hss.ntu.edu.sg/omw/

sus respectivos clasificadores, relaciones y sinónimos; los cuales, en nuestro caso, serán utilizado para reducir el conjunto de palabras que constituyen el perfil de un curso, junto con los Ngramas y tokens extraídos de los textos.

El filtro de aprendizaje no supervisado, denominado `StringToWordVector`, ofrece la posibilidad de realizar cálculos de frecuencia de aparición, TF-IDF y parametrizar el mínimo de apariciones de una palabra para ser considerada dentro de la matriz de confusión¹². Así, una vez aplicado el primer filtro, podemos convertir los datos cargados a una estructura más representativa a nivel computacional:

```
@relation '15051_di-weka.filters.unsupervised.attribute.String...'
@attribute num_sesion {a7,a8,a2,a3,a4,a5,a1,a6}
@attribute '\\011identificaci\\303\\263n' numeric
@attribute actores/objetos/entidades numeric
@attribute atributos numeric
.....
@attribute usuario numeric
@attribute usuarios numeric

@data

{2 1,13 1,18 2,23 1,26 1}
{1 3,3 1,4 1,5 1,6 1,7 1,8 1,14 1,16 1,18 1,21 3,23 1,25 1,27 1}
{0 a8,39 1}
{0 a2,12 1,46 1,47 1,53 1,54 1,55 1,56 1,58 1,62 1}
...
```

Esta estructura es la que se presentó en la sección de diseño en la tabla 4.1 y representa la cantidad de apariciones de cada término en las distintas actividades del agente. Estos mismos valores se pueden normalizar con el fin de delimitarlos a un rango específico de valores (generalmente 0 y 1) como se muestra en la figura 5.1.

En el caso de PostgreSQL, el FTS permite realizar búsquedas sobre una estructura denominada *tsvector*, el cuál, reduce un texto completo a un vector de palabras. Las búsquedas

¹²<http://weka.sourceforge.net/doc.dev/weka/filters/unsupervised/attribute/StringToWordVector.html>

No.	1: num_sesion Nominal	2: 2000 Numeric	3: \\\\011atributos Numeric	4: \\\\011entidad Numeric	5: \\\\011identificaci... Numeric	6: \\\\011llaves Numeric	7: \\\\011presentaci... Numeric	8: \\\\011presentar Numeric	9: \\\\011proceso Numeric	10: \\\\011relac Numeric
7	a4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	a4	0.566...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	a5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	a1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	a1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	a1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
13	a1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
14	a2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
15	a2	0.0	0.0	0.0	0.0	0.0	0.8589415326898612	0.0	0.0	0.0
16	a5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
17	a5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
18	a5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
19	a6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.905403873...	0.0
20	a6	0.0	0.658778083...	0.65877808...	0.0	0.6587780...	0.6587780830400227	0.0	0.0	0.65877808...
21	a6	0.480...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
22	a7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
23	a7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
24	a8	0.0	0.0	0.0	0.0	0.0	0.0	0.6402172185...	0.0	0.0
25	a8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.818968617...	0.0
26	a8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
27	a7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figura 5.1: Muestra de matriz normalizada

das sobre este tipo de estructuras están reducidas a consultas simples de pertenencia de términos y porcentajes de similitud. Por ejemplo, para consultar por la frase “mapa de conceptos” dentro de un texto; se debe introducir por ejemplo {mapa & concepto} como condición, o bien un texto convertido al formato de `ts_query`. Por otro lado, la medida de similitud ofrecida por la librería de trigramas está directamente relacionada con la cantidad de trigramas extraídos de las cadenas de texto a analizar, de modo que se puede caer en errores de interpretación al encontrar que cadenas que contienen al mismo subconjunto de palabras buscado, poseen medidas de similitud distintas debido su longitud. Existen otros mecanismos de consulta que se puede aplicar sobre PG, ya que existen otros comandos que se pueden utilizar para realizar búsquedas sobre textos sin la conversión explícita a un modelo vectorial.

Como caso de estudio se utilizó el diseño instruccional de un curso equivalente al TI-3600: Bases de Datos para Administración de Tecnologías de Información y se aplicaron una serie de procesos que buscan mejorar la calidad de los criterios de búsqueda y selección automáticos.

En la figura 5.2 se muestran los resultados obtenidos al aplicar distintas técnicas de extracción de tokens sobre documentos. Cómo se puede apreciar, las distintas técnicas generan resultados con variantes significativas en la cardinalidad de sus conjuntos de

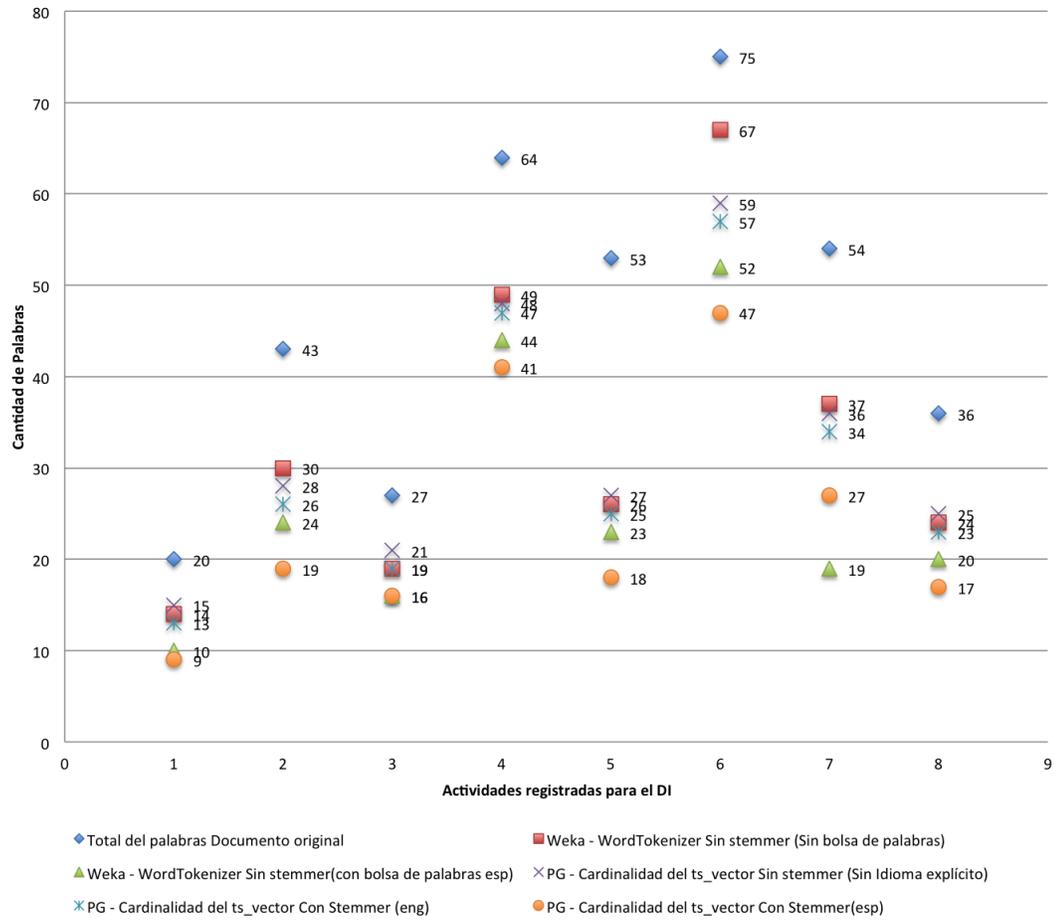


Figura 5.2: Cardinalidad de resultados de los métodos para extracción de tokens aplicados en el caso de estudio.

respuesta. La aplicación de diccionarios o colecciones de palabras a extraer, reduce significativamente la cantidad de resultados obtenidos; del mismo modo que lo hace la correcta identificación del idioma bajo en que se está trabajando.

En el caso particular de aquellas técnicas aplicadas sobre textos con stemmers, los conjuntos fueron mucho menores a aquellos otros en los que no se utilizó esta misma técnica; siendo la aplicación de stemmers en español sobre un tsvector el conjunto con menor cardinalidad de respuesta, pero también con menor utilidad para un lector humano o un buscador. Mientras que el conjunto con mayor cardinalidad de elementos se obtuvo al aplicar el wordtokenizer de weka sin stemmer y sin una bolsa de palabras, en cuyo caso, se conservó el sentido de las palabras, pero se generó un mayor volumen de información que no es precisamente relevante.

Como parte de los resultados obtenidos, aquellos conjuntos con mayor cardinalidad habían sido afectados por la detección de palabras con errores ortográficos como palabras nuevas e independientes; del mismo modo que la aparición de las mayúsculas en distintas partes de la oración ejerció cierta influencia en la forma de escribir las siglas, por lo que fue necesario un preprocesamiento para mejorar los resultados obtenidos.

La figura 5.3 muestra el diagrama simplificado de preprocesamiento de los textos para

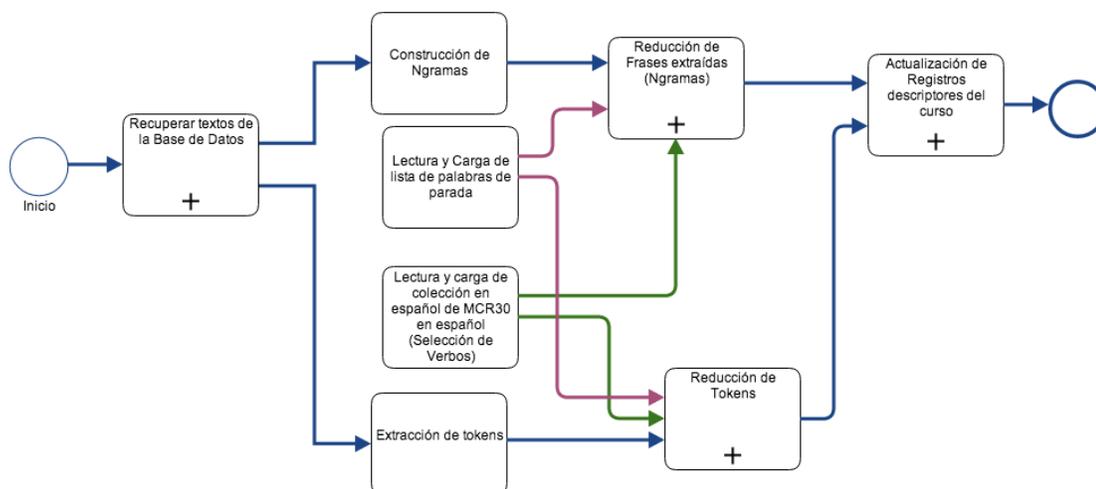


Figura 5.3: Diagrama simplificado de Preprocesamiento de datos

extraer los elementos a buscar en el repositorio. En este diagrama se reflejan las tres entradas básicas para realizar la reducción: los registros de la base de datos con las descripciones de las actividades, el conjunto de palabras de parada preseleccionadas para reducir los tokens a extraer y un subconjunto de la colección MCR30 con verbos y sus conjugaciones.

En dicho diagrama se muestran tres subprocesos principales, la extracción de información de la base de datos, la reducción de Ngramas y la reducción de Tokens. Dichos procesos encapsulan de una serie de tareas que buscan aproximar las funciones de los chunkers y los NER, ya que este tipo de tareas excede los alcances de la tesis, pero forman una parte crítica de la construcción de los modelos de búsqueda y se describirán a continuación.

El proceso de reducción de frases representado en la figura 5.4 inicia con la extracción

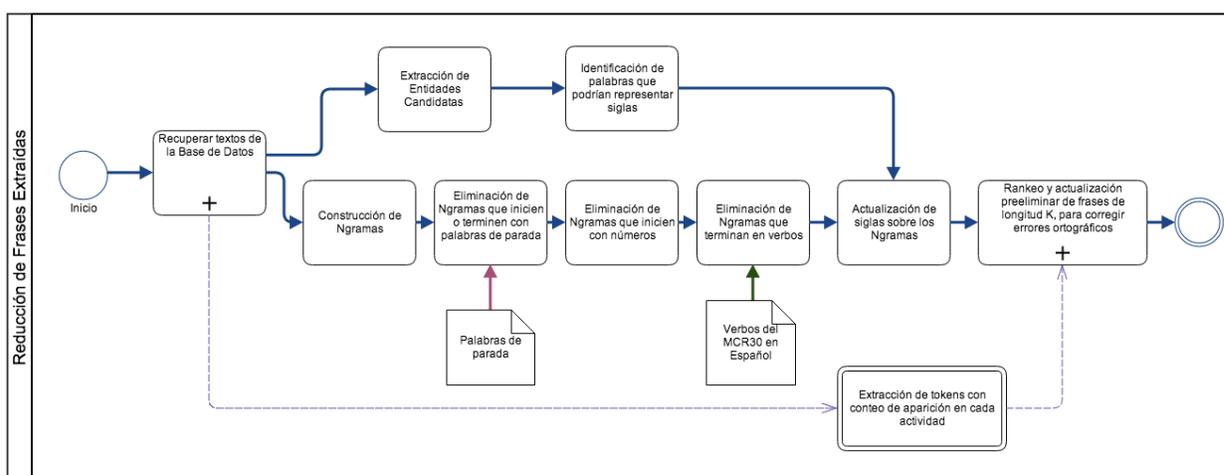


Figura 5.4: Diagrama del proceso de reducción de frases

de información de la base de datos y se procesa mediante Weka. Como se mencionó anteriormente, en Weka, la extracción de Ngramas consiste en la construcción de frases, no en combinaciones de letras como ocurre en el caso de PostgreSQL.

Para la construcción de los Ngramas, se definió un rango de dos a N palabras, donde el valor por omisión cuando no se especifica en valor de N máximo se asigna automática-

mente a 8. El resultado de esta construcción de palabras retorna todas las combinaciones de frases consecutivas posibles a partir del texto original que describe cada actividad. De modo que si procesamos la frase: “sistemas administradores de bases de datos”, el resultado sin reducir consistirá de 21 Ngramas posibles a partir de la misma expresión si no se descartan las palabras de parada, y 19 si se excluyen las palabras de parada.

Por esta razón, se utilizó un proceso empírico a fin de reducir la cantidad de expresiones que se puedan generar a partir de un mismo texto. La razón principal para no utilizar una herramienta de procesamiento de lenguaje natural en esta fase se debe a que existen pocas herramientas específicas para trabajar con el idioma español, la herramienta que presentó mejores resultados fue *freeling 3.1*¹³, sin embargo al extraer los archivos del API se produjeron errores que no permitieron su integración con el código, pero no se descarta su posterior integración con el sistema.

Una vez construídos los posibles Ngramas, se procedió a reducir la cardinalidad del conjunto de resultados mediante una serie de reglas simples:

- Eliminar los Ngramas que inician o terminen con palabras de parada: Las palabras de parada incluyen expresiones como “y”, “de”, “la”, “los”, “en”, entre otras. Estas palabras por lo general representan conectores gramaticales y carecen de sentido para una búsqueda. Cualquier frase que inicie o termine con una de las palabras de esta lista, contará con otra dentro del conjunto de respuestas que contenga la misma frase, pero sin las palabras de parada al inicio o al final, por lo que no representa una pérdida su eliminación.
- Eliminar los Ngramas que inician con números: Al construir las frases, los números se incluyen dentro de las combinaciones de palabras. En esta reducción se consideraron solo los números del inicio de las frases para no descartar las características de versión o año de un elemento, por ejemplo “SQL 2000 en el servidor” tiene sentido, pero “2000 en el servidor” carece del mismo o bien, nos podría desviar del contexto de la búsqueda.

¹³<http://nlp.lsi.upc.edu/freeling/>

- Eliminar los Ngramas que representan verbos: Tanto en español como en otros idiomas existen verbos compuestos o formados por más de una palabra. Se usó la colección MCR30 como referencia para intentar reducir la frecuencia de captura de estas expresiones. Por ejemplo: “salida del sol” = “amanecer”.

Una vez reducido el conjunto de Ngramas que representan a cada actividad, se registran en la base de datos y se procede a una segunda fase de reducción del conjunto de respuestas. Para este punto aún se cuenta con inconsistencias en la sintaxis y con diversas representaciones de la misma palabra, ya sea por faltas ortográficas, o por simples cambios en la forma de representar las entidades. Para lo cual que se aprovecharán las características de los trigramas de PostgreSQL, donde cada trigrama es representado por tres letras, en lugar de tres palabras.

La selección de las entidades se realizó de un modo sencillo, primero se eligieron todas aquellas palabras que inician por mayúscula (extraídas directamente de los documentos) y no están definidas después de un punto y seguido como posibles entidades candidatas. Posteriormente se seleccionaron aquellas que no representan a verbos o palabras conjugadas (se extrae la raíz mediante los diccionarios de postgresql) y se seleccionan aquellas que no sufrieron cambios.

Como tercer paso, se eligieron aquellas que incluyen más de una mayúscula como posibles siglas y para aquellas que aparecen en formas distintas, por ejemplo “SQL” y “Sql”, se eligió como representación la que aparezca más veces dentro de los documentos y se procedió a realizar una actualización sobre los Ngramas y Tokens generados. Cabe rescatar que esta no es una solución definitiva para la extracción de entidades, ya que se detectaron inconsistencias con verbos o apellidos que quedaron catalogados como entidades candidatas, pero no se encontró un procedimiento efectivo para separarlas y esta es un área que compete al procesamiento de lenguaje natural, sin embargo, su aporte para reducir los Ngramas y Tokens generados no es despreciable.

La detección de cambios sobre frases similares no es un proceso trivial a partir de tri-

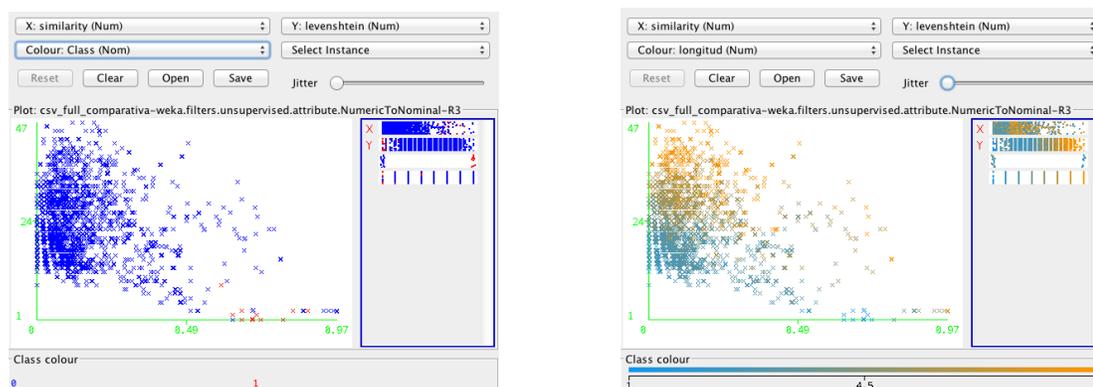


Figura 5.5: Clasificación de los pares de fra-
ses aplicando la distancia de Levenshtein y res de frases con respecto a la distancia de Similarity.

Figura 5.6: Relación de longitud de los pa-
res de frases con respecto a la distancia de Levenshtein y Similarity.

gramas. El uso de la distancia de Levenshtein¹⁴ como parámetro de referencia no es suficiente para determinar si una palabra fue convertida en otra, por ejemplo: entre “base” y “bases” se está trabajando sobre el mismo concepto y la distancia es igual a 1, sin embargo, entre “ldf” y “mdf” la distancia también es igual a 1, pero se trata de un concepto distinto.

Si por otro lado se utiliza la medida de similitud provista por el paquete de trigramas, nos encontramos con un comportamiento similar. La similitud entre “base” y “bases” ronda el 60 %, mientras que “ldf” y “mdf ” rondan el 30 %. Pero para casos donde la longitud de las palabras es mayor, el margen de error se amplía significativamente. Es por esta razón que se planteo la idea de aprovechar las ventajas de ambas técnicas de comparación para modificar únicamente aquellas frases que cumplan con ciertos criterios de similitud. Como primer criterio, se seleccionaron solo las frases de la misma longitud de palabra para ser evaluadas contra sus similares; en este caso, la longitud está dada por la longitud del arreglo que se puede formar al separar el texto mediante espacios.

Una vez seleccionadas las palabras de la misma longitud, se procede a calcular su dis-

¹⁴Distancia de Levenshtein: Distancia que indica la cantidad de transformaciones que deben aplicarse a una palabra para convertirla en otra y que es aplicable sobre cadenas de texto de distintas longitudes.

tancia de Levenshtein y su porcentaje de similitud. Para ser consideradas de similares, se necesita un porcentaje de similitud superior al 60 % y una distancia de Levenshtein menor o igual a la longitud de la frase + 1.

Esta aproximación en los valores se dió tras analizar los Ngramas y Tokens recuperados para las actividades del caso de prueba. En el gráfico 5.5 se muestra un clúster con la relación de los textos encontrados en el caso de prueba. Para esto, se comparó cada uno de las frases encontradas contra aquellas que tuvieran su misma longitud de palabra para identificar las posibles variaciones de la misma frase. Para generar los valores de los ejes del conjunto de entrenamiento se calculó su similitud y la distancia de Levenshtein existente entre ellas, y posteriormente, se clasificaron de forma semiautomatizada para determinar si se trataba del mismo texto pero con variantes mínimas (errores de ortografía, diferencias entre siglas, tildes, entre otras) o de textos completamente distintos. Las instancias representadas en azul representan aquellos pares que no estaban relacionados de ninguna manera; mientras que los representados en rojo, son aquellos que tenían algún grado de relación entre sí.

De los 3822 pares de frases construídos, 21 pares representan frases con algún tipo de relación entre ellas. Se logró detectar un leve incremento en la distancia de Levenshtein, pero la muestra no fue lo suficientemente grande como para determinar una función de crecimiento que satisfaga todos los casos. En cuanto a las medidas de similitud, se mantuvieron relativamente constantes, con un valor mínimo cercano al 0.6 en aquellos pares relacionados como se muestra en la figura 5.7 .

L. de frase	Similarity			Levenshtein	
	Menor	Mayor	Media aritmetica	Menor	Mayor
1	0.636364	0.8	0.707431667	1	2
2	0.666667	0.666667	0.666667	3	3
3	0.8	0.833333	0.811111	1	3
4	0.6	0.833333	0.8566665	3	3
5	0.88	0.88	0.88	3	3
6	?	?	?	?	?
7	?	?	?	?	?
8	?	?	?	?	?

Figura 5.7: Resumen de valores extraídos para los pares de frases que tienen relación.

Para frases de longitud mayor o igual a cinco no se encontraron casos de inconsistencias en cuanto a la escritura de los textos. Esto se debe a que las frases largas similares generalmente fueron extraídas de una misma actividad, por lo que su similitud va a ser cercana a uno (casi idénticos) o si coinciden en longitud y son distintos, se va a tratar de frases que provienen de textos completamente distintos, como se puede apreciar en la misma representación del clúster de resultados pero con una métrica de color asociada a la longitud de textos en la figura 5.6. Por esta razón, se decidió asumir que para aquellas frases más largas se esperaría una mayor cantidad de cambios en las palabras pero con un comportamiento en los valores de similitud proporcional al comportamiento identificado en las frases de menor longitud. Este aspecto se puede solventar posteriormente con la integración de una herramienta específica para procesamiento de lenguaje natural que ofrezca detección de grupos nominales, donde la evaluación por distancias de frases pasaría a ser problema de una aplicación especializada en clasificación semántica de expresiones textuales.

Para seleccionar la frase que debe reemplazar a otra, se utilizó el criterio de mayoría simple de apariciones de las frases cuando el caso así lo permite. De otro modo, cuando existe un empate en las frecuencias de aparición de cada una de ellas, se procede a descomponerlas por cada una de sus palabras, por ejemplo en el caso de que “bases de dato” y “base de datos” posean la misma frecuencia de aparición dentro de los documentos, se procederá a comparar sus elementos de uno a uno, seleccionando aquél que aparezca con más frecuencia; por ejemplo, “bases” y “base”, “dato” y “datos” para construir una frase a partir de ambas expresiones. Finalmente, en caso de llegar a palabras sueltas y no encontrar un valor para elegir por mayoría simple, se procederá a elegir aquella palabra que sea de mayor longitud, y si aún así se encuentra una igualdad de condiciones, se elegirá la palabra mayor mediante comparación de caracteres.

La figura 5.8 representa el proceso de extracción y reducción de tokens y que se utiliza parcialmente en la extracción de los Ngramas. El proceso se desarrolla en dos flujos con objetivos distintos, el primero de ellos es la extracción de tokens con las métricas de

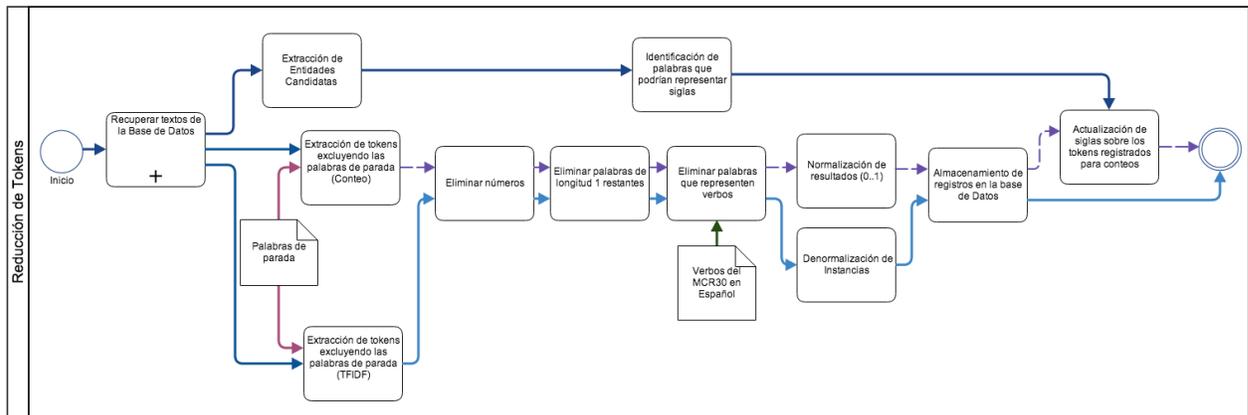


Figura 5.8: Diagrama del proceso de reducción de tokens

conteo simples, que serán utilizadas en las evaluaciones de los Ngramas y el segundo es la extracción de tokens con su TF-IDF asociado para determinar la relevancia de cada palabra en las actividades con respecto al total de documentos a tratar.

El proceso de reducción de los tokens extraídos es similar al aplicado a los Ngramas, sin embargo, se utilizaron algunas condiciones distintas para restringir aún más el conjunto de resultados. En la extracción de tokens, se asume automáticamente que los elementos dentro lista de palabras de parada que se le dió como parámetro a Weka serán eliminadas del conjunto de resultados, por lo que los filtros adicionales que se aplicaron son los siguientes:

- Eliminar los tokens que representan números: Por la misma razón mencionada en los Ngramas, un número por sí solo no ofrece información para ser significativo al describir algo.
- Eliminar los tokens que poseen longitud uno: Se eliminan las letras sueltas que no se contemplaron dentro de la lista de palabras de parada.
- Eliminar los tokens que representan verbos.

Después de aplicar estas reducciones los flujos se bifurcan. De este modo, se procede a realizar una denormalización¹⁵ de las instancias usando la suma como operación para el conteo de apariciones y se agrupan todas las dimensiones generadas en un solo registro. Mientras que para las instancias que llevan el TF-IDF se aplica una normalización de resultados para ajustar el rango de valores entre 0 y 1.

Dado que los valores de frecuencia generados para el TF-IDF se ven afectados por las inconsistencias en los textos y la cantidad de actividades (documentos); se realizó un preprocesamiento de los mismos antes de aplicar el algoritmo. Para este paso se trabajó sobre copias de los contenidos originales y se corrigieron utilizando los valores recuperados a partir de los Ngramas generados y el conteo de apariciones de cada palabra; así, una vez reducidas las ambigüedades textuales, se procedió a iniciar el cálculo del mismo. Una vez finalizado el proceso de preprocesamiento, la cardinalidad de los conjuntos de resultados obtenidos se redujo significativamente y los resultados de la aplicación de los distintos procesos se reflejan en la siguiente tabla.

Origen de datos	Sigla	Total	Num. Elementos únicos
Ngramas (weka)	N	218	168
Tokens con TF-IDF ** (weka)	T	152	99
Tokens con conteo de aparición ** (weka)	C	163	91
Siglas (Postgres)	S	6	4
Entidades Candidatas (Postgres)	P	18	12

Los elementos etiquetados con “***” son listas extraídas utilizando palabras de parada y reducción de frases como fueron descritas en los distintos procesos de preprocesamiento; de modo que se redujo la cardinalidad de los conjuntos de miles de combinaciones a unos pocos cientos.

Finalmente, estos resultados se transformaron en formato XML para representar los des-

¹⁵Denormalización: En weka, consiste en un filtro que agrupa los conjuntos que pertenecen a distintas instancias de la misma clase en una sola. Es el equivalente a agrupar todos los registros que cumplen una condición dada.

criptores de cada actividad, donde la categoría del término indica la forma en que fue extraído del texto y más adelante se utilizará para dar pesos durante el proceso de recomendación. La figura 5.9 muestra un fragmento de este perfil y cada uno de estos términos se utilizará para disparar las búsquedas contra el ROA remoto. Los términos identificados con la clase 0 son siglas (por ejemplo: SQL), los de clase 1 son frases que no contienen conectores gramaticales como “y”, “o”, “u”, “e”; seguidamente, los de clase 2 son palabras sueltas que no se encuentran como parte de ninguna frase extraída en los Ngramas y los de clase 3 son aquellas frases que sí poseen conectores gramaticales.

```
<curso id_di="15051">
  <actividad id="actividad_1">
    <termino clase="2">bases</termino>
    <termino clase="1">conceptos de datos</termino>
    <termino clase="3">conceptos de datos e información</termino>
    <termino clase="3">datos e información</termino>
    <termino clase="3">entrega y presentación</termino>
    <termino clase="3">entrega y presentación del programa</termino>
    <termino clase="2">introducción</termino>
    <termino clase="1">presentación de conceptos</termino>
    <termino clase="1">presentación de conceptos de proceso</termino>
    <termino clase="1">presentación del programa</termino>
    <termino clase="1">presentación del programa del curso</termino>
    <termino clase="1">programa del curso</termino>
    <termino clase="2">programas</termino>
  </actividad>
  <actividad id="actividad_2">
    ...
  </actividad>
  <actividad id="actividad_3">
    ...
  </actividad>
</curso>
```

Figura 5.9: Ejemplo de XML con el perfil de los textos que describen a un curso

5.3. Conexión con ROA

Para consumir recursos de un ROA es necesario construir o utilizar un cliente que ofrezca soporte para la cosecha de metadatos respetando las normativas establecidas para los agentes robóticos HTTP. Esta normativa busca que los agentes no se conviertan

atacantes contra los repositorios ¹⁶ en los que se desean realizar las cosechas de datos.

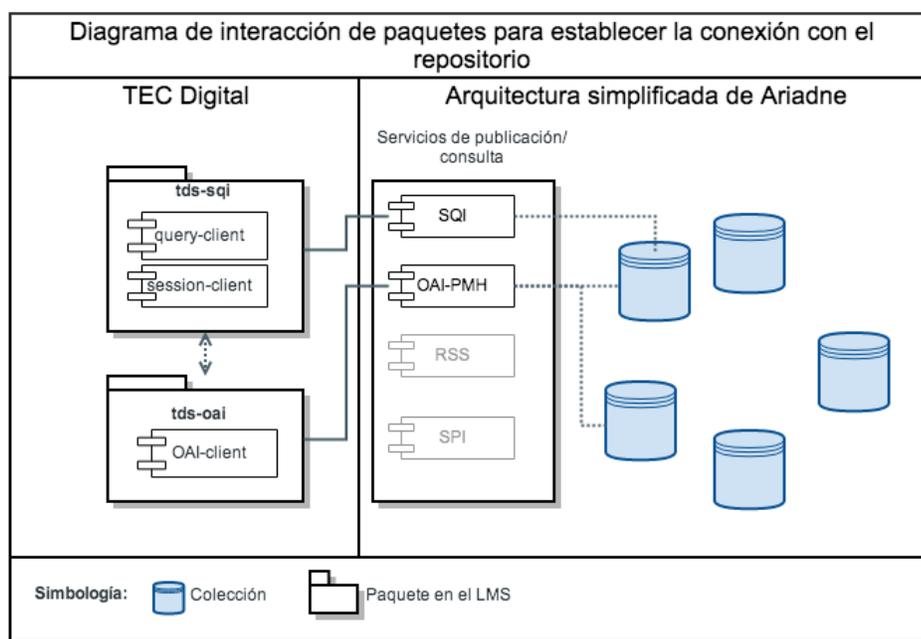


Figura 5.10: Diagrama simplificado de conexión con el ROA

Ariadne dispone de una serie de servicios y tecnologías para exponer sus recursos. Entre las tecnologías que se utilizan, destacan el uso de OAI-PMH para realizar cosecha entre repositorios, SQL para realizar consultas, SPI¹⁷ para realizar publicaciones y RSS¹⁸ para notificar a los interesados sobre los nuevos OA agregados a las colecciones disponibles (Odriozola, Luis, Ochoa, Parra Chico, y Duval, 2011). La selección de uno u otro mecanismo para consumir los recursos depende de las necesidades del usuario final, en nuestro caso particular, se intentó realizar la cosecha mediante dos mecanismos: el OAI-PMH y el SQL.

¹⁶Guías para desarrollo de agentes sobre OAI disponibles en <http://www.openarchives.org/OAI/2.0/guidelines-harvester.htm>

¹⁷SPI, Acrónimo de Simple Publishing Interface y responde al identificador CWA 16097:2010 del CEN.

¹⁸RSS, Acrónimo de Really Simple Syndication.

Los OA retornados por el servicio de OAI-PMH pueden venir dados tanto en LOM-IEEE¹⁹ como en Dublin Core. La especificación de LOM-IEEE, si se aplica completa, ofrece más información sobre el tipo de OA y su contenido de la que nos ofrece Dublin Core, además, los servicios de SQL para los que se encontraron referencias en la wiki de ARIADNE²⁰ utilizan este mismo formato para retornar los metadatos; por estas dos razones, se seleccionó el LOM-IEEE para recuperar los recursos.

En términos generales, LOM agrupa sus descriptores en nueve categorías principales, y a su vez, cada una de estas categorías se descompone en una serie de elementos que pueden variar en cuanto al número de apariciones según está definido en el estándar. Estas categorías aportan distintos tipos de información y según la descripción brindada en (*Draft Standard for Learning Object Metadata, 2002*), las funciones de cada una de ellas son las siguientes:

- General: contiene información que describe al OA como un todo.
- LifeCycle: posee datos sobre la historia del OA y como esto ha afectado su evolución.
- Meta-metadata: agrupa información sobre las instancias de los metadatos en sí.
- Technical: contiene los requerimientos técnicos y características técnicas del OA.
- Educational: presenta las características educativas y pedagógicas del OA.
- Rights: contiene los derechos de propiedad intelectual y condiciones del uso.
- Relation: presenta información sobre la relación que tiene el OA con otros OA.
- Annotation: posee información sobre el uso del objeto, quién lo creó y cuando.
- Classification: describe al OA de acuerdo a un sistema de clasificación en particular.

¹⁹LOM-IEEE, estándar de LOM propuesto por la IEEE y disponible en http://ltsc.ieee.org/wg12/files/LOM_1484_12.1_v1_Final_Draft.pdf

²⁰Disponible en http://ariadne.cs.kuleuven.be/globe-wiki/index.php/Members_Technical_Information

1. General	Identifier		Catalog Entry
	Title		
	Language		
	Description		
	Keyword		
	Coverage		
	Structure		
2. LifeCycle	Aggregation Level		
	Version		
	Status		
	Contribute	Role	Entity Date
3. Metametadata	Identifier		Catalog Entry
	Contribute		Role Entity Date
	Metadata Schema		
	Language		
4. Technical	Format		
	Size		
	Location		
	Requirement	OrComposite	Type Name Maximun Version MinimunVersion
	Installation Remarks		
	Other Platform Requirements		
	Interactivity Type		
	Learning Resource Type		
5. Educational	Interactivity level		
	Semantic Density		
	Intended End User Role		
	Context		
	Typical Age Range		
	Difficulty		
	Typical Learning Time		
	Description		
	Language		
	6. Rights	Cost	
CopyRight and Other Restrictions			
Description			
7. Relation	Kind		
	Resource	Identifier	Catalog Entry Description
8. Annotation	Entity		
	Date		
	Description		
9. Classification	Purpose		
	Taxon Path	Source	Id
	Description		Taxon Entry
	Keyword		

Figura 5.11: Elementos y estructura del esquema conceptual de LOM.

En la figura 5.11 se muestra a grandes rasgos el contenido de cada una de las clases que están definidas en LOM-IEEE y su ubicación a nivel jerárquico dentro del esquema. La información contenida en estas categorías es utilizada por los buscadores para identificar los elementos relevantes para un usuario; sin embargo, utilizar todos los metadatos en una búsqueda no es práctico, ya que una búsqueda sobre todos los elementos descriptores de los OA en una colección de millones de registros incrementaría los tiempos de respuesta de forma significativa. Por ejemplo, el *AriadneFinder* realiza las búsquedas sobre metadatos específicos a través de un archivo denominado `finder.js` que indica los parámetros a utilizar y dispara las búsquedas contra elementos específicos de los descriptores del objeto; como lo son el formato, idioma y el contexto, dejando de lado otros aspectos que también forman parte de los metadatos disponibles.

5.3.1. Integración del OAI-PMH

El paquete `tds-oai` desarrollado en el TEC Digital, ofrece un cliente para realizar cosecha de metadatos en un servidor OAI-PMH a través de los verbos estándar definidos para el protocolo; por lo que se usó inicialmente como herramienta intermedia para recuperar la información en formato LOM-IEEE asociada a los registros de los OA.

Con el fin de realizar un intercambio exitoso de recursos, en el OAI-PMH se definen una serie verbos estándar para realizar las comunicaciones y que constituyen el protocolo de solicitudes y respuestas entre los agentes y repositorios. Es a partir de estos seis verbos que es posible el intercambio información²¹ :

- *GetRecord*: Recupera un registro de metadatos específico del repositorio a partir de su identificador de ítem y el formato de metadatos que se incluirá en el mismo.
- *Identify*: Recupera información sobre el repositorio. Alguno de los elementos retornados incluyen el nombre del repositorio, la versión del protocolo de OAI-PMH que soporta, URL y datos sobre su granularidad, entre otros.

²¹<http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm#ProtocolMessages>

- *ListIdentifiers*: Es una versión reducida de ListRecords que solo retorna los encabezados de los registros. Entre sus parámetros se puede agregar filtros para seleccionar registros en un rango de fechas, con un tipo específico de formato de metadatos y el parámetro *set* le permite realizar búsquedas bajo ciertos criterios de pertenencia, entre otros.
- *ListMetadataFormats*: Retorna los formatos de metadatos soportados por el repositorio y se puede limitar la búsqueda a un elemento específico dentro del repositorio.
- *ListRecords*: Retorna los registros del repositorio (harvesting) y permite restringir la cosecha con criterios de pertenencia o rangos de fechas.
- *ListSets*: Devuelve la estructura del repositorio. Por ejemplo, los catálogos que contiene, de modo que esta información se puede usar para cosechas selectivas.

Los flujos de control de las solicitudes realizadas en Ariadne son regulados mediante listas incompletas de 100 OA en cada consulta y se administra la secuenciación mediante tokens de reanudación o *resumptionTokens*. Cada uno de estos tokens tiene la función de hacer manejable el flujo de datos entre proveedores y clientes, ya que en repositorios muy grandes la transferencia de miles de registros en una única consulta culmina en sobrecargas durante la transferencia.

Los tokens de reanudación solo se incluyen cuando se deben manejar listas de elementos incompletas, de modo que la concatenación de todos los elementos retornados constituya la lista completa de recursos disponibles. Además, su sintaxis no está especificada por el OAI-PMH, por lo sus parámetros son opcionales, pero se podrían encontrar elementos como la fecha de vencimiento (o tiempo de vida del token), el tamaño de la lista completa disponible en el repositorio (cardinalidad) y el cursor, o número de elementos que fueron retornados ²². Este mecanismo permite realizar consultas sobre un OA único mediante su identificador o sobre catálogos específicos, y recuperar los metadatos disponibles para los OA que se encuentren dentro de este rango.

²² <http://www.openarchives.org/OAI/openarchivesprotocol.html>

Una vez iniciadas las pruebas se hizo evidente que el volumen de metadatos retornados durante la cosecha excedía la capacidad del equipo en el que se ejecuta el prototipo y no se encontró un mecanismo estándar para cuantificar el total de OA que serían recuperados desde el repositorio; ya que como se mencionó anteriormente, la cardinalidad del repositorio es un elemento opcional dentro del resumptionToken y sin este dato, no es posible ni siquiera aproximar un tiempo de respuesta por parte del sistema sin mantener una copia de los OA del repositorio, por lo que se procedió a buscar una alternativa mediante otro protocolo para reducir la cardinalidad de los conjuntos de OA obtenidos; en este caso, se procedió a implementar una interface para el SQL.

5.3.2. Integración de SQL

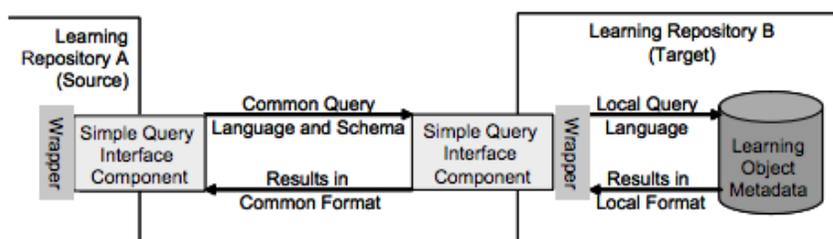


Figura 5.12: Diagrama de interacción entre dos repositorios. Tomado de (Simon y cols., 2005).

Como segundo mecanismo para la recuperación de información, se implementó un cliente para utilizar el servicio de SQL de Ariadne. El SQL fue planteado para las interacciones entre los repositorios asociados a GLOBE y permite realizar búsquedas sobre los metadatos de los OA contenidos en el repositorio, retornando solo aquellos que cumplan la condición dada mediante una consulta. De este modo, el primer filtro aplicado sobre los OA se delega al repositorio en sí, por lo que el volumen de OA se reduce significativamente. La implementación del servicio se resume en trece métodos disponibles mediante servicios web que describen las conexiones sincrónicas y asincrónicas con el

ROA destino, e incluye un método que retorna la cardinalidad de la respuesta dentro de su especificación (Simon y cols., 2005).

El lenguaje para consulta sobre los repositorios está construido en capas y se denomina PLQL, o *ProLearn Query Language*, y se utiliza como formato de intercambio de consultas sobre el protocolo SQL. Esto se logra combinando la búsqueda por patrones de texto exactos o bien, mediante palabras clave.

VSQL	PLQL0
<pre><simpleQuery> <term> Learning Object <term> <term> Database <term> </simpleQuery></pre>	"Learning Object" and "Database"
PLQL1	PLQL2
<pre>lom.general.title = "Relational Database" and lom.educational.intendedEndUserRole="learner"</pre>	lom.general.identifier.(catalog=isbn and entry=1234)

Figura 5.13: Ejemplos de sintaxis de los distintos niveles de PLQL

La figura 5.13 muestra un ejemplo de la sintaxis utilizada en los distintos niveles de PLQL. Según la página referente a los lenguajes de consulta de Ariadne²³, el nivel 0 corresponde a la búsqueda de palabras clave o términos y ofrece el mismo tipo de consultas que el VSQL. El nivel 1 se pueden especificar campos específicos dentro de los metadatos sobre los que se desean realizar las búsquedas; mientras que en el nivel 2 se podrían utilizar paréntesis sobre las rutas. En el caso particular del servicio web que brinda soporte para los colaboradores de Ariadne para consumir servicios mediante SQL, el lenguaje predeterminado está definido para PLQLv1, sin embargo, se puede modificar la configuración deseada para trabajar con otras capas de sintaxis.

Otra ventaja en la definición del protocolo radica en que permite realizar búsquedas sincrónicas y asincrónicas utilizando cuentas anónimas y autenticadas. Para realizar las pruebas con los OA se utilizó una cuenta anónima realizando consultas sincrónicas; ya

²³http://ariadne.cs.kuleuven.be/lomi/index.php/QueryLanguages_stable

que no todos los repositorios implementan la búsqueda asincrónica, la cuál es uno de los principios para realizar búsquedas federadas y exceden los alcances de la investigación. En la práctica, el protocolo ofrece un modelo simplificado de consultas sobre el ROA; sin embargo, no todos los repositorios implementan la totalidad de los métodos al publicar el servicio. De modo que la potencia del protocolo se ve reducida por una subutilización de las funcionalidades definidas. Un ejemplo identificado durante las pruebas fue el comando para modificar la cardinalidad de las respuestas deseadas, el cuál no se encuentra implementado en el servicio web de Ariadne-Partners a Mayo del 2014. Al realizar consultas específicas sobre el repositorio, los conjuntos de respuesta vienen dados en bloques de 12 OA y si se desea recuperar un conjunto mayor de metadatos, se debe recurrir a un modelo de iteraciones sobre listas parciales muy similar al necesario para utilizar el OAI-PMH; de modo que se recae en el problema de las copias parciales del repositorio; con la ventaja de conocer la cardinalidad máxima que puede alcanzar la respuesta.

Para refinar las consultas utilizando SQL, se puede especificar el bloque exacto de metadatos sobre el que se deben cumplir los criterios dados, o bien, se pueden especificar búsquedas abiertas que intenten encontrar coincidencias en cualquier parte de los descriptores. Por esta razón, es importante conocer el estado de los metadatos que se encuentran en el ROA, con el fin de determinar en cuáles casos resulta relevante especificar la ubicación y cuales bloques contienen más información para compararla con las descripciones generadas en el perfil del curso.

5.3.3. Tendencias en la disponibilidad de metadatos

En el estudio empírico realizado por (Najjar, Ternier, y Duval, 2003) sobre 3.900 instancias de OA en ARIADNE, se expone que para esa fecha, solamente los descriptores de granularidad (por ejemplo: lección, capítulo, entre otros) eran utilizados en casi todas las descripciones de los OA, principalmente porque eran un requisito obligatorio en las versiones previas de la herramienta de autoría.

Otros elementos de descripción como el contexto didáctico, el nivel de interactividad,

densidad semántica y nivel de dificultad existían en alrededor del 50 % de las instancias. Además, los OA con texto narrativo, ejercicios, hipertextos y presentaciones representaban la mayoría de los formatos de documentos disponibles en el repositorio.

Según el análisis de metadatos disponible en la wiki de Globe ²⁴ con fecha de su última actualización al 17 de Febrero del 2010, sobre una muestra de 400.000 instancias de OA extraídas de LACLO, ARIADNE y EUN, más del 95 % de los OA contenían los elementos definidos en LOM como: General (Identifier, Title, Language), Technical (Location), Metadata (Identifier, Contributor, Contributor: date, rol, entity) y Rights (CopyrightAndOtherRestrictions, Description); mientras que más del 60 % poseían dentro de su definición los bloques de General (Description, Keywords), Lifecycle (Contributor: date, entity), Educational (LearningResourceType), Rights (Cost) y Classification (Taxon, Purpose).

Estos datos revelan un incremento en la utilización de los descriptores especificados en los OA en general con respecto a su utilización si lo comparamos con el estudio que le precede; sin embargo, el detalle brindado en el sitio, no es suficiente para determinar la calidad de los mismo y sólo se puede asumir su existencia en el XML.

Posteriormente, en (Ochoa, Klerkx, Vandeputte, y Duval, 2011) se realizó otro estudio sobre 630.317 instancias, que representan más del 50 % de los recursos registrados en el consorcio de GLOBE para ese momento y fueron obtenidos mediante harvesting con OAI-PMH. El origen de los OA se dió a partir de la siguiente distribución: ARIADNE Foundation (374.857 instancias), LRE (169.736 instancias), LACLO (49.943 instancias), OER (25.794 instancias), KOCK (7.183 instancias), LORNET (1.804 instancias) y OUJ (1.000 instancias).

En este documento se planteó la existencia de distintos orígenes de datos para los repositorios, así como de importantes diferencias en cuanto a la forma de administrar los metadatos.

Una de las principales diferencias mencionadas radica en que a pesar de que muchos de los repositorios incluyen recursos de otros ROA, existen casos como LRE, OER y LOR-

²⁴<http://ariadne.cs.kuleuven.be/globe-wiki/index.php/MetadataAnalysis>

NET que enriquecen los metadatos de los OA recuperados (de forma manual, automática o semiautomática) para mantener el estándar de sus repositorios; o bien, el caso de LACLO, el cuál mencionan que construye todos sus metadatos de forma automatizada (Ochoa y cols., 2011).

En cuanto a la utilización de las distintas secciones definidas para un OA en LOM, los autores realizan la aclaración de que para el momento de la publicación (2011) muchos de los ROA no utilizan el estándar en su máxima extensión, por lo que es posible encontrar bloques agregados dentro del nodo padre, lo cuál se ve reflejado en el análisis de frecuencia de aparición de cada uno de los metadatos que se presentan en el artículo.

Otro dato relevante es que según su análisis de frecuencia, más del 90 % de la muestra de datos incluían los metadatos asociados a General (Title, Identifier, Language), Technical (Location) y Meta-metadata (Identifier), mientras que si se amplía el margen de tolerancia a más del 60 % de las muestras, aparecen metadatos como General (Description), Lifecycle (Contribute: Role, Entity), Educational (LearningResourceType), Meta-metadata (Contribute : Role, Entity), Rights (CopyRight, Cost, Description, taxonPath) y Classification (TaxonPath, Purpose).

En este estudio, los autores presentan una gráfica en la que expone que solo entre el 40 % y 50 % de las muestras incluían algún tipo de información referente al contexto educativo, el tipo de usuario para el que fueron diseñados o su edad. De este bloque de instancias con algún tipo de información educacional, el 40 % incluía el tipo de recurso, el rol de usuario para el que fue diseñado, contexto y edad. Por otro lado, más del 10 % ofrecía datos sobre los niveles y tipos de interactividad e idiomas y menos del 10 % aportaba algún tipo de detalle sobre la dificultad, densidad y tiempo.

En cuanto a la disponibilidad de recursos por idiomas en particular; los repositorios han atravesado por una serie de cambios donde los autores plantean la incorporación de nuevos colaboradores como causa de la diversificación. Incluso mencionan el problema existente con las nomenclaturas para especificar el idioma en los metadatos.

La definición formal de LOM, establece que los idiomas deben ser dados siguiendo las

normas del ISO 639:1988 con el subcódigo de país definido en el ISO 3166-1:1997, y el valor “none” sólo es aceptado en algunas secciones (*Draft Standard for Learning Object Metadata*, 2002); por lo que es posible encontrar “spa”, “es”, “es-CR”, “es-ES”, entre otros valores reconocidos para representar el idioma español.

Estas diferencias se vuelven importantes al intentar recuperar la cardinalidad de una respuesta mediante SQL; ya que los valores obtenidos cuando se especifica la búsqueda para un metadata específico, difieren significativamente de los obtenidos cuando no se especifica donde buscar.

La figura 5.14 refleja estas diferencias en cuanto a la cardinalidad de respuestas. La

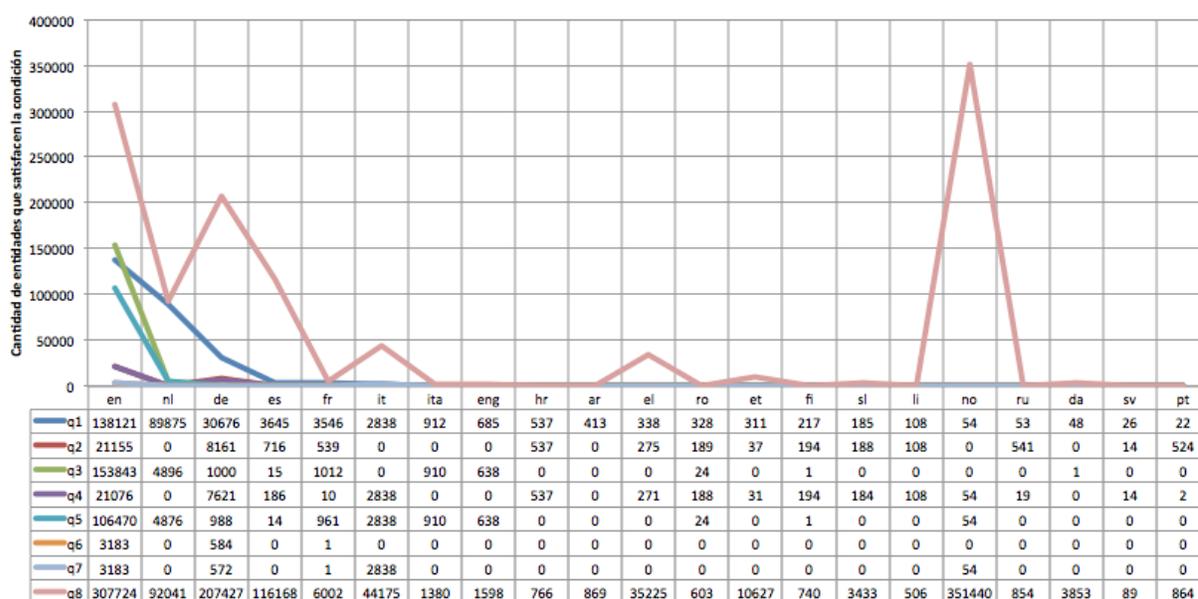


Figura 5.14: Elementos más frecuentes al especificando la búsqueda sobre los metadatos de idioma utilizando el ISO 639-1 e ISO 639-2 como criterio de búsqueda

gráfica representa los 20 valores que aparecen con mayor frecuencia al lanzar consultas mediante el protocolo SQL apuntando hacia las colecciones de ARIADNE Partners y utilizando las definiciones brindadas por el ISO 639-1 e ISO 639-2 (correspondientes al alpha2 y alpha 3 del del ISO 639:1988), así como la cardinalidad de una consulta lanzada utilizando esa misma etiqueta de idioma, si no se especifica el bloque de metadatos en el

que se desea restringir la búsqueda.

Los distintos valores de q representan las consultas sobre secciones específicas del LOM de modo que $q1$ representa una búsqueda sobre `lom.general.language`, $q2$ está asociado a una búsqueda sobre `lom.educational.language` y $q3$ al `lom.metametadata.language`. Mientras que de $q4$ a $q7$, se representan los elementos que se encuentran en las intersecciones de estos conjuntos y el $q8$ coincide con la aparición de esa secuencia de texto en cualquier parte de los metadatos como se muestra en la figura 5.15.

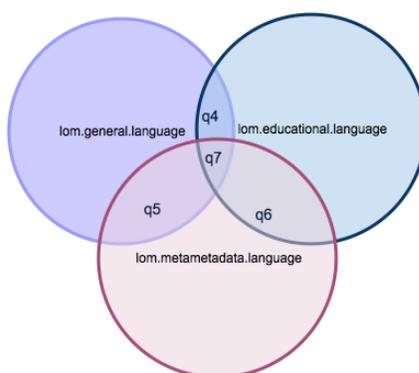


Figura 5.15: Diagrama de Venn con la representación de las consultas realizadas mediante SQL al repositorio.

Para Mayo del 2014, Ariadne reporta 830.297 instancias de OA en el Finder, sobre las cuales se realizaron consultas para extraer los idiomas registrados. De las instancias con el rubro `lom.general.language` en español, se identificaron 3.645 OA con la etiqueta “es” y 106.710 con la etiqueta “es-EC”. Para los otros grupos que contienen un descriptor de idioma (Educational y Meta-metadatos), solamente se encontraron recursos bajo la etiqueta “es”.

Por otro lado, dentro de las muestras de datos recuperados se identificaron OA etiquetados como objetos en español, cuyas descripciones venían dadas en otros idiomas; por lo que la aplicación de una consulta específica para OA etiquetados en español, no es una garantía de que la respuesta será completamente confiable.

5.3.4. Preselección de los OA

Para la preselección de los OA que se van a evaluar se tomaron en consideración tres premisas básicas:

- Los OA recuperados deben ser afines a las etiquetas extraídas al construir el perfil de las actividades del curso.
- No es razonable intentar procesar todos los OA que responden a una etiqueta, por lo que debemos intentar maximizar la afinidad de los OA desde la fase de preselección.
- La aplicación de un filtro de idioma explícito no garantiza que los OA retornados realmente van a estar dados en dicho idioma y un mismo término puede existir en múltiples idiomas.

Para controlar las dimensiones de los conjuntos a evaluar, se definieron dos parámetros para definir topes de la cardinalidad de los conjuntos de OA: el primero asociado a la cantidad máxima de elementos que se van a recuperar por cada etiqueta, de modo que si el conjunto de OA que se podría recuperar excede este valor; se procedería a modificar la consulta para ajustar las dimensiones del mismo y en última instancia, se truncaría el conjunto de OA que satisfacen la etiqueta. Mientras que el segundo parámetro, está asociado a la cantidad máxima de OA que serán retornados como respuesta al final del proceso de evaluación y recomendación.

Antes de realizar la recuperación de los OA, se realiza una serie de consultas para identificar aquellas que podrían retornar algún resultado con respecto al perfil de la actividad, con lo que se vuelve a aplicar una reducción de las expresiones que lo representan. En esta ocasión, los elementos no son descartados del perfil y solamente se remueven de la lista de consultas. En caso de encontrar expresiones para las que se exceda la cantidad de elementos máxima a recuperar, se procede a formar expresiones, utilizando la sintaxis de PLQL que se presentó en la sección 5.3.2, para buscar sólo aquellos OA que se encuentren en los puntos de intersección de dos o más expresiones.

```

Cardinalidad de Respuestas: {bases 12744} {introducción 13} {programa+del+curso 86} {programas 2131}
Textos que exceden la cardinalidad maxima esperada ( 150 OAs ): bases , programas
Intersecciones de Respuestas: { {bases and introducción} 3} { {bases and programas} 50} { {programas and introducción} 1} { {programas and
programa+del+curso} 86} { {programas and bases} 50}

```

```

Al Jufra Air Base
Termino buscado: bases
ID> oai:oai.mace-project.eu:mace:external:a0cb744983160b83cda83eacddf24420 Catalogo> oai
ID> mace:external:a0cb744983160b83cda83eacddf24420 Catalogo> mace:external

Pelican Seaplane Base
Termino buscado: bases
ID> oai:oai.mace-project.eu:mace:external:0966afa2688d32e01fb2caff7cafad5b Catalogo> oai
ID> mace:external:0966afa2688d32e01fb2caff7cafad5b Catalogo> mace:external

RAAF WilliamsRAAF Base WilliamsBases aériennes Williams
Termino buscado: bases
ID> oai:oai.mace-project.eu:mace:rwo:9b520834341db6e70687d25d80077e6b Catalogo> oai
ID> mace:rwo:9b520834341db6e70687d25d80077e6b Catalogo> mace:rwo
ID> http://dbpedia.org/resource/RAAF_Williams Catalogo> dbpedia

Al Jufra Air Base
Termino buscado: bases
ID> oai:oai.mace-project.eu:mace:rwo:a0cb744983160b83cda83eacddf24420 Catalogo> oai
ID> mace:rwo:a0cb744983160b83cda83eacddf24420 Catalogo> mace:rwo
ID> http://dbpedia.org/resource/Al_Jufra_Air_Base Catalogo> dbpedia

```

Figura 5.16: Muestra de datos obtenida al realizar la búsqueda sin el filtro por idioma para la actividad 1.

La figura 5.16 muestra un fragmento de los metadatos de los OA recuperados al realizar una búsqueda de materiales para la primera actividad del DI de pruebas. En esta búsqueda no se aplicó ningún tipo de filtro y solamente se recuperaron aquellos OA que contienen los textos identificados como parte del perfil del curso. Como se puede apreciar, la ausencia de un filtro de idioma ha ocasionado que los recursos recuperados se encuentren en idiomas distintos al español, a pesar de que los textos brindados como parámetro respondían a este idioma.

Al final del proceso de preselección de OA, se descartan aquellos que estén duplicados ya que un mismo OA puede contener varias etiquetas dadas para un perfil y los elementos restantes son los que se procederían a evaluar para generar la recomendación.

5.4. Generación de la recomendación

En esta sección se procederá a describir el proceso utilizado para generar el ranking de los OA recuperados del repositorio y los resultados obtenidos de la aplicación de los algoritmos utilizados.

Para evaluar los OA y generar las recomendaciones se descompuso el problema de hi-

bridación en dos fases, la primera en un mecanismo de recomendación en cascada a partir de una recomendación basada en contenidos y otra colaborativa; mientras que la segunda está centrada en un despliegue de información mixto para incorporar la recomendación por estilos de aprendizaje.

5.4.1. Hibridación en cascada

El proceso de hibridación en cascada consiste en utilizar la salida de un recomendador como entrada del segundo (Burke, 2007). La hipótesis que se maneja para realizar esta hibridación, es que al trabajarse con contenidos textuales se puede aproximar la similitud que mantienen los OA con respecto a la descripción de la actividad y con la navegación del usuario, se puede mejorar dicha aproximación para identificar elementos que fueron excluidos durante la fase de construcción del perfil del curso.

5.4.1.1. Recomendación basada en contenidos

Para construir este recomendador se marcó como objetivo aprovechar el contenido textual disponible en los metadatos de los recursos para definir una serie de atributos que los describan.

Tomando como referencia la disponibilidad de contenidos en los metadatos definidos para LOM que fueron descritos en la sección 5.3.3, se eligieron como descriptores el título del OA y las descripciones brindadas por las secciones de General y Educational, sin embargo no se descarta la posibilidad de ampliar las entradas de datos.

Para estimar el grado de afinidad entre los textos se utilizó primeramente la similitud de cosenos; de modo que se construyeron los vectores descriptores de cada OA y se realizó una comparación utilizando esta medida con los tokens como parámetro de referencia.

Para establecer los tokens a conservar para cada OA, se procedió a aplicarles el mismo proceso al que fueron sometidas las actividades de aprendizaje para extraer el conteo de tokens; de modo que para cada uno de ellos solamente se recuperaron los tokens que

sobrevivieron al proceso de poda y extracción del palabras de parada que fue definido para la construcción de los perfiles.

Una vez calculado el ángulo de la descripción de cada OA, con respecto a la descripción de la actividad, se seleccionaron aquellos top-N más cercanos y se procedió a realizar un análisis por frases, con el fin de refinar el ranking generado por esta primer clasificación, usando como N, un parámetro definido dentro de la base de datos para restringir la cardinalidad de la respuesta final esperada.

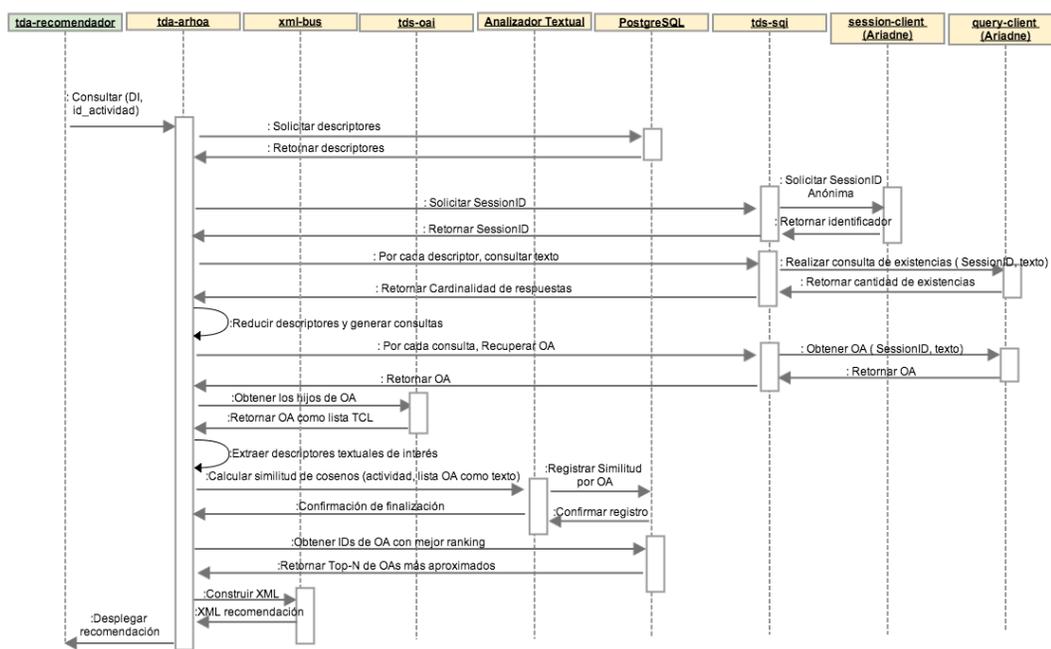


Figura 5.17: Diagrama de secuencia para recomendación por contenidos

La figura 5.17 representa el proceso que se desencadena cuando se solicita calcular una recomendación para una actividad para la que ya han sido registrados descriptores a partir del calculo del perfil de un DI particular.

A grandes rasgos, el proceso se puede resumir de la siguiente forma:

- Recuperar el bloque del perfil: Consiste en buscar las palabras o frases que se extrajeron para describir la actividad durante la construcción del perfil del curso. Estos

textos son los que se utilizarán para realizar las búsquedas contra el repositorio.

- **Iniciar sesión:** Para realizar cualquier tipo de consulta, ya sea sincrónica o asincrónica en un ROA con servicio de SQL se debe contar con identificador de sesión. Este número es el que nos permite autenticarnos en el repositorio y es requerido para cada consulta o comando que se desee ejecutar, además, un mismo identificador es válido para toda la sesión de trabajo que se ejecute.
- **Preselección de OA:** Como se mencionó en el capítulo de diseño, no se está trabajando con diccionarios de dominio específico; de modo, que muchas de las frases que se conservan para un perfil pueden carecer de relevancia al realizar una búsqueda. Se incluyó una validación a partir de una de los métodos del SQL para retornar la cardinalidad esperada al realizar una consulta y está descrito en la preselección de OA.
- **Extraer los descriptores y calcular similitudes:** Una vez recuperados los OA de la preselección, se procedió a descomponer el bloque de LOM-IEEE que lo conforma para extraer los textos que aportan una descripción textual del OA. Para esta tarea se consumió una función del tds-oai para extraer los hijos de un documento en formato LOM-IEEE y retornarlos en forma de lista de TCL; posteriormente se extrajeron los elementos de interés para ser analizados. En este caso, se recuperó el título de OA y las descripciones del bloque de General y Educational. Para cada OA se realizó la extracción de atributos para construir un vector que lo describa y se comparó mediante el cálculo de similitud de cosenos contra el vector que describe la actividad que está en evaluación.
- **Recuperar los elementos más afines:** Una vez finalizados los cálculos con la similitud de cosenos se procede a seleccionar solo aquellos top-N elementos que ofrezcan mayor similitud con la descripción de la actividad y se construye un documento para ser desplegado en pantalla a partir de las funciones del paquete xml-bus.

Después de aplicar la similitud de cosenos sobre los metadatos de los OA, los que quedaron dentro del top-N venían dados en español. Esto es relevante porque no fue indicado explícitamente en la consulta, pero el conjunto de palabras que describen a la actividad traen implícito el idioma en su definición; de modo que las descripciones de los OA más afines cumplirán esta misma condición y esto representa una importante diferencia en cuanto a los resultados obtenidos por la preselección de OA sin ningún criterio de filtro. Como parte de las dificultades técnicas identificadas para calcular la similitud, se encontró que algunos de los textos de las descripciones venían dados en formato URL UTF-8, de modo que aparecen caracteres en hexadecimal que debían ser modificados para ser asociados a las dimensiones apropiadas.

Además, dentro de los recursos que retornó el ROA se encuentran elementos que satisfacen un patrón de texto, por ejemplo, “base” y “bases” nuevamente. De modo que los OA que contengan uno u otro valor van a obtener distintas medidas de similitud que no serán las más aproximadas al modelo original. Esto se puede validar al encontrar similitudes de coseno iguales a cero generadas para OA que se recuperaron bajo un criterio de texto existente dentro del perfil. La figura 5.18 muestra un ejemplo de esta situación en la que OA que ya pasaron por un proceso de preselección caen a cero tras un cálculo de similitud de cosenos, de modo que se excluyen del top-N que va a pasar como entrada al siguiente recomendador.

```

catalog ESPOL entry 67032 > 0.002709738551294974 >> angulo >> 89.84474341742691--28.697309522148355---1.5680865849274654
Fallo al registrar coseno para catalog ESPOL entry 67032
catalog ESPOL entry 67839-esp0l > 0.004039401286610375 >> angulo >> 89.76855935451752--28.7215598024067---1.5667569145232134
Fallo al registrar coseno para catalog ESPOL entry 67839-esp0l
catalog ESPOL entry 90717-esp0l > 0.0 >> angulo >> 90.0--28.64788975654116---1.5707963267948966
Fallo al registrar coseno para catalog ESPOL entry 90717-esp0l
catalog ESPOL entry 90717 > 0.0 >> angulo >> 90.0--28.64788975654116---1.5707963267948966
Fallo al registrar coseno para catalog ESPOL entry 90717
catalog ESPOL entry 95032-esp0l > 0.0013225847188936494 >> angulo >> 89.9242214775589--28.672010816426777---1.5694737416904185
Fallo al registrar coseno para catalog ESPOL entry 95032-esp0l
catalog ESPOL entry 96756 > 0.0 >> angulo >> 90.0--28.64788975654116---1.5707963267948966
Fallo al registrar coseno para catalog ESPOL entry 96756

```

Figura 5.18: Muestra de resultados al calcular similitud de cosenos a los descriptores de OA.

La primer recomendación de tipo top-N para los OA usando la similitud de coseno como referencia no es suficiente para decir que los recursos recuperados podrían ser de utilidad

para una actividad de este DI. Siguiendo con el ejemplo de la actividad 1, en cuyo caso la descripción es:

”Presentación de conceptos de datos e información.

Introducción a Bases de datos.

Entrega y presentación del programa del curso.

Presentación.”

Se obtuvieron una serie de OA relacionados con el tema de bases de datos; sin embargo, las aproximaciones distan mucho de ser las más certeras para la descripción textual brindada por el profesor para la actividad.

	Título	Similitud
1	Base de Datos MÃ°sica (DATOS%2BBASE%2BDE%2BDATOS%2BMUSICA.xls)	0.36265925
2	Base de Datos MÃ°sica (Base%2Bde%2BDatos%2Bde%2BM%25C3%25BAAsica-VARIOS.mdb)	0.33084658
3	Base de Datos MÃ°sica (Base%2Bde%2BDatos%2Bde%2BM%25C3%25BAAsica-SIN%2BRELACIONES.mdb)	0.31429362
4	Nueva InformaciÃ°n para base de datos MÃ°sica (DATOS%2BBASE%2BDE%2BDATOS%2BMUSICA.xls)	0.26352313
5	BASES DE DATOS HISTÃ°RICOS: ITALIA (MILAN+2001.zip)	0.22821774
6	BASES DE DATOS HISTÃ°RICOS: ITALIA (MILAN+2001.zip)	0.20080483
7	Base de Datos: ACCESS 2003 (ACCESS%2B2003.ppt)	0.19756584
8	Base de Datos MÃ°sica (Base%2Bde%2BDatos%2Bde%2BM%25C3%25BAAsica-UNO.mdb)	0.1860521
9	BASES DE DATOS HISTÃ°RICOS: ITALIA (MILAN+2001.zip)	0.15655607
10	ImplementaciÃ°n de Base de Datos (SQLServer_Cap2_Desarrollo.pdf)	0.14344382

Figura 5.19: Top-10 OA sugeridos para la actividad 1 del DI.

5.4.1.2. Recomendaci3n colaborativa

La aplicaci3n del algoritmo de TF-IDF puede ser considerado como un algoritmo propio de entornos basados en contenidos (Adomavicius y Tuzhilin, 2005) o de entornos colaborativos (Burke, 2007) segun la propuesta del autor.

El principio te3rico detr3s de dicho algoritmo permitir3a determinar cuales palabras son m3s significativas para un documento que para otro y estar3a recibiendo como entrada el resultado de la aplicaci3n del algoritmo de similitud de cosenos para priorizar los OA .

Para aplicar este algoritmo sobre los OA que fueron reorganizados mediante la aplicaci3n de similitud de cosenos, se le di3 prioridad dentro del subconjunto de elementos a aquellos que contaran con una sumatoria de valores mayor. Para esto se extrajo una 3nica

aparición de cada término contenido dentro del OA que fuera correspondiente a otro del perfil; de modo, que sólo se contabilizó una aparición de cada término por OA.

Como mecanismo auxiliar se utilizó la medida de similarity y de Levenshtein de PostgreSQL, ya que en la similitud de cosenos se encontró que se estaban ignorando dimensiones por detalles de escritura. La figura 5.20 muestra los primeros 15 elementos

Identificador	Similitud de Coseno	Sumatoria(TF-IDF)	Coincidencias	Descripción del OA
catalog ESPOL entry 265276	0.036625993	2.03957707683245	4	Introducción a la base de datos (Bases+de+Datos+Espaciales.ppt)
catalog ESPOL entry 265276-espol	0.03717829	2.03957707683245	4	Introducción a la base de datos (Bases+de+Datos+Espaciales.ppt)
catalog ESPOL entry 26547	0.041939847	1.84017106890678	4	Conceptos básicos de Bases de Datos Relacionales (INTRODUCCION++A+BASES+DE+DATOS.ppt)
catalog ESPOL entry 26547-espol	0.04255097	1.84017106890678	4	Conceptos básicos de Bases de Datos Relacionales (INTRODUCCION++A+BASES+DE+DATOS.ppt)
catalog ESPOL entry 122899	0.12931515	1.64076505899429	2	Implementación de Base de Datos (SQLServer_Cap3_Desarrollo.pdf)
catalog ESPOL entry 122899-espol	0.12309149	1.64076505899429	2	Implementación de Base de Datos (SQLServer_Cap3_Desarrollo.pdf)
catalog ESPOL entry 122898	0.14344382	1.64076505899429	2	Implementación de Base de Datos (SQLServer_Cap2_Desarrollo.pdf)
catalog ESPOL entry 122898-espol	0.13608277	1.64076505899429	2	Implementación de Base de Datos (SQLServer_Cap2_Desarrollo.pdf)
catalog ESPOL entry 112545	0.26352313	1.16031204164028	2	Nueva Información para base de datos Música (DATOS+BASE+DE+DATOS+MUSICA.xls)
catalog ESPOL entry 122377	0.069212854	1.16031204164028	2	Base de datos (hairdata.zip)
catalog ESPOL entry 122377-espol	0.0745356	1.16031204164028	2	Base de datos (hairdata.zip)
catalog ESPOL entry 112026	0.36265925	0.679859027266502	2	Base de Datos Música (DATOS+BASE+DE+DATOS+MUSICA.xls)
catalog ESPOL entry 1126-espol	0.13975425	0.679859027266502	2	BASES DE DATOS HISTÓRICOS: ITALIA (MILAN+2002.zip)
catalog ESPOL entry 1126	0.12578864	0.679859027266502	2	BASES DE DATOS HISTÓRICOS: ITALIA (MILAN+2002.zip)
catalog ESPOL entry 1125	0.15655807	0.679859027266502	2	BASES DE DATOS HISTÓRICOS: ITALIA (MILAN+2001.zip)

Figura 5.20: Aplicación de Hibridación en cascada

obtenidos tras aplicar la hibridación para la actividad 1. La columna de Sumatoria (TF-IDF) muestra el valor total que se obtuvo para el OA tras sumarizar los valores de TF-IDF de los tokens del perfil que contiene. Estos valores fueron generados para cada palabra a partir de la colección total de actividades, de modo que un término que aparece solamente en una actividad, tendrá un valor mayor que otros que son compartidos por varias actividades. La columna de coincidencias contiene la cantidad de términos contenidos en ese OA tras aplicar la comparación con las herramientas de PostgreSQL.

La muestra de resultados reorganizada indicó que los OA con mayor afinidad para la actividad 1 están asociados al tema “Introducción a las bases de datos” y “Conceptos básicos de bases de Datos Relacionales”; sin embargo, si se aplica la similitud de cosenos con un tolerancia a errores, los conjuntos de resultados pueden variar. Para probar las diferencias de resultados en cuanto al orden de aplicación de los algoritmos, las figuras 5.21 y 5.22 muestran los resultados tras dar prioridad en distinto orden a los mecanismos involucrados. La figura 5.21 muestra una priorización de la cantidad de coincidencias que se hallaron con respecto al perfil de la actividad. En este caso, los OA más afines aún giran en torno al tema de bases de datos seguidos por los que se asocian a los programas de

Descripción del OA

Introducción a la base de datos (Bases+de+Datos+Espaciales.ppt) .
 Introducción a la base de datos (Bases+de+Datos+Espaciales.ppt) .
 Conceptos básicos de Bases de Datos Relacionales (INTRODUCCION++A+BASES+DE+DATOS.ppt) .
 Conceptos básicos de Bases de Datos Relacionales (INTRODUCCION++A+BASES+DE+DATOS.ppt) .
 Introducción a Bases de datos (Introducc%C3%B3n+a+Base+de+Datos.pptx) . Conceptos de bases de datos
 Introducción a Bases de datos (Introducc%C3%B3n+a+Base+de+Datos.pptx) . Conceptos de bases de datos
 Políticas y Programa del Curso + Temas de Proyectos (programa_Course_Graduacion_detallado.pdf) .
 Políticas y Programa del Curso + Temas de Proyectos (programa_Course_Graduacion_detallado.pdf) .
 Nuevo (Chapter_09.ppt) .
 Nuevo (Chapter_09.ppt) .
 PROGRAMA DEL CURSO (Escalado+de+Procesos+Industriales.docx) .
 PROGRAMA DEL CURSO (Escalado+de+Procesos+Industriales.docx) .
 Programa del curso (Políticas+Alumnos+Ing.+Economica+I.pdf) .
 Programa del curso Ingeniería Económica I (Políticas+Alumnos+Ing.+Economica+I.pdf) .
 Programa de Curso (PROGRAMA+DE+CURSOS_PROD+1_I_2011.doc)

Descripción del OA

Políticas y Programa del Curso + Temas de Proyectos (programa_Course_Graduacion_detallado.pdf)
 Políticas y Programa del Curso + Temas de Proyectos (programa_Course_Graduacion_detallado.pdf)
 Nuevo (Chapter_09.ppt) .
 Nuevo (Chapter_09.ppt) .
 PROGRAMA DEL CURSO (Escalado+de+Procesos+Industriales.docx) .
 PROGRAMA DEL CURSO (Escalado+de+Procesos+Industriales.docx) .
 Programa del curso (Políticas+Alumnos+Ing.+Economica+I.pdf) .
 ProgramaCurso (PRODUCCION_Y_OPERACIONES_AMBIENTALMENTE_SUSTENTABLES_1_doc
 Programa del curso (Políticas+Alumnos+Ing.+Economica+I.pdf) .
 PROGRAMA DE ING. ECONOMICA I (INGENIERIA+ECONOMICA+I.doc) .
 programas de estudio y políticas del curso (POLITICAS_pui.doc) .
 Programa de Curso (PROGRAMA+DE+CURSOS_PROD+1_I_2011.doc) .
 Programa del Curso (Programa+Intro+Micro+V2008.pdf) .
 Programa del Curso (programa+economia+de+los+mercados+y+regulacion+alicia+guerrero.doc) .
 Programa del curso (Syllabus+Administracion+de+operaciones.doc) .

Figura 5.21: Priorización de coincidencias

entre perfiles con ajuste de textos.

Figura 5.22: Priorización del TF-IDF

curso. En el caso de la figura 5.22 , la priorización se dió con respecto a los valores de TF-IDF encontrados; en este caso, aquellos OA que contengan palabras únicas para la actividad van a quedar mejor posicionados dentro del ranking. Por este motivo, los OA relacionados con los programas de curso consiguieron mejores posiciones con respecto a los que podrían ser más afines a otros contenidos de la actividad.

Las variaciones en los resultados obtenidos muestran que aún es necesario un ajuste que permita refinar los procesos de recomendación; principalmente si se toma en cuenta que la muestra de datos utilizada pertenece a un único dominio en particular.

5.4.2. Hibridación mixta

El proceso de hibridación mixta consisten en brindar al usuario una recomendación con los resultados por separado de dos sistemas recomendadores (Burke, 2007).

La representación de esta hibridación se dará a partir del despliegue de la hibridación en cascada en conjunto con el despliegue de resultados de la clasificación por estilos de aprendizaje.

5.4.2.1. Clasificación por estilos de aprendizaje

Para la recomendación por estilos de aprendizaje se construyó un clasificador a partir de los principales tipos de recursos encontrados en el repositorio.

Como herramienta para adquirir el conjunto de entrenamiento inicial, se utilizó una en-

cuesta enfocada hacia profesionales con conocimientos en el área de estilos de aprendizaje; donde se presentó una serie de tipos de recursos y se le solicitó al experto en la materia clasificarlos de acuerdo al posible aprovechamiento de cada uno de ellos con respecto a los distintos estilos de aprendizaje. Esta herramienta constó de cinco preguntas, donde se presentó cada una de las dicotomías de Felder y Silverman, sin embargo tras la apertura de la encuesta sólo se contó con la colaboración de un experto en la materia para facilitar los datos solicitados.

Seguidamente se procedió a extraer la información recopilada en limesurvey en formato CSV para cargarlo sobre WEKA y realizar un análisis de los resultados obtenidos.

Producto de estos datos recopilados, se generaron registros que fueron destinados a

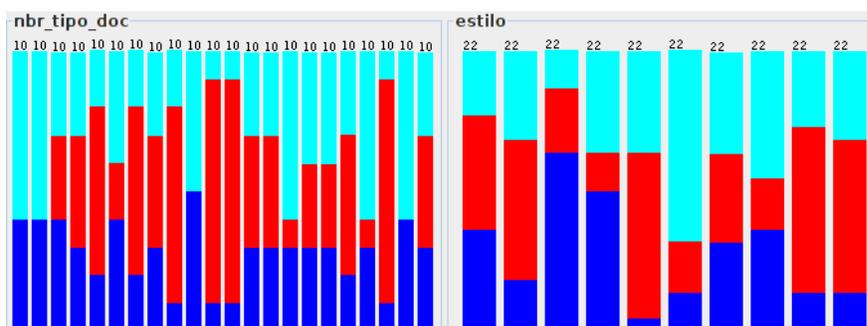


Figura 5.23: Gráfica de utilidad de los recursos generado con WEKA.

construir el modelo de entrenamiento para el recomendador a partir de 22 categorías de documentos separados cada una en las 5 dicotomías.

La figura 5.23 está dividida en dos secciones; la primera corresponde a la utilidad general que le fue dada a cada categoría de documento; mientras que la segunda parte muestra un resumen de la utilidad total que se encontró en las categorías por cada estilo de aprendizaje. La clase representada por los colores corresponde a tres posibles valores de utilidad, que en este caso serían: bajo (azul), medio (rojo) y alto (cian).

Estos datos se utilizaron para generar un modelo de entrenamiento que se aplicó posteriormente a los OA recuperados por el agente, donde inicialmente se consideró la posibilidad de utilizar un árbol de decisión J48 para realizar la clasificación de los recursos. Esta

primer prueba retornó una tasa de aciertos muy baja (alrededor del 54% en el mejor de los casos), por lo que se procedió a analizar otros posibles clasificadores que obtuvieran mejores rendimientos.

Para seleccionar el clasificador se analizaron los resultados de aplicar distintos clasificadores sobre el conjunto de entrenamiento disponible y se seleccionaron aquellos con mejores rendimientos para proceder a probarlos con un subconjunto de datos de prueba. La figura 5.24 muestra los resultados obtenidos mediante la aplicación de clasificadores

Tipo de Clasificador	Clasificador	% Aciertos	% Errores	MAE	RMSE	% RAE	%RRSE	T (s)
Bayes	AODEsr	100	0	0.0687	0.0741	15.5521	15.7732	0
	AODE	99.5455	0.4545	0.2933	0.3168	66.4017	67.405	0
	Red Bayesiana	64.5455	35.4545	0.3292	0.392	74.518	83.4137	0
	HNB	100	0	0.0179	0.0263	4.0499	5.604	0.01
	naïve Bayes	64.5455	35.4545	0.3392	0.3952	76.783	84.1045	0
	WAODE	100	0	0.0724	0.0784	16.3808	16.6918	0
	BF (Best First)	73.6364	26.3636	0.208	0.3225	47.0778	68.6161	0.13
	DecisionStump	40.9091	59.0909	0.4275	0.4623	96.7853	98.3836	0.35
	Functional Trees	85.4545	14.5455	0.1237	0.2576	28.0045	54.8243	0.12
	ID3	100	0	0	0	0	0	0
Árboles	J48	54.0909	45.9091	0.3685	0.4292	83.4209	91.3388	0
	LAD	56.3636	43.6364	0.3536	0.4148	80.0607	88.2619	0.06
	LMT	61.8182	38.1818	0.3541	0.4072	80.1544	86.6493	0.49
	NB	82.2727	17.7273	0.2846	0.3381	64.4411	71.9397	0.14
	RandomForest	100	0	0.0989	0.1324	22.397	28.1673	0.01
	RandomTree	100	0	0	0	0	0	0
	SimpleChart	80.4545	19.5455	0.1834	0.3029	41.5308	64.4471	0.11
	DecisionTable	52.7273	47.2727	0.3914	0.4411	88.6118	93.8565	0
	DTNB	64.5455	35.4545	0.339	0.3952	76.7504	84.106	0.01
	Reglas	Nnge	100	0	0	0	0	0
OneR		54.0909	45.9091	0.3061	0.5532	69.2887	117.7238	0
PART		58.1818	41.8182	0.3411	0.413	77.2246	87.8812	0.02
PRISM		100	0	0	0	0	0	0.02

Figura 5.24: Tabla de rendimiento de los clasificadores

de tipo bayesiano, basados en reglas y mediante árboles de decisión. Los datos presentados corresponden a aquellos algoritmos cuya tasa de aciertos superó el 50% y se incluyeron otras mediciones correspondientes a los márgenes de error que fueron previamente mencionadas en el capítulo de diseño. Además, sobre ellos se aplicó un parámetro que indica que se debe utilizar el mismo conjunto que lo generó como conjunto de entrenamiento; así que los resultados obtenidos son los asociados al mejor caso posible.

A partir de esta tabla podemos extraer que los mejores exponentes dentro de esta lista de clasificadores tomando solamente en cuenta el porcentaje de aciertos están dados

por los clasificadores bayesianos AODEsr²⁵, HNB²⁶ y WAODE²⁷, por los árboles ID3²⁸, RandomForest y RandomTree y los clasificadores basados en reglas PRISM y Nnge²⁹. Por las características del conjunto de datos que vamos a evaluar en los OA no se tiene previsto la incorporación de incógnitas ajenas al conjunto de entrenamiento, ni se considera necesario un ajuste exhaustivo de las ramas del clasificador. Tomando en cuenta las métricas con márgenes de error más bajos y que los algoritmos de clasificación basados en reglas se generan a partir de árboles de decisión, se eligió ID3 para construir el modelo de clasificación final. Este algoritmo siempre cubre el 100 % de los casos identificados en el conjunto de entrenamiento con éxito, y además sirvió de base para otros algoritmos como el J48 y PRISM.

5.4.3. Recomendación por Competencias

En cuanto a la recomendación por competencias se intentó aplicar la misma metodología utilizada para el análisis de los contenidos del DI; sin embargo se llegó a un agujero entre lo que la teoría reconoce como correcto y las posibilidades de aplicarlo a nivel práctico al intentar reflejar un objetivo en términos de contenidos.

A grandes rasgos, la definición de una competencia o habilidad está asociada a los objetivos que debería alcanzar un estudiante al realizar una actividad o al finalizar sus programas de estudios; pero la información disponible dentro de la definición de los metadatos a nivel educativo es escasa y no fue posible identificar un patrón o solución para traducir un esquema basado en objetivos a otro basado en contenidos utilizando LOM.

En (Baldiris, Santos, Boticario, y Fabregat, 2008; Baldiris, Santos, Barrera, y cols., 2008) se modelan los conocimientos por objetivos a partir de la taxonomía de Bloom utilizando

²⁵AODEsr, Acrónimo de Average One-Dependence Estimator Subsumption Resolution

²⁶HNB, Acrónimo de Hidden Naïve Bayes

²⁷WAODE, Acrónimo de Weightily Average One Dependence Estimator

²⁸ID3, Acrónimo de Iterative Dichotomiser 3

²⁹Nnge, Acrónimo de Nearest Neighbour using non-nested generalized exemplars

IMS-LIP³⁰ e IMS-RCDEO³¹, pero para cada recurso utilizado en la adaptación se modelaron manualmente los niveles de conocimiento esperados y el estilo de aprendizaje; de modo que este enfoque no se podría aplicar dinámicamente en el modelo de recomendación esperado.

La definición del IMS-RCDEO provee referencias a las descripciones de las competencias para ser intercambiadas entre diversos sistemas como los LMS y permite su extensión a partir de la incorporación de modelos en XML. Además, incluye documentación específica segmentada en distintos manuales donde se incluyen rubros para describir la accesibilidad, guías para la interoperabilidad entre herramientas y reutilización de competencias, entre otras.

En las guías de información del modelo, se mencionan cinco elementos que lo constituyen:

- Identifier: Es un identificador único que sigue la misma estructura de LOM (Catalogue, Entry) para referenciar las competencias en cualquier sistema.
- Title: Es un texto entendible por humanos para etiquetar la competencia y puede venir en varios idiomas.
- Description: Es un texto opcional entendible por humanos que describe la competencia y puede venir en varios idiomas.
- Definition: Es una estructura opcional que brinda una descripción más completa de la competencia u objetivo y suele estar asociada a modelos que describen como está estructurada la competencia o cómo se definió su estructura. Puede incluir elementos como habilidades, estándares e indicadores descritos a través de una serie de subelementos que lo conforman.

³⁰IMS-LIP, Acrónimo de *IMS Learner Information Package* . Disponible en <http://www.imsglobal.org/profiles/>

³¹RCDEO, Acrónimo de *IMS Reusable Definition of Competency or Educational Objective* . Disponible en <http://www.imsglobal.org/competencies/>

- Meta-metadata: Es un registro opcional que se sugiere que debe seguir las normas de la sección meta-metadata de IEEE-LOM.

A nivel teórico estos elementos permitirían realizar una búsqueda estructurada de recursos y catalogarlos apropiadamente de acuerdo a un modelo de competencias. Sin embargo para lograr esto, los metadatos de los OA deben estar dados de acuerdo a la especificación del modelo que se está analizando y como se presentó en la sección 5.3.3, los metadatos de los OA no están disponibles en todos los casos.

Por otro lado, la construcción de los identificadores está sujeta a un modelo de competencias definido para el sistema y para el momento de entrega de esta investigación, el TEC no posee un esquema de competencias oficial aprobado por la institución, de modo que cualquier propuesta que se pueda presentar sería realizada sin un modelo apropiado para el contexto.

5.5. Interfaz gráfica de usuario

El diseño de interfaces de usuario accesibles y eficaces ha tomado fuerza en los últimos años; de modo que ya no solo se busca generar recomendaciones; sino que el despliegue de la información sea útil para quien la reciba.

El TEC Digital cuenta con un equipo de profesionales encargados de velar por esta misión, y se encargan tanto de los proceso de diseño, como los de validación una vez que ha sido puesto a prueba una implementación.

La interfaz de usuario sugerida por el equipo de comunicación visual, busca conservar la armonía con los diseños de las interfaces previamente diseñadas para el MoGA y el DI, como se puede apreciar en la imagen 5.25; que corresponde a la pantalla de inicio para el usuario de la aplicación.

En esta pantalla se podrá apreciar un menú lateral donde el usuario podrá seleccionar la actividad dentro del DI para la cuál desear realizar la búsqueda de recursos de aprendizaje en el repositorio de su selección. Actualmente, sólo se está realizando la conexión

con el servicio publicado para Ariadne-Partners.

Recomendador de Objetos de Aprendizaje

Todos los repositorios ▼

Búsqueda ó tema

🔍

Criterios de búsqueda aplicados

Accesibilidad

▼ Estilos de aprendizaje

Visual

Auditivo

Kinestésico

▼ Formato del archivo

Audio

Video

Imágen

Texto

▼ Actividades

Act 1

Act 2

Act 3

Resultados de la búsqueda | Tema

Filtrar resultados por Seleccionar ▼

Objeto de aprendizaje	Fecha de publicación	Repositorio	Url	Calificación
<input type="checkbox"/> Objeto 1	dia-mes-año	Ariadne	Ubicación	★ ★ ★ ★ ★
<input type="checkbox"/> Objeto 2	dia-mes-año	Flor	Ubicación	★ ★ ★ ★ ★
<input type="checkbox"/> Objeto 3	dia-mes-año	Merlot	Ubicación	★ ★ ★ ★ ★
<input type="checkbox"/> Objeto 4	dia-mes-año	Ariadne	Ubicación	★ ★ ★ ★ ★
<input type="checkbox"/> Objeto 5	dia-mes-año	Ariadne	Ubicación	★ ★ ★ ★ ★
<input type="checkbox"/> Objeto 6	dia-mes-año	Flor	Ubicación	★ ★ ★ ★ ★
<input type="checkbox"/> Objeto 7	dia-mes-año	Merlot	Ubicación	★ ★ ★ ★ ★

Lista 📄

◀ 1 2 3 4 ... 8 ▶

Figura 5.25: Vista del menú para seleccionar actividades

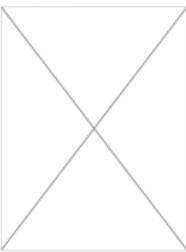
Posteriormente, se le ofrecerá al usuario final una serie de OA ordenados de acuerdo a la afinidad que tiene con la actividad seleccionada. Esta pantalla incluye elementos de navegación y está pensada para que las preferencias del usuario final se almacenen mediante un click sobre el OA en pantalla.

Los detalles de un OA estarán disponibles mediante un click sobre la lista que se despliega en la pantalla de inicio y será posible visualizar algunos de los metadatos relevantes disponibles a través de la descripción dada en LOM. En el detalle se incluirá el vínculo de descarga para el OA, de modo que si un usuario desea, puede ir directamente al URL origen del recurso y evaluarlo por sí mismo, más allá de la información disponible en la descripción.

Lista			Información
<input checked="" type="checkbox"/>	Objeto 1	día-mes-año	Ariadne
<input type="checkbox"/>	Objeto 2	día-mes-año	Flor
<input type="checkbox"/>	Objeto 3	día-mes-año	Merlot
<input type="checkbox"/>	Objeto 4	día-mes-año	Ariadne
<input type="checkbox"/>	Objeto 5	día-mes-año	Ariadne
<input type="checkbox"/>	Objeto 6	día-mes-año	Flor
<input type="checkbox"/>	Objeto 7	día-mes-año	Merlot

◀ 1 2 3 4 ... 8 ▶

Vista Previa



Nombre del OA
Nombre del ítem

Autor
Nombre Apellidos
Título

Repositorio
Nombre del repositorio al que pertenece

Descripción
Breve descripción del OA

Url
[Dirección de localización del OA](#)

Calificación
★ ★ ★ ★ ★

Figura 5.26: Vista previa de OA que se le ofrecerán al usuario final

6 Conclusiones

Este capítulo de conclusiones presenta los principales hallazgos identificados a lo largo del proceso de investigación y validación de la propuesta.

A modo general, la propuesta de un proceso de automatización para la recomendación de OA a partir de los documentos del DI y la teoría de estilos de aprendizaje es viable. Sin embargo, falta madurar los procesos de diseño y representación de los meta-elementos que son necesarios para llevar a cabo esta tarea.

Según las descripciones brindadas en (Adomavicius y Tuzhilin, 2005; Burke, 2007), nuestro sistema recomendador se ubica en la categoría de híbrido por el uso de características de los sistemas colaborativos (aplicación de la similitud de cosenos), demográficos (delimitación de vecindarios de acuerdo a las características del curso) y de los basados en contenidos (aplicación del TF-IDF) para generar la recomendación en cascada y mediante representaciones mixtas.

Al generar el modelo para los estilos de aprendizaje se encontró una serie de elementos sin métricas numéricas definidas dentro de la documentación analizada. Estos elementos describen características que son relevantes para la toma de decisiones con respecto a cuáles recursos seleccionar y su interpretación es un reto para los sistemas recomendadores.

Por otro lado, el esquema utilizado para modelar la gestión de atributos del MoGA actualmente está enfocado en la trazabilidad de las evidencias de aprendizaje del estudiante a través de la utilización de recursos. La información que puede aportar un esquema basado en evidencias sería mejor aprovechada en sistemas enfocados en modelos de usuario (o User Model) ya que la información con que se alimenta y su evolución depende de los progresos que tengan los estudiantes.

La aplicación de técnicas de clustering y distintas medidas de similitud sobre los metadatos permite realizar aproximaciones de recomendación sobre recursos disponibles en

los ROA; sin embargo, es necesario mejorar la calidad de la información contenida en los metadatos si se desea establecer un modelo efectivo de evaluación y catalogación.

La disponibilidad de textos sin calidad en cuanto a la redacción, contexto, nivel educativo esperado, entre otros; dificulta las tareas de preprocesamiento y validación de las hipótesis que se pueden plantear a partir de ellos.

La validación del proceso para generar recomendaciones a partir de información incompleta requiere de la incorporación de personal especializado en procesos de enseñanza y aprendizaje. De este modo, se podría aprovechar su conocimiento para validar los resultados y colaborar con el diseño de las adaptaciones del algoritmo de recomendación para contrarrestar los efectos de los datos faltantes.

La especificación de LOM-IEEE está pensada para compartir recursos educativos pero presenta importantes deficiencias si se desea modelar las características de los procesos educativos que se aplican en la actualidad. Estas características se podrían representar si se agrega una dimensión semántica a la especificación o bien, como se sugiere en (Baldiris, Santos, Boticario, y Fabregat, 2008), utilizando otra especificación como el IMS-RCDEO para representar los modelos de atributos y los objetivos de aprendizaje.

Hallazgos derivados del análisis del contexto

Después de analizar la documentación relacionada tanto con los recursos de aprendizaje como con los sistemas recomendadores y sus aplicaciones en los entornos educativos, se encontró una serie de aspectos que aún se encuentran sin resolver y que representan una oportunidad de mejora.

La aplicación de métricas numéricas a características psicológicas humanas está sujeta a la interpretación de quién desarrolla la métrica y a la capacidad de validación de los instrumentos utilizados al realizar la medición.

Los métodos de investigación cualitativa asociados a los procesos psicológicos requieren de modelos difusos para su utilización dentro de los paradigmas computacionales. Por esta razón es necesario difuminar los márgenes que establecen los puntos exactos de in-

flexión para una categoría con respecto a otra y esto solo se puede establecer mediante el trabajo con equipos multidisciplinarios que apoyen y fundamenten los cambios sobre los modelos construídos desde sus distintas perspectivas.

Con el proceso de preselección y reducción de los conjuntos de OA candidatos, se identificó la duplicidad de registros como una deficiencia de colección particular con la que se está trabajando. Estos OA duplicados generan registros adicionales que deben ser evaluados y se podrían mejorar los rendimientos de la aplicación y posiblemente del repositorio, si se realizaran mantenimientos periódicos para eliminar o agrupar estos OA que comparten ubicación y descripciones.

El tiempo de respuesta requerido para evaluar los metadatos de un OA aumenta en función del tamaño de las descripciones que contiene y la longitud de dicha descripción no es un indicador relevante para determinar la afinidad de un recurso con respecto a una actividad.

Al analizar la información asociada a la disponibilidad de los metadatos se evidenció que a pesar de los esfuerzos por mejorar su calidad, los elementos con mayor soporte desde el 2003 son los identificadores, la ubicación del recurso y el idioma. Esta información es suficiente para diferenciar los OA, pero no para realizar una recomendación de recursos para las actividades de un curso.

Se identificó que para realizar una recomendación de OA es necesario enfocarse en metadatos que describen aspectos académicos y de contenido, lo que implica utilizar metadatos con menor soporte en el repositorio pero que aportan un valor agregado a las descripciones de los recursos.

En cuanto al protocolo de consultas SQL, se encontraron diferencias importantes asociadas a la cardinalidad de una respuesta obtenida al especificar la búsqueda para un metadato específico y cuando no se especifica donde buscar. Esto toma importancia cuando se desea aplicar filtros a los OA, por ejemplo, el idioma que puede venir dado en múltiples formatos y estar en distintas secciones de la especificación en LOM.

El proceso de etiquetado manual en los contenidos de los metadatos produce errores en

las descripciones que van desde elementos ortográficos hasta la total ausencia de descriptores y este tipo de problemas debe ser enfrentado por cualquier recomendador que se desee diseñar para el trabajo con OA, del mismo modo que la duplicidad de registros dentro de los repositorios afecta el procesamiento y las recomendaciones generadas.

Hallazgos sobre la propuesta de solución

De las principales contribuciones en la construcción de la propuesta de solución a los problemas planteados se han obtenido en las siguientes conclusiones :

El proceso de reducción de los Ngramas mediante “stripping” no es suficiente para formar frases coherentes y relevantes dentro del idioma español, del mismo modo que la aplicación de algoritmos como el TF-IDF o similitudes de coseno de forma independiente no son indicadores reales de la relevancia práctica de los términos para un tema en particular.

Además, las ambigüedades semánticas y sintácticas del español dificultan la construcción de un analizador de expresiones que tome en consideración los tiempos verbales, sujetos, signos de puntuación y errores ortográficos, entre otras propiedades del idioma que requieren de la implementación de mecanismos de desambiguación estructural del idioma.

La aplicación de mecanismos de desambiguación estructural propios de los sistemas de procesamiento de lenguaje natural, representan por sí solos un proyecto de investigación y desarrollo complejo, pero su aplicación para identificar grupos nominales puede mejorar la calidad de los descriptores utilizados para construir el perfil de curso.

Por otro lado, cualquier herramienta que desee realizar un análisis de los contenidos de los metadatos debe contemplar al menos:

- Soporte para múltiples idiomas: O bien, definir estrategia para filtrar OA de acuerdo al idioma real que contienen.
- Soporte para estándares técnicos específicos: Puesto que las descripciones pue-

den venir codificadas en formatos específicos que introducen ruido a los textos, por ejemplo las codificaciones para URL en UTF8.

- Tolerancia a errores de redacción: Debido a que muchos de los contenidos de los metadatos son introducidos manualmente por usuarios del repositorio.
- Tolerancia a fallos en el etiquetado de los metadatos: Puesto que muchos de los OA no cumplen con la especificación de LOM completa, de modo que se pueden combinar etiquetas sobre el campo padre de la estructura; o bien, no existir del todo.

Hallazgos sobre disponibilidad de recursos

Las especificaciones de las interfases establecidas para comunicaciones entre los repositorios ofrecen modelos robustos, que permiten personalizar y mejorar la calidad de las respuestas esperadas ante una consulta. Estas consultas pueden ser realizadas en distintos modelos de sintaxis; sin embargo su utilidad real está limitada por la implementación de los métodos disponibles para el repositorio que se está referenciando.

A partir de las pruebas realizadas con el SQL para recuperar metadatos desde distintos ROA, se encontró que los métodos relacionados con la cardinalidad de las respuestas no están disponibles en todos los repositorios. Esto representa una dificultad técnica al realizar búsquedas de recursos fuera de las interfases de navegación web que ofrecen los ROA y complica la tarea de recuperar información.

Se encontró que el acceso mediante servicios web produce una latencia en el tiempo de respuesta que podría mejorarse si los algoritmos estuvieran directamente integrados al repositorio o bien, si las implementaciones del SQL y OAI-PMH estuvieran completas para reducir la cantidad de consultas necesarias para obtener una respuesta.

En la descripción general de IEEE-LOM no se incluyen rubros para identificar el mecanismo de encapsulación de los recursos digitales; de modo que las lecciones con objetivos de aprendizaje definidos y otros elementos de carácter educativo quedan reducidos a los detalles que aporte quién redacte los metadatos del OA. Además, dentro de los OA recu-

perados para las pruebas no se encontró ninguno que incluya descripciones asociadas a los estándares técnicos de su encapsulación.

Ante la cantidad de metadatos obtenidos mediante el OAI-PMH y SQL se hizo evidente que el volumen de información recuperada durante la cosecha excede la capacidad del equipo en el que se ejecuta el prototipo. Además, se determinó que se pueden conseguir resultados equivalentes mediante la aplicación de ambos mecanismos, ya que el OAI-PMH se puede configurar con comandos de cosecha selectiva para comportarse como lo haría una consulta con SQL; con la diferencia de no ser posible determinar la cardinalidad de la respuesta a obtener.

Al ejecutar el prototipo con el DI de pruebas se identificaron 1108 posibles asociaciones hacia 701 recursos distintos que podrían ser útiles de acuerdo a los parámetros de la búsqueda definidos. Al evaluar estos recursos, se identificaron 335 elementos correspondientes a lecturas (archivos en formato .doc, .pdf y .html), 89 fueron presentaciones (archivos en formato .ppt y .pptx) y sólo 22 recursos de imágenes (archivos en formato .jpg y .tiff), de modo que al menos para la muestra obtenida, se ha mantenido vigente el patrón de disponibilidad de recursos desde el 2003, donde la mayoría de los recursos compartidos corresponden a elementos textuales y presentaciones.

6.1. Resumen del cumplimiento de los objetivos

- Al realizar el estudio del estado del arte de los sistemas recomendadores en el ámbito educativo, se encontró que su aplicación es relativamente reciente y que existen muchas oportunidades para aprovechar sus características en distintos procesos de las plataformas de gestión del aprendizaje.
- La implementación de una herramienta que analice los descriptores de los OA es viable y permite trabajar con los contenidos disponibles; sin embargo, la calidad de los textos contenidos es un factor crítico para cualquier tarea que se desee realizar con ellos.

- El diseño instruccional aporta información relevante que permite aproximar una recomendación de recursos; sin embargo, su utilidad está limitada por la calidad de las descripciones de la lección brindadas por el docente. Las descripciones deben ser claras y con suficiente contenido para limitar los contextos en los que se buscarán los recursos, además, utilizar descripciones muy densas no garantiza que se encontrarán recursos apropiados.
- Encontrar los recursos que ayuden a fomentar las competencias de aprendizaje a partir de una descripción de contenidos no es viable en este momento. Existen elementos que aún no han sido definidos para realizar el mapeo con respecto a LOM y los ROA. Además, actualmente el TEC no cuenta con un modelo de atributos oficial para trabajarlos y los mecanismos que han implementados otras entidades se basan en catálogos, más que en búsquedas de contenidos dinámicos.

6.2. Resumen de la contribución al estado del arte

A continuación se citan de manera puntual los aportes realizados a partir de este trabajo de investigación, partiendo de la hipótesis inicial, en la cuál se propone una solución para el problema de filtrar y seleccionar un banco de recursos de forma automática a partir de los documentos de diseño instruccional de un curso, los estilos de aprendizaje y los modelos de competencias con la aplicación de técnicas de minería de datos.

- Se demostró que la construcción de perfiles para un sistema recomendador a partir de contenidos dados en prosa implica la integración de procesos asociados al procesamiento de lenguaje natural que requieren de procesos complejos y costosos a nivel computacional, pero permiten acotar las frases y transformarlas en estructuras manipulables dentro de un sistema de recomendación.
- Se demostró que es posible generar recomendaciones para construir un banco de recursos para un curso a partir de los documentos de diseño instruccional de un

curso y los estilos de aprendizaje mediante la aplicación de técnicas de minería de datos. En este caso se validó mediante la utilización de medidas de similitud como lo son la similitud coseno, el TF-IDF y el análisis de clústers para reconocer las relaciones entre variables.

- Se ofreció evidencia puntual de la necesidad de integrar expertos en el área de enseñanza-aprendizaje durante el proceso de diseño de los estándares de intercambio de recursos digitales educativos, debido a que el modelo actual no toma en consideración los requerimientos de los modelos educativos vigentes.
- Se ofreció una aproximación para la corrección ortográfica en español sin integrar un NLP para reducir las dimensiones de los vectores que representan los perfiles de un curso y sus actividades, de propósito general y sin requerir de un diccionario de dominio específico.
- Se identificó una serie de mejoras importantes que se pueden aplicar al modelo con el fin de refinar los procesos de recomendación de un banco de recursos que no está sujeto a un dominio particular.

6.3. Trabajos futuros

A partir de la investigación realizada se identificaron nuevas líneas de trabajo que ofrecen nichos sin explorar, o bien, que ofrecen oportunidades de mejorar en las diversas áreas de conocimiento involucradas en este proceso.

En cuanto a las mejoras a la propuesta, se identificaron los siguientes elementos:

- **Proceso de extracción de descriptores:** Mediante la integración de herramientas especializadas en distintas fases del procesamiento de lenguaje natural como mecanismo de mejora para el prototipo. En esta tesis no se consideró el procesamiento

de lenguaje natural dentro de los alcances de la aplicación, sin embargo, se identificaron tareas específicas que deben ser enlazadas en este tipo de sistemas para facilitar las tareas de recomendación.

Por ejemplo, incorporar un diccionario de corrección ortográfica y un diccionario del tipo thesaurus, con el fin de mejorar la calidad de los textos actualizados de forma automática; o bien, identificar el idioma de los textos retornados para los casos donde el etiquetado de los metadatos es deficiente.

Durante la fase de investigación se encontraron servicios web gratuitos con limitantes sobre la cantidad de consultas por hora, o bien servicios de pago para detectar el idioma de las cadenas de caracteres. También se encontró una librería “Open-source” en java para esta misma funcionalidad, pero trabaja a partir de archivos de texto; por lo que su eventual adaptación hacia cadenas de caracteres podría ser aprovechada por el recomendador y mejorar los filtros sobre los OA útiles.

La incorporación de un POSTagger para el idioma español como parte del preprocesamiento mejoraría de forma significativa la calidad y el tiempo de respuesta de las búsquedas, ya que se podrían extraer únicamente las entidades formales dentro de los criterios de búsqueda y se daría un paso hacia la construcción de un modelo semántico a partir de texto sencillos.

Otra posible área de interés que se abre con esta investigación es la aplicación de mecanismos de desambiguación estructural en el análisis de contenidos, tanto de las descripciones de los cursos como en los contenidos de los OA.

- **Diseño de algoritmos aplicados:** En el área de diseño de algoritmos siempre hay oportunidades de mejora. En esta línea en particular queda mucho por hacer para mejorar la aproximación de los recursos sugeridos y los tiempos de respuesta.

La aplicación del modelo sugerido en el prototipo demostró que es posible generar las recomendaciones para el banco de datos; pero es necesario mejorar la definición y la precisión de las funciones utilizadas para generar la utilidad esperada de un recurso.

Además, es necesario generar un estudio formal de las diferencias al aplicar la propuesta comparada con otros algoritmos de recomendación que se puedan ajustar a modelos vectoriales.

- **Análisis del esquema de etiquetado automático de LACLO:** Durante la revisión de materiales se encontró una mención a con respecto a un modelo de etiquetado automatizado en el repositorio LACLO, pero no fue posible encontrar referencias puntuales sobre su implementación.

El análisis del modelo utilizado para generar metadatos de forma automática puede contribuir a mejorar el proceso de desambiguación de metadatos y con esto, a mejorar la precisión de las recomendaciones.

- **Adaptación para soporte a ontologías:** Durante las fases de revisión de materiales y trabajos de distintas fuentes se encontraron múltiples modelos basados en ontologías que permitirían ampliar las capacidades de expansión y adaptación del sistema.
- **Redacción de artículo para difusión de resultados:** Se escribirá al menos una publicación para difundir el conocimiento generado en esta investigación.

6.4. Futuras investigaciones

Aparte de las mejoras que se pueden aplicar al modelo propuesto, se identificaron otras áreas de interés para investigaciones futuras que podrían aportar conocimientos nuevos o oportunidades de mejoras en áreas relacionadas.

Estructuración de procesos de mantenimiento de los ROA

Para los alcances de esta investigación no se analizaron los procesos internos de los repositorios, pero sí fue posible identificar duplicados y errores de etiquetado dentro de los recursos retornados. Además, en las fases iniciales de la investigación, las pruebas

con ARIADNE-MEMBERS fueron canceladas debido a labores de mantenimiento sobre esa colección. Específicamente, las modificaciones sobre la estructura de metadatos de LOM en los recursos a retornar.

A nivel de diseño de procesos, se identificaron deficiencias en cuanto a la calidad de los metadatos que describen los OA y su disponibilidad, por lo que es necesario establecer un protocolo de mantenimientos o rediseño de los procesos de etiquetado que ayude a reducir las duplicidades y los agujeros de meta-información.

Las mejoras en la calidad de los metadatos permitirían un intercambio más fluido de información y simplificarían los procesos de identificación y clasificación de recursos relevantes que sería aprovechado tanto por usuarios como por quienes están a cargo de los ROA.

Estudios sobre la aplicación y utilidad real de los protocolos de intercambio de información

A pesar de existir la definición de protocolos para el intercambio de información entre los repositorios, no fue posible utilizar todas las funciones en los ROA de prueba y no se encontró información actualizada sobre la implementación de las funciones específicas entre los miembros del consorcio.

Las tablas de disponibilidad de funciones se limitan a indicar si un miembro ofrece soporte para OAI-PMH bajo cuál versión, y si ofrece SQI y con cuál nivel. Pero no ofrecen información sobre funciones específicas como cambios en la sintaxis a utilizar, o bien, si ofrecen soporte a cambios en la cardinalidad de la respuesta esperada.

Un estudio formal de la disponibilidad de funciones contribuiría a actualizar el estado del arte del proceso de intercambio de información entre repositorios.

Establecer un modelo para alinear las competencias con respecto a los OA

En el caso del modelo por competencias, los perfiles de egreso se plantean integrando conocimientos, habilidades y actitudes que deben ser adquiridas durante el proceso educativo.

Estas habilidades no han sido modeladas dentro de LOM, por lo que aún queda abierto a propuestas de representación y está pendiente un modelo para representarlo o inferirlo de los contenidos de los OA.

A Anexos

Requerimientos de instalación

Para consumir recursos de terceros, siempre es necesario identificar los requerimientos mínimos para realizar la instalación y configuración de los recursos. En nuestro caso, estaremos consumiendo funciones de dos fuentes especializadas: los servicios provistos por el ROA y una serie de librerías de PostgreSQL encargada de facilitar funciones para XML y minería de datos.

Inicialmente se consideró utilizar el API de recomendación provisto por el Lenskit, pero el aporte de la integración no era significativo en esta fase del prototipo. Para su consumo se necesita contar al menos con los siguientes elementos:

- Java JDK 1.6.
- Maven 2.2 o superior.¹

En cuanto a los requerimientos especiales sobre PostgreSQL, solo se necesitará agregar las extensiones disponibles en el paquete contrib. En las versiones posteriores a la 9.1 y 9.2, este paquete viene incluido dentro de la instalación básica; sin embargo, en los servidores del TEC Digital, se encuentra instalada la versión 8.4.

Las extensiones previamente mencionadas ofrecen funciones para manejo de XML mediante Xpath 1.0 y librerías para procesamiento de textos con algoritmos propios de los procesos de minería de datos.

También es necesario contar con una librería que permita la integración de código en Java y Tcl; debido a que el toolkit de recomendación está escrito en Java y la plataforma del TEC Digital está contruida sobre dotLRN. El código escrito para este LMS está escrito en

¹Apache Maven, disponible en <http://maven.apache.org/>. Es una herramienta utilizada para construir y administrar proyectos desarrollados en Java.

su mayoría sobre Tcl y se publica sobre aolserver.

Algunos ejemplos de las librerías que facilitan este intercambio de código y se pueden utilizar son las siguientes:

- Jacl: es un intérprete de Tcl 8.x escrito en Java para ejecutar código de Java sobre aplicaciones en Tcl.
- Swank²: es un toolkit gráfico para Jacl que permite usar código en java sobre los widgets de Tk.
- Jtcl³: es un intérprete de Tcl escrito en Java que implementa la mayoría de sus funciones y es compatible con Tcl 8.4.
- Tcl Blend⁴: es una extensión de Tcl que permite la comunicación entre el interprete de Java y el de Tcl. Se usa para ejecutar código escrito en Java sobre un proceso de Tcl.

Finalmente, las dependencias de la información del TEC Digital se centran en la existencia de datos para el curso en las aplicaciones de DI y MoGa; mientras que con respecto al DAR se requiere la disponibilidad de los servicios web institucionales asociados a las equivalencias de cursos.

² Swank, disponible en <https://swank.kenai.com/>

³ Jtcl, disponible en <https://jtcl.kenai.com/>

⁴ Tcl Blend, disponible en <http://tcljava.sourceforge.net/docs/website/getstart.html>

Resumen de OA disponibles por dialecto de español en Ariadne Partners

OA por dialecto de español disponibles en Ariadne-Partners				
Idioma		Cantidad de OA encontrados		
Código	País	General	Metametadatos	Educational
es-AR	Argentina	0	0	0
es-BO	Bolivia	0	0	0
es-CL	Chile	0	0	0
es-CO	Colombia	0	0	0
es-CR	Costa Rica	0	0	0
es-DO	República Dominicana	0	0	0
es-EC	Ecuador	106710	0	0
es-ES	Castilla-España	0	0	0
es-GT	Guatemala	0	0	0
es-HN	Honduras	0	0	0
es-MX	México	0	0	0
es-NI	Nicaragua	0	0	0
es-PA	Panamá	0	0	0
es-PE	Perú	0	0	0
es-PR	Puerto Rico	0	0	0
es-PY	Paraguay	0	0	0
es-SV	El Salvador	0	0	0
es-UY	Uruguay	0	0	0
es-VE	Venezuela	0	0	0
es	Español genérico	3645	15	716
spa	Español genérico	13	13	0

Tabla A.1: Resumen de la cardinalidad de conjuntos recuperados al consultar por dialectos de Español

Tabla comparativa de técnicas de recomendación

Técnica	Ventajas	Desventajas
Basados en utilidad	Puede evaluar características ajenas a los ítemes en la ecuación (por ejemplo: tiempos de entrega) (Burke, 2007)	La función de utilidad debe ser capaz de revisar todas las características de los objetos que estén en evaluación (Burke, 2007) Poco útil para dominios con muchas características (Burke, 2007).
Basados en conocimientos	Se puede rastrear el origen de las recomendaciones (Ricci y cols., 2011) . Las soluciones a las inconsistencias en los requerimientos se pueden solucionar automáticamente (Ricci y cols., 2011) .	Las recomendaciones son solo para un dominio específico (Ricci y cols., 2011) La definición del catálogo de conocimientos, el conocimiento funcional y el conocimiento del usuario es complicado y debe ajustarse a las necesidades de los usuarios.
Basados en contenidos	Utiliza la información de otros usuarios con perfiles o preferencias similares para determinar la mejor recomendación (Manouselis y Costopoulou, 2007)	No posee capacidad de aprendizaje, por lo que requiere de conocimientos para generar la recomendación (Manouselis y Costopoulou, 2007) .
Demográficos	Permite realizar personalizaciones por área, edad, idioma, etc (Ricci y cols., 2011). No requiere de análisis contextual (Manouselis y cols., 2011). No presenta el problema de "cold-start" (Manouselis y cols., 2011).	Existe poca literatura de su aplicación directa a sistemas recomendadores (Ricci y cols., 2011) Solo recomienda los ítemes populares
Colaborativos	Son mejores para universos con pocos ítemes y usuarios dinámicos (Burke, 2007). La calidad mejora con el tiempo (Burke, 2007) Independientes del dominio (Manouselis y cols., 2011) Puede recomendarle al usuario ítemes que aun no ha visto (Manouselis y cols., 2011) Es el modelo de implementación más sencillo (Ricci y cols., 2011)	Varios usuarios pueden evaluar el mismo ítem muchas veces (Manouselis y cols., 2011). La calidad depende de la información histórica. Depende de las evaluaciones cruzadas de los usuarios. Problemas al inicio para manipular nuevos usuarios y nuevos ítemes (Manouselis y cols., 2011) No puede recomendar algo que no ha sido evaluado (Ricci y cols., 2011). Insensible a cambios en preferencias (Ricci y cols., 2011).

Tabla A.2: Ventajas y Desventajas de las técnicas de recomendación

Entorno de desarrollo para el recomendador

Java JDK7

Apache Maven6 Lenskit

IDE para Java

Crear un proyecto en Maven

Instalar: `apt-get install maven`

Una vez concluida la instalación se puede probar la instalación de maven mediante el siguiente comando:

```
mvn --version
```

Al ejecutar este comando, el sistema operativo debe retornar la información asociada a la versión de java y el equipo sobre el que se está ejecutando la aplicación:

```
root@tec-digital-laptop:/usr/share/dotlrn/packages/tda-apc/www/ajax# mvn --version
Apache Maven 2.2.1 (rdebian-1)
Java version: 1.6.0_27
Java home: /usr/lib/jvm/java-6-openjdk/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux" version: "2.6.32-21-generic" arch: "i386" Family: "unix"
root@tec-digital-laptop:/usr/share/dotlrn/packages/tda-apc/www/ajax# █
```

Posteriormente, se procede a crear el proyecto. Esto puede realizarse mediante un entorno de desarrollo para Java, o bien desde línea de comando con el siguiente código:

```
mvn archetype:generate \
  -DarchetypeGroupId=org.grouplens.lenskit \
  -DarchetypeArtifactId=lenskit-archetype-simple-analysis \
  -DarchetypeVersion=2.0 \
  -DinteractiveMode=no \
  -DgroupId=tecDigital \
  -DartifactId=arhoa \
  -Dversion=0.1
```

Lo que nos creará un proyecto nuevo sobre la carpeta en la que nos encontremos utilizando la plantilla para análisis simple de lenskit, como se muestra a continuación:

```

root@tec-digital-laptop:/usr/share/dotlrn/packages/tda-apc/www/ajax# mvn archetype:generate
-DarchetypeGroupId=org.grouplens.lenskit -DarchetypeArtifactId=lenskit-archetype-simple-analysis
-DarchetypeVersion=2.0 -DinteractiveMode=no -DgroupId=tecDigital -DartifactId=arhoa_
-Dversion=0.1
[INFO] Scanning for projects...
[INFO] Searching repository for plugin with prefix: 'archetype'.
[INFO] -----
[INFO] Building Maven Default Project
[INFO]   task-segment: [archetype:generate] (aggregator-style)
[INFO] -----
[INFO] Preparing archetype:generate
[INFO] No goals needed for project - skipping
[INFO] [archetype:generate {execution: default-cli}]
[INFO] Generating project in Batch mode
[INFO] Archetype repository missing. Using the one from [org.grouplens.lenskit:lenskit-archetype-simple-analysis:2.0.1] found in catalog remote
[INFO] -----
[INFO] Using following parameters for creating project from Archetype: lenskit-archetype-simple-analysis:2.0
[INFO] -----
[INFO] Parameter: groupId, Value: tecDigital
[INFO] Parameter: artifactId, Value: arhoa_
[INFO] Parameter: version, Value: 0.1
[INFO] Parameter: package, Value: tecDigital
[INFO] Parameter: packageInPathFormat, Value: tecDigital
[INFO] Parameter: package, Value: tecDigital
[INFO] Parameter: version, Value: 0.1
[INFO] Parameter: groupId, Value: tecDigital
[INFO] Parameter: artifactId, Value: arhoa_
[INFO] project created from Archetype in dir: /usr/share/dotlrn/packages/tda-apc/www/ajax/arhoa_
[INFO] -----
[INFO] BUILD SUCCESSFUL
[INFO] -----
[INFO] Total time: 41 seconds
[INFO] Finished at: Mon Sep 30 15:27:05 CST 2013
[INFO] Final Memory: 19M/47M
[INFO] -----
root@tec-digital-laptop:/usr/share/dotlrn/packages/tda-apc/www/ajax# █

```

Dentro de la carpeta de proyecto, se crearán una serie de archivos y carpetas. La carpeta `src/main/java` contiene el código fuente, la carpeta `src/test/java` contiene el código de prueba y el archivo `pom.xml` file contiene el POM⁵ del proyecto. La creación del proyecto puede demorarse debido a que maven descarga archivos desde repositorios hacia el equipo donde se está creando el nuevo proyecto ⁶.

⁵POM, acrónimo de Project Object Model

⁶<http://maven.apache.org/guides/getting-started/maven-in-five-minutes.html>

```

root@tec-digital-laptop:/home/tec-digital# tree arhoa/
arhoa/
├── chart.R
├── eval-simple.groovy
├── pom.xml
├── src
│   ├── main
│   │   └── java
│   │       ├── tecDigital
│   │       └── App.java
│   └── test
│       └── java
│           ├── tecDigital
│           └── AppTest.java

```

7 directories, 5 files

Para usar lenskit con Maven, existen dos arquetipos⁷ (plantillas de proyecto) que permiten configurar proyectos nuevos:

- lenskit-archetype-simple-analysis : configura el proyecto al mínimo y coloca todos los archivos sobre la misma carpeta.
- lenskit-archetype-fancy-analysis: para proyectos más sofisticados que utilizan las estructuras de carpetas de src y target de Maven apropiadamente y permiten escalar hacia evaluaciones múltiples y análisis de scripts.

Además, el archivo pom.xml es el archivo de configuración para los proyectos en Maven y contiene la mayoría de la información necesaria para construir el proyecto.

Para construir el proyecto, se debe posicionar sobre la carpeta del proyecto. En nuestro caso, arhoa, y ejecutar el siguiente comando en la terminal:

```
mvn package
```

Si el comando no encuentra errores, inicia la descarga de los paquetes asociados mediante el arquetipo y sus dependencias (entre ellas se incluye el groovy).

```

root@tec-digital-laptop:/home/tec-digital# cd arhoa/
root@tec-digital-laptop:/home/tec-digital/arhoa# mvn package
[INFO] Scanning for projects...
Downloading: http://repo1.maven.org/maven2/org/grouplens/lenskit/lenskit-eval-maven-plugin/2.0/lenskit-eval-maven-plugin-2.0.pom
4K downloaded (lenskit-eval-maven-plugin-2.0.pom)
Downloading: http://repo1.maven.org/maven2/org/grouplens/lenskit/lenskit/2.0/lenskit-2.0.pom
21K downloaded (lenskit-2.0.pom)
Downloading: http://repo1.maven.org/maven2/org/grouplens/grouplens-oss-parent/2/grouplens-oss-parent-2.pom

```

⁷<https://bitbucket.org/grouplens/lenskit/wiki/GettingStarted>

Una vez finalizadas las descargas, se le retorna al usuario un mensaje con el estado y el nuevo tamaño del proyecto:

```
[INFO] [jar:jar {execution: default-jar}]
[INFO] Building jar: /home/tec-digital/arhoa/target/arhoa-0.1.jar
[INFO] -----
[INFO] BUILD SUCCESSFUL
[INFO] -----
[INFO] Total time: 8 minutes 28 seconds
[INFO] Finished at: Mon Sep 30 13:52:32 CST 2013
[INFO] Final Memory: 25M/60M
[INFO] -----
```

Y podemos visualizar la nueva estructura del proyecto con los archivos actualizados por maven:

```
root@tec-digital-laptop:/home/tec-digital/arhoa# tree
.
├── chart.R
├── eval-simple.groovy
├── pom.xml
├── src
│   ├── main
│   │   └── java
│   │       ├── tecDigital
│   │       └── App.java
│   └── test
│       └── java
│           ├── tecDigital
│           └── AppTest.java
└── target
    ├── arhoa-0.1.jar
    ├── classes
    │   └── tecDigital
    │       └── App.class
    ├── maven-archiver
    │   └── pom.properties
    ├── surefire-reports
    │   ├── tecDigital.AppTest.txt
    │   └── TEST-tecDigital.AppTest.xml
    ├── test-classes
    │   └── tecDigital
    │       └── AppTest.class
    └──
```

14 directories, 11 files

Finalmente, para probar que el .jar que genera la aplicación es válido, se realiza la prueba mediante el comando:

```
java -cp target/arhoa-0.1.jar tecDigital.App
```

El cuál, muestra como resultado:

```
root@tec-digital-laptop:/home/tec-digital/arhoa# java -cp target/arhoa-0.1.jar tecDigital.App
Hello World!
root@tec-digital-laptop:/home/tec-digital/arhoa# █
```

Configuración de Jacl y Tcl Blend

Jacl y Tcl Blend ofrecen un paquete en Tcl llamado "java", el cual permite usar comandos para que objetos creado en Java interactúen desde el Tcl ⁸.

Instalación de Jacl

Asumiendo que la carpeta de instalación será /opt/jacl y que el openjdk se encuentra sobre /usr/lib/jvm:

```
cd /tmp/jacl
tar -xzvf jacl1.4.1.tar.gz
mkdir build
cd build
../jacl1.4.1/configure --prefix=/opt/jacl \
    --with-jdk=/usr/lib/jvm/java-6-openjdk/
```

Después de ejecutar estos comandos, iniciará una descompresión hasta que creen los archivos de configuración como se muestra en la siguiente figura:

```
root@tec-digital-laptop:/opt/jacl/build# ../jacl1.4.1/configure --prefix=/opt/jacl/ --with-jdk=/usr/lib/jvm/java-6-openj
/
srcdir is /opt/jacl/jacl1.4.1
checking for zip or jar files to include on CLASSPATH...
checking to see if the java compiler works... yes
Java found in /usr/lib/jvm/java-6-openjdk
checking type of jvm... jdk
checking java API version... 1.5
checking for java... /usr/lib/jvm/java-6-openjdk/bin/java
checking for java.g... no
checking for javah... /usr/lib/jvm/java-6-openjdk/bin/javah
checking for jar... /usr/lib/jvm/java-6-openjdk/bin/jar
checking for jdb... /usr/lib/jvm/java-6-openjdk/bin/jdb
checking for ddd... no
checking for gdb... /usr/bin/gdb
checking the Java compiler... the Java compiler works
checking the the Java runtime... the Java runtime works
checking the Java archive program... the jar program creates uncompressed archives
the jar program extracts uncompressed archives
the jar program creates compressed archives
the jar program extracts compressed archives
checking that the jar program did not corrupt the test file... it did not
configure: creating ./config.status
config.status: creating Makefile
config.status: creating tcljavaConfig.sh
config.status: creating jaclsh
config.status: creating tjc
root@tec-digital-laptop:/opt/jacl/build# █
```

Posteriormente se procede a ejecutar contruir e instalar el jacl con los siguientes comandos:

⁸<http://tcljava.sourceforge.net/docs/website/getstart.html>

```
make
make install
```

Con lo que finalizará el proceso de instalación del jacl:

```
root@tec-digital-laptop:/opt/jacl/build# make install
#
# Installing tcljava
#
Making directory /opt/jacl/lib/tcljava1.4.1
Making directory /opt/jacl/bin
Installing tcljava.jar in /opt/jacl/lib/tcljava1.4.1
Installing tcljavasrc.jar in /opt/jacl/lib/tcljava1.4.1
#
# Installing jacl
#
Installing jacl.jar in /opt/jacl/lib/tcljava1.4.1
Installing jaclsrc.jar in /opt/jacl/lib/tcljava1.4.1
Installing jaclsh in /opt/jacl/bin
Testing installed program
Installed program is working correctly
#
# Installing itcl
#
Installing itcl.jar in /opt/jacl/lib/tcljava1.4.1
Installing itclsrc.jar in /opt/jacl/lib/tcljava1.4.1
#
# Installing tjc
#
Installing tjc.jar in /opt/jacl/lib/tcljava1.4.1
Installing tjcsrc.jar in /opt/jacl/lib/tcljava1.4.1
Installing tjc in /opt/jacl/bin
Creating jdk.cfg in /opt/jacl/bin
#
# Installing janino
#
Installing janino.jar in /opt/jacl/lib/tcljava1.4.1
Installing janinosrc.jar in /opt/jacl/lib/tcljava1.4.1
root@tec-digital-laptop:/opt/jacl/build# █
```

Para verificar los módulos instalados se puede usar el siguiente comando:

```
make test
```

Y durante su ejecución se indican aquellos elementos que han reportado errores. El test de Jacl encontró múltiples errores en el manejo de buffers e hilos al ser compilado; así como elementos relacionados con la codificación de los comandos. La última actualización de este proyecto fue en el 2009; sin embargo fue posible ejecutar funciones escritas en tcl sobre él, como se puede apreciar en la siguiente figura.

```

root@tec-digital-laptop:/opt/jacl/bin# ./jaclsh
% proc compare {howmany value} {
  for {set i 1} {$i < $showmany} {incr i} {
    if {$i == $value} {
      puts $i
    }
  }
}

set bench 1000

set bench2 [expr $bench - 1]

puts "comparing an int against $bench others..."

compare $bench $bench2
% % 1000
% % 999
% % comparing an int against 1000 others...
% % 999
% █

```

Al utilizar esta herramienta para ejecutar comandos desde un archivo, se obtiene el mismo resultado:

```

root@tec-digital-laptop:/opt/jacl/bin# ./jaclsh test.tcl
comparing an int against 1000 others...
999

```

A partir del 26 de noviembre del 2010, se lanzó Jtcl como parte del proceso de modernización del Jacl⁹ que fue iniciado por *Google Summer of Code* en el 2009; entregando su versión 2.6.0 el 4 de Agosto del 2013¹⁰. Esta versión solo incluye cambios sobre el Jacl y los usuarios de TclBlend pueden seguir utilizando las extensiones bajo este nombre.

Para instalar el Jtcl, simplemente se descarga la versión a utilizar y se descomprime sobre la carpeta seleccionada para la instalación.

```

root@tec-digital-laptop:/opt/jtcl# unzip jtcl-2.6.0-bin.zip
Archive:  jtcl-2.6.0-bin.zip
  creating:  jtcl-2.6.0/
  inflating:  jtcl-2.6.0/license.terms
  inflating:  jtcl-2.6.0/license.ucb
  inflating:  jtcl-2.6.0/license.tcllib
  inflating:  jtcl-2.6.0/license.janino
  inflating:  jtcl-2.6.0/license.itcl
  inflating:  jtcl-2.6.0/license.amd
  inflating:  jtcl-2.6.0/jtcl-2.6.0.jar
  inflating:  jtcl-2.6.0/README
  inflating:  jtcl-2.6.0/jtcl.bat
  inflating:  jtcl-2.6.0/tjc.bat
  inflating:  jtcl-2.6.0/jtcl
  inflating:  jtcl-2.6.0/tjc
  inflating:  jtcl-2.6.0/echopath
  inflating:  jtcl-2.6.0/paraffin.tcl
root@tec-digital-laptop:/opt/jtcl# █

```

⁹<http://wiki.tcl.tk/27453>

¹⁰<https://jtcl.kenai.com/news.html>

Posteriormente, se ingresa a la carpeta jtcl-(versión descargada) y se ejecuta el archivo jtcl con el nombre del archivo tcl o bien, utilizando el modo interactivo (al igual que con Jacl):

```

root@tec-digital-laptop:/opt/jtcl# jtcl
% proc compare {howmany value} {
  for {set i 1} {$i < $showmany} {incr i} {
    if {$i == $value} {
      puts $i
    }
  }
}

set bench 1000

set bench2 [expr $bench - 1]

puts "comparing an int against $bench others..."

compare $bench $bench2
%% 1000
%% 999
%% comparing an int against 1000 others...
%% 999
%

```

En caso de desear ejecutar un archivo en particular, se utiliza la siguiente sintaxis:

```
jtcl filename arg arg ...
```

Y al ejecutar el archivo de pruebas utilizado con el Jacl, nos da el siguiente resultado:

```

root@tec-digital-laptop:/opt/jtcl# jtcl ../jacl/bin/test.tcl
comparing an int against 1000 others...
999
root@tec-digital-laptop:/opt/jtcl#

```

Instalación de TclBlend

Tcl Blend es un paquete en TCL que provee acceso a clases en java desde el tcl y su última versión estable fue la 1.4.1, entregada en el Abril del 2008.

Para realizar la instalación del tclBlend, es necesario contar con el tcl8.4 y la librería de threads habilitada. En un sistema con Ubuntu estas librerías se encuentra en /usr/share/tcltk/tcl8.4 y en /usr/lib/tcltk/thread2.6.5/, una vez identificadas estas rutas podemos proceder a compilar el tclBlend. Asumiremos que la carpeta de instalación será /opt/tclBlend y que se utilizará /usr/lib/jvm/java-6-openjdk/ como jdk. Entonces, ejecutamos:

```

cd build_tclblend/
../tclBlend1.4.1/configure --prefix=/opt/tclblend \

```

```

--with-tcl=/usr/share/tcltk/tcl8.4 \
--with-thread=/usr/lib/tcltk/thread2.6.5 \
--with-jdk=/usr/lib/jvm/java-6-openjdk/

make
make install

```

Para ejecutarlo deben estar instaladas las librerías de desarrollo del tcl (tcl8.4-dev), de otro modo el archivo de configuración no reconocerá el tcl instaladas. Además, es necesario modificar el script de instalación para adecuar la ruta hacia el tclsh.

```

if test "$ac_cv_tcl_win32" = "yes"; then
  TCLSH_LOC=$TCL_BIN_DIR/tclsh${TCL_VERSION_NO_DOTS}${TCL_DBGX}
else
# jacsolis> modificacion al script. Originalmente busca el tclsh en la misma carpeta que el tcl8.4, sin
#   embargo, en Ubuntu se encuentran en rutas distintas.
#   TCLSH_LOC=$TCL_BIN_DIR/tclsh
#   TCLSH_LOC=/usr/bin/tclsh
fi
if test ! -x "$TCLSH_LOC"; then
  { { echo "$as_me:$LINENO: error: Tcl was configured in $TCL_BIN_DIR, but it has not been built, please $
echo "$as_me: error: Tcl was configured in $TCL_BIN_DIR, but it has not been built, please build it and ru$
  { (exit 1); exit 1; }; }
fi

```

Según se indica el instalador, además se pueden tener problemas al compilar con gcc 4.0.1 y 4.0.2; por lo que el sistema automáticamente aborta la instalación con las versiones superiores a 4 para evitar la incompatibilidad con el optimizador de código.

Al igual que con el Jacl, con tclBlend se detectaron errores al ejecutar los test de funcionamiento:

```

tclblend/javaIdle.test
#
# A fatal error has been detected by the Java Runtime Environment:
#
# SIGSEGV (0xb) at pc=0x400a3d6b, pid=32760, tid=1076386736
#
# JRE version: 6.0_27-b27
# Java VM: OpenJDK Client VM (20.0-b12 mixed mode, sharing linux-x86 )
# Derivative: IcedTea6 1.12.6
# Distribution: Ubuntu 10.04.1 LTS, package 6b27-1.12.6-1ubuntu0.10.04.2
# Problematic frame:
# C [libtcl8.4.so.0+0x73d6b] Tcl_FindCommand+0x2b
#
# An error report file with more information is saved as:
# /opt/tclBlend/build_tclblend/hs_err_pid32760.log
#
# If you would like to submit a bug report, please include
# instructions how to reproduce the bug and visit:
# https://bugs.launchpad.net/ubuntu/+source/openjdk-6/
# The crash happened outside the Java Virtual Machine in native code.
# See problematic frame for where to report the bug.
#
Aborted
make: *** [test tclblend.exec] Error 134

```

Al finalizar el proceso de instalación, se instalaron los siguientes archivos:

- /opt/tclblend/bin/jtclsh: es el archivo de inicialización del tclBlend
- tcljava.jar: archivos comunes tanto para el tclBlend como para el Jacl.
- tclblend.jar: archivos específicos del tclBlend.
- libtclblend.so: librería compartida para el tclBlend.

Otras opciones para ejecutar java con TCL

Las pruebas realizadas con el tclBlend no fueron exitosas. El riesgo de fallo al ejecutar los códigos en java es muy alto y afecta al sistema completo. Además, implica utilizar software que no ha sido sometido a mantenimiento desde hace algunos años y bajar la versión de algunas de las librerías reconocidas como estables para el sistema operativo con el fin de completar los procesos de instalación (específicamente gcc y sus dependencias).

Tras revisar las implementaciones realizadas sobre la plataforma, se encontraron dos mecanismos alternativos para utilizar código desarrollado en Java sobre TCL. La primera de ellas consiste en utilizar applets enlazados directamente sobre el archivo adp de la página web a desplegar en la plataforma. En esta forma de implementación se pueden pasar parámetros tomados desde el código TCL asociado a dicha página y no queda comprometida la integridad de la instalación, ya que la utilización del recurso queda restringida al elemento que lo invoque.

```
<applet code="org.adl.samplerte.client.APIAdapterApplet.class" archive="stuff.jar" width=@app_width@ height=@app_height@></xmp>
  <param name = "code" value = "org.adl.samplerte.client.APIAdapterApplet.class" >
  <param name = "type" value="application/x-java-applet">
  <param name = "JS" value="false">
  <param name = "cookie" value="@cookie@">
  <if @debuglevel@ gt 0>
    <param name = "debug" value="true">
  </if>
  <else>
    <param name = "debug" value="false">
  </else>
  <param name = "mayscript" value="true" >
  <param name = "scriptable" value="true" >
  <if @debuglevel@ gt 0>
    <param name = "archive" value = "stuff.jar?@random@" >
  </if>
  <else>
    <param name = "archive" value = "stuff.jar?@random@" >
  </else>
</applet>
```

La segunda opción consiste en utilizar el comando `eval` directamente sobre un archivo TCL, con el fin de invocar un código en java. Este comando le delega al sistema operativo la ejecución del comando que fue brindado como cadena de texto.

```
eval "java -cp target/arhoa-0.1.jar tecDigital.App"
```

Finalmente, se encontraron referencias sobre la posibilidad de utilizar código en java invocado desde el javascript de una aplicación.

B Glosario

I

ILO: Acrónimo para *Intelligent Learning Object* o Objetos de Aprendizaje Inteligentes. Son OA que autogestionan sus atributos para interactuar con un ILOR (Gomes, 2005).

ILOR: Acrónimo de *Intelligent Learning Object Repository* o, Repositorio de Objetos de Aprendizaje Inteligentes (Gomes, 2005).

L

LOM-IEEE: Acrónimo de *Learning Object Metadata* propuesto por la IEEE. Es un estándar que describe las características de los metadatos que debería contener un OA que utilice LOM; incluye tipos de datos, longitudes y cardinalidades permitidas.

M

MoGa: Modelo de Competencias o de Gestión de Atributos

O

OA: Objeto de Aprendizaje. Según la IEEE, es “cualquier entidad, digital o no digital, que puede ser usada para el aprendizaje, educación o capacitación” (*IEEE-Learning Technology Standards Committee*, s.f.).

OAI-PMH, Acrónimo de *Open Archives Initiative Protocol for Metadata Harvesting* o Iniciativa de Protocolo de Archivos Abiertos para la Recolección de Metadatos.

R

ROA: Repositorio de Objetos de Aprendizaje o *Learning Object Repository* . Es una biblioteca de recursos digitales donde se almacenan colecciones de Objetos de Aprendizaje.

S

SPI, Acrónimo de *Simple Publishing Interface* y responde al identificador CWA 16097:2010 del CEN.

SQI, Acrónimo de *Simple Query Interface* y responde al identificador CWA 15454:2005 del Comité Europeo de Normalización(CEN).

T

TEA: Siglas utilizadas para referirse a la teoría de estilos de aprendizaje. En este documento se trabajó alrededor de las dicotomías de Felder y Silverman.

TF-IDF: Acrónimo de *Term Frequency / Inverse Document Frequency*.

U

UA: Unidad de Aprendizaje, son paquetes de materiales que pueden incluir actividades, evaluaciones, servicios, entre otros; que son ejecutables por un LMS y en su definición se pueden incluir objetivos de aprendizaje, roles, actividades y otros recursos (Chacón Rivas y Garita, 2013).

W

WEKA: Es una colección de algoritmos opensource para análisis de datos implementado por la universidad de Waikato.

Referencias

- ACM. (2012, 12 de septiembre). *6th acm conference on recommendation systems*. <http://recsys.acm.org/2012/>. Descargado de <http://recsys.acm.org/2012/>
- Adomavicius, G., y Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Xplore*, 17(6), 734–749. doi: 10.1109/TKDE.2005.99
- Advanced distributed learning initiative: SCORM 2004 4th edition*. (s.f.). <http://www.adlnet.gov/scorm/scorm-2004-4th>. Descargado 2013-04-12, de <http://www.adlnet.gov/scorm/scorm-2004-4th>
- Al-Khalifa, S., Jagadish, H., Koudas, N., Patel, J. M., Srivastava, D., y Wu, Y. (2002). Structural joins: A primitive for efficient xml query pattern matching. En *Data engineering, 2002. proceedings. 18th international conference on* (pp. 141–152).
- Anderson, M., Ball, M., Boley, H., Greene, S., Howse, N., Lemire, D., y McGrath, S. (2003, October). Racofi: A rule-applying collaborative filtering system. En *Proceedings of cola'03*. Descargado de http://www.daniel-lemire.com/fr/documents/publications/racofi_nrc.pdf
- Arjona, M., y Blando, M. (2005). Diseño instruccional, elemento clave en el desarrollo de cursos para ambientes virtuales de aprendizaje. *Recuperado el*, 6.
- Atserias, J., Casas, B., Comelles, E., González, M., Padró, L., y Padró, M. (2006). Freeling 1.3: Syntactic and semantic services in an open-source nlp library. En *Proceedings of Irec* (Vol. 6, pp. 48–55).
- Baldiris, S., Bacca, J., Rojas, A., Guevara, J., y Fabregat, R. (2011). LORSE: intelligent meta-searcher of learning objects over distributed educational repositories based on intelligent agents. En *Frontiers in education conference (FIE), 2011* (pp. F1E–1-F1E-5). doi: 10.1109/FIE.2011.6142751
- Baldiris, S., Santos, O., Boticario, J., y Fabregat, R. (2008). Los estándares educativos como herramienta de modelado de cursos que proveen adaptaciones dinámicas a

- los usuarios. *Revista Iberoamericana de Informática Educativa*.
- Baldiris, S., Santos, O. C., Barrera, C., Boticario, J., Velez, J., y Fabregat, R. (2008). Integration of educational specifications and standards to support adaptive learning scenarios in adaptaplan. *IJCSA*, 5(1), 88–107.
- Bannan-Ritland, B., Dabbagh, N., y Murphy, K. (2000). Learning object systems as constructivist learning environments: Related assumptions, theories, and applications. *The Instructional Use of Learning Objects: Online Version*. Consultado el, 24.
- Bick, E. (2006). A constraint grammar parser for spanish. *Proceedings of TIL. Bologna process*. (1999, junio). <http://www.ond.vlaanderen.be/hogeronderwijs/bologna/>. Descargado de <http://www.ond.vlaanderen.be/hogeronderwijs/bologna/>
- Boticario, J. G., y Santos, O. C. (2007). An open ims-based user modelling approach for developing adaptive learning management systems. *Journal of Interactive Media in Education*, 2, 1–19.
- Bramer, M. (2007). *Principles of data mining*. Springer.
- Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4), 331–370.
- Burke, R. (2007, enero). Hybrid web recommender systems. En P. Brusilovsky, A. Kobsa, y W. Nejdl (Eds.), *The adaptive web* (pp. 377–408). Springer Berlin Heidelberg. Descargado 2013-04-12, de http://link.springer.com/chapter/10.1007/978-3-540-72079-9_12
- Canadian engineering accreditation board. accreditation criteria and procedures*. (2010). http://www.engineerscanada.ca/e/files/Accreditation_Criteria_Procedures_2010.pdf.
- Chacón Rivas, M., y Garita, C. (2013). Comparación y evaluación de especificaciones internacionales de empaquetamiento de objetos de aprendizaje en el tec digital. *Conferencias LACLO*, 4(1).
- De Bra, P., Aerts, A., Berden, B., De Lange, B., Rousseau, B., Santic, T., ... Stash, N. (2003). Aha! the adaptive hypermedia architecture. En *Proceedings of the fourteenth*

acm conference on hypertext and hypermedia (pp. 81–84).

Declaración mundial sobre la educación superior en el siglo xxi: Visión y acción. (1998, 9 de octubre).

Drachsler, H., Hummel, H., Van den Berg, B., Eshuis, J., Waterink, W., Nadolski, R., ... Koper, R. (2009). Effects of the isis recommender system for navigation support in self-organised learning networks.

Draft standard for learning object metadata. (2002, julio). Itsc.ieee. Descargado de http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

Duval, E., Forte, E., Cardinaels, K., Verhoeven, B., Van Durm, R., Hendrikx, K., ... others (2001). The ariadne knowledge pool system. *Communications of the ACM*, 44(5), 72–78.

Ekstrand, M. D., Ludwig, M., Konstan, J. A., y Riedl, J. T. (2011). Rethinking the recommender research ecosystem: reproducibility, openness, and lenskit. En *Proceedings of the fifth acm conference on recommender systems* (pp. 133–140).

Enfoques estratégicos sobre las tics en educación en américa latina y el caribe. (2013). Descargado de <http://www.unesco.org/new/fileadmin/MULTIMEDIA/FIELD/Santiago/pdf/TICS-enfoques-estrategicos-sobre-TICs-ESP.pdf>

Felder, R. M., Felder, G., y Dietz, E. (2002). The effects of personality type on engineering student performance and attitudes. *Journal of Engineering Education*, 91(1), 3–17.

Felder, R. M., y Silverman, L. K. (1988). Learning and teaching styles in engineering education. *Engineering education*, 78(7), 674–681.

Globe alliance. (2012). Descargado de <http://www.globe-info.org/>

Gluga, R., Kay, J., y Lever, T. (2012). Foundations for modelling university curricula in terms of multiple learning goal sets.

Gomes, E. R. (2005). *Objetos inteligentes de aprendizagem: uma abordagem baseada em agentes para objetos de aprendizagem*.

Gorunescu, F. (2011). *Data mining: Concepts, models and techniques* (Vol. 12). Springer.

Gross, T. (2003). Cyclades: A distributed system for virtual community support based

- on open archives. En *Parallel, distributed and network-based processing, 2003. proceedings. eleventh euromicro conference on* (pp. 484–491).
- Gámez Suazo, I. Y. (2012). *Sistema para generación de sugerencias de rutas de aprendizaje adaptativas en entornos de e-learning*.
- Han, J., Kamber, M., y Pei, J. (2006). *Data mining: concepts and techniques*. Morgan kaufmann.
- iee-learning technology standards committee. (s.f.). Itsc.ieee. Descargado de <http://www.ieeeltsc.org:8080/Plone/>
- Ims global common cartridge profile: Implementation. (2014, 30 de mayo). Descargado de <http://www.imsglobal.org/cc/>
- Ims global learning consortium. (2012, marzo). Itsc.ieee. Descargado de <http://www.imsglobal.org/>
- Jameson, A. (2009). Adaptive interfaces and agents. *Human-Computer Interaction: Design Issues, Solutions, and Applications*, 105.
- Lagoze, C., Van de Sompel, H., Nelson, M., y Warner, S. (2002, junio). *Open archives initiative - protocol for metadata harvesting - v.2.0*. <http://www.openarchives.org/OAI/openarchivesprotocol.html>. Descargado 2013-04-12, de <http://www.openarchives.org/OAI/openarchivesprotocol.html>
- López, C. (2005). *Los repositorios de objetos de aprendizaje como soporte a un entorno e-learning*. Tesis Doctoral no publicada. Descargado 2013-04-12, de http://www.biblioweb.dgsca.unam.mx/libros/repositorios/bibliotecas_digitales.htm
- Manouselis, N., y Costopoulou, C. (2007). Analysis and classification of multi-criteria recommender systems. *World Wide Web*, 10(4), 415–441.
- Manouselis, N., Drachsler, H., Verbert, K., y Duval, E. (2012). Recommender systems for learning.
- Manouselis, N., Drachsler, H., Vuorikari, R., Hummel, H., y Koper, R. (2011). Recommender systems in technology enhanced learning. En *Recommender systems handbook* (pp. 387–415). Springer.

- Mertens, L. (1998). *La gestión por competencia laboral en la empresa y la formación profesional*. Organización de Estados Iberoamericanos para la Educación, la Ciencia y la Cultura.
- Montaner, M., López, B., y De La Rosa, J. L. (2003). A taxonomy of recommender agents on the internet. *Artificial intelligence review*, 19(4), 285–330.
- Najjar, J., Ternier, S., y Duval, E. (2003). The actual use of metadata in ariadne: an empirical analysis. En *Proceedings of the 3rd annual ariadne conference* (pp. 1–6).
- Ochoa, X., Klerkx, J., Vandeputte, B., y Duval, E. (2011). On the use of learning object metadata: The globe experience. En C. Kloos, D. Gillet, R. Crespo García, F. Wild, y M. Wolpers (Eds.), *Towards ubiquitous learning* (Vol. 6964, p. 271-284). Springer Berlin Heidelberg. Descargado de http://dx.doi.org/10.1007/978-3-642-23985-4_22 doi: 10.1007/978-3-642-23985-4_22
- Odrizola, S., Luis, J., Ochoa, X., Parra Chico, G. A., y Duval, E. (2011). La experiencia de ariadne: Creando una red de reutilización de objetos de aprendizaje a través de estándares y especificaciones. *IEEE-RITA*, 6(3), 112–117.
- Palomar, M., Saiz-Noeda, M., Muñoz, R., Suárez, A., Martínez-Barco, P., y Montoyo, A. (2001). Phora: A nlp system for spanish. En A. Gelbukh (Ed.), *Computational linguistics and intelligent text processing* (Vol. 2004, p. 126-139). Springer Berlin Heidelberg. Descargado de http://dx.doi.org/10.1007/3-540-44686-9_12 doi: 10.1007/3-540-44686-9_12
- Pascual-Nieto, I., Santos, O. C., Perez-Marin, D., y Boticario, J. G. (2011). Extending computer assisted assessment systems with natural language processing, user modeling, and recommendations based on human computer interaction and data mining. En *Proceedings of the twenty-second international joint conference on artificial intelligence-volume volume three* (pp. 2519–2524).
- Pérez-Marín, D., Alfonseca, E., y Rodríguez, P. (2006). On the dynamic adaptation of computer assisted assessment of free-text answers. En *Adaptive hypermedia and adaptive web-based systems* (pp. 374–377).

- Perez-Marin, D., Alfonseca, E., Rodriguez, P., y Pascual-Nieto, I. (2006). Willow: Automatic and adaptive assessment of students free-text answers. En *Proceedings of the 22nd international conference of the spanish society for the natural language processing (sepln)*.
- Propuestas y acciones para la transformación de la educación superior en américa latina.* (s.f.). Descargado de <http://www.6x4uealc.org>
- Resnick, P., y Varian, H. R. (1997, marzo). Recommender systems. , *40*(3), 56–58. Descargado 2013-04-12, de <http://doi.acm.org/10.1145/245108.245121> doi: 10.1145/245108.245121
- Ricci, F., Rokach, L., y Shapira, B. (2011). *Recommender systems handbook*. Springer.
- Russell, S., y Norvig, P. (2002). *Artificial intelligence: A modern approach* (2.^a ed.). Prentice Hall.
- Santos, O. C., y Boticario, J. G. (2010). Modeling recommendations for the educational domain. *Procedia Computer Science*, *1*(2), 2793–2800.
- Santos, O. C., y Boticario, J. G. (2011). Requirements for semantic educational recommender systems in formal e-learning scenarios. *Algorithms*, *4*(2), 131–154.
- Simon, B., Massart, D., van Assche, F., Ternier, S., Duval, E., Brantner, S., . . . Miklós, Z. (2005). A simple query interface for interoperable learning repositories. En *Proceedings of the 1st workshop on interoperability of web-based educational systems* (pp. 11–18).
- Stash, N., y De Bra, P. (2004). Incorporating cognitive styles in aha!(the adaptive hypermedia architecture). En *Proceedings of the iasted international conference web-based education* (pp. 378–383).
- Sternberg, R. J., y Grigorenko, E. L. (1997). Are cognitive styles still in style? *American psychologist*, *52*, 700–712.
- Suazo, I. G., Rodriguez, C. G., y Rivas, M. C. (2012). Generating adaptive learning paths in e-learning environments. En *Informatica (clei), 2012 xxxviii conferencia latinoamericana en* (pp. 1–10).

- Síntesis de la legislación de la ue.* (s.f.). http://europa.eu/legislation_summaries/education_training_youth/general_framework/index_es.htm.
- Tatarinov, I., Viglas, S. D., Beyer, K., Shanmugasundaram, J., Shekita, E., y Zhang, C. (2002). Storing and querying ordered xml using a relational database system. En *Proceedings of the 2002 acm sigmod international conference on management of data* (pp. 204–215).
- Vélez Ramos, J. B., Baldiris, S., Nassiff, S., y Fabregat Gesa, R. (2008). Generación de cursos virtuales adaptativos basados en scorm e ims-ld. *Avances en Sistemas e Informática*, 5(3), 49–59.
- Wang, L. (2007). Sociocultural learning theories and information literacy teaching activities in higher education. *Reference & User Services Quarterly*, 47(2), 149–158.
- Webb, G. I., Pazzani, M. J., y Billsus, D. (2001). Machine learning for user modeling. *User modeling and user-adapted interaction*, 11(1-2), 19–29.
- WeiB, G. (1999). *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT Press.
- Wiley, D. A. (2003). *Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy*.
- Witkin, H. A., Goodenough, D. R., y cols. (1981). Cognitive styles: essence and origins. field dependence and field independence. *Psychological issues*(51), 1.
- Witten, I. H., Frank, E., y Hall, M. A. (2011). *Data mining: Practical machine learning tools and techniques: Practical machine learning tools and techniques*. Elsevier.
- Zapata-Gonzalez, A., Menendez, V., Prieto, M., y Romero, C. (2011). Using data mining in a recommender system to search for learning objects in repositories. En *The 4th international conference on educational data mining (edm 2011) eindhoven, netherlands*.