

2014

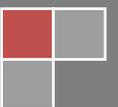
# Integración de sistemas de recomendación en un entorno educativo virtual

Proyecto de Grado

Diego Abete Luzi  
Juan Gabriel Ibañez Gonnet

Tutor: Libertad Tansini

Montevideo, Uruguay 2014





## Resumen

La plataforma Moodle es una aplicación web que se utiliza como ambiente educativo virtual. Es un sistema de gestión de cursos que ayuda a los educadores a crear comunidades de aprendizaje en línea. En ella se puede subir todo tipo de materiales, como libros, ejercicios, artículos académicos, etc. Estos materiales junto con información descriptiva son denominados Objetos de Aprendizaje (OA). Muchas veces en la plataforma se publica gran cantidad de material educativo por lo que resulta difícil buscar algún artículo o encontrar el material deseado. Para abordar este problema, se desarrolló un módulo de recomendación de OA que también permite la búsqueda de los mismos a través de sus metadatos. Las recomendaciones se realizan usando diversas técnicas de recomendación basadas en las preferencias de los usuarios y las relaciones entre ellos. En este caso se asume que la preferencia de un usuario puede estar dada en base a las calificaciones que otorgó a los OA utilizados y a partir de esta se pueden recomendar OA similares.

Para el funcionamiento de este sistema de recomendación, se desarrolló un módulo de extracción de OA y usuarios de la plataforma Moodle. Estos datos se almacenan en un repositorio y son utilizados por el sistema desarrollado para ejecutar la recomendación en la plataforma. Además se permite evaluar los recursos recomendados y estas evaluaciones se utilizan tanto para mejorar, como para medir la utilidad y efectividad de las recomendaciones.

El módulo de recomendación se encuentra actualmente integrado con Moodle en un curso de la Facultad de Ingeniería y fue puesto a prueba con 27 estudiantes. Las recomendaciones permitieron demostrar su efectividad ya que más del 80% de los OA recomendados fueron calificados con puntajes altos.

Este trabajo permite concluir que los sistemas de recomendación traen importantes beneficios en los entornos educativos ayudando a los usuarios a encontrar los materiales más adecuados para sus necesidades contribuyendo así al proceso de aprendizaje académico.

**Palabras claves:** Sistemas de recomendación, algoritmo de recomendación, filtrado colaborativo, Moodle, metadatos, educación virtual.



# Agradecimientos

Agradecemos a todas las personas que colaboraron en este proyecto, especialmente a la profesora Regina Motz que nos permitió experimentar con su curso, al personal de la Unidad de Recursos Informáticos que nos ayudó a montar toda la infraestructura necesaria, a los estudiantes que se ofrecieron generosamente como parte del experimento, y muy especialmente a nuestra tutora Libertad Tansini que nos apoyó y guio incondicionalmente durante todo el transcurso de este proyecto.



# Índice

<b>RESUMEN</b> .....	<b>2</b>
<b>AGRADECIMIENTOS</b> .....	<b>4</b>
<b>ÍNDICE</b> .....	<b>6</b>
<b>ÍNDICE DE FIGURAS</b> .....	<b>10</b>
<b>GLOSARIO</b> .....	<b>12</b>
<b>1 INTRODUCCIÓN</b> .....	<b>14</b>
1.1 ÁREA DE ESTUDIO .....	14
1.2 OBJETIVOS DEL PROYECTO .....	14
1.2.1 Extracción de OA .....	15
1.2.2 Recomendación de OA .....	15
1.2.3 Búsqueda de OA .....	17
1.2.4 Sitio web .....	18
1.2.5 Resultados .....	18
1.3 ESTRUCTURA DEL DOCUMENTO .....	18
<b>2 ESTADO DEL ARTE</b> .....	<b>20</b>
2.1 OBJETOS DE APRENDIZAJE .....	20
2.1.1 Definición .....	20
2.1.2 Granularidad .....	21
2.1.3 Metadatos .....	21
2.1.4 Estándares de metadatos .....	22
2.1.4.1 Learning Object Metadata (LOM) .....	22
2.1.4.2 Learning Object Metadata (LOM-ES) .....	23
2.1.4.3 Dublin Core (DC) .....	23
2.1.4.4 Sharable Content Object Reference Model (SCORM) .....	24
2.1.5 Repositorios de Objetos de Aprendizaje .....	24
2.1.5.1 GLOBE .....	25
2.1.5.2 MERLOT .....	25
2.1.5.3 ARIADNE .....	25
2.1.5.4 CAREO .....	26
2.2 MOODLE .....	26
2.2.1 Integración de módulo de recomendaciones con Moodle .....	28
2.2.1.1 Módulo Moodle .....	28
2.2.1.2 Bloque HTML .....	29
2.2.1.3 Recurso URL .....	29
2.2.1.4 Herramienta externa .....	29
2.2.2 Extracción de metadatos de OA .....	30
2.3 TÉCNICAS DE RECOMENDACIÓN .....	30
2.3.1 Crowdsourcing .....	30
2.3.2 Friendsourcing .....	31

2.3.3	<i>Filtrado colaborativo</i> .....	31
2.3.3.1	Basados en contenido .....	32
2.3.3.2	Basados en la demografía de los usuarios .....	33
2.3.3.3	Basados en la utilidad .....	33
2.3.3.4	Basados en el conocimiento .....	33
2.3.3.5	Comparaciones entre las técnicas de recomendación .....	34
2.3.4	<i>Recomendaciones grupales</i> .....	36
<b>3</b>	<b>ANÁLISIS</b> .....	<b>38</b>
3.1	EXTRACCIÓN DE OBJETOS DE APRENDIZAJE EN MOODLE .....	38
3.1.1	<i>Alternativas para la extracción</i> .....	39
3.1.1.1	Extracción web .....	39
3.1.1.2	Extracción directa .....	39
3.1.2	<i>Evaluación de herramientas de extracción de datos</i> .....	41
3.1.3	<i>Descripción del problema</i> .....	43
3.1.4	<i>Solución propuesta para la extracción</i> .....	44
3.2	RECOMENDACIÓN .....	45
3.2.1	<i>Evaluación de técnicas de recomendación</i> .....	46
3.2.1.1	Friendsourcing .....	46
3.2.1.2	Crowdsourcing .....	47
3.2.1.3	Recomendaciones grupales .....	48
3.2.1.4	Conclusión de la evaluación de las técnicas de recomendación .....	48
3.2.2	<i>Evaluación de herramientas de recomendación</i> .....	49
3.2.3	<i>Descripción del problema</i> .....	49
3.2.4	<i>Solución propuesta para las recomendaciones y búsquedas</i> .....	50
<b>4</b>	<b>DISEÑO</b> .....	<b>56</b>
4.1	ARQUITECTURA.....	58
4.1.1	<i>Capa de presentación</i> .....	58
4.1.1.1	Seguridad .....	59
4.1.2	<i>Capa de negocio</i> .....	60
4.1.2.1	Extracción.....	61
4.1.2.2	Recomendación .....	62
4.1.3	<i>Capa de persistencia</i> .....	63
4.2	RECOMENDACIONES INTEGRADAS A LA PLATAFORMA MOODLE .....	65
4.2.1	<i>Alternativas para la integración</i> .....	65
4.2.2	<i>Solución</i> .....	66
<b>5</b>	<b>PUESTA EN PRODUCCIÓN</b> .....	<b>68</b>
5.1	INSTALACIÓN .....	68
5.2	PRUEBAS DE PERFORMANCE .....	68
<b>6</b>	<b>EVALUACIÓN DE ALGORITMOS</b> .....	<b>72</b>
6.1	MÉTRICAS .....	72
6.1.1	<i>Predictive accuracy</i> .....	72
6.1.2	<i>Classification accuracy</i> .....	73
6.1.3	<i>Rank accuracy</i> .....	74
6.2	SELECCIÓN DE DATOS PARA LA EVALUACIÓN.....	74

6.3	EVALUACIÓN DE ALGORITMOS CON APACHE MAHOUT.....	74
<b>7</b>	<b>RESULTADOS .....</b>	<b>76</b>
<b>8</b>	<b>CONCLUSIONES Y TRABAJOS A FUTURO .....</b>	<b>80</b>
8.1	CONCLUSIONES.....	80
8.2	TRABAJOS FUTUROS.....	81
	<b>BIBLIOGRAFÍA .....</b>	<b>84</b>
	<b>ANEXO .....</b>	<b>90</b>
1.	DESCRIPCIÓN DE ACTIVIDADES Y RECURSOS QUE SE PUEDEN INCLUIR EN MOODLE.....	92
2.	PROCESO DE FILTRADO COLABORATIVO .....	98
	<i>Algoritmos basados en los ítems.....</i>	<i>98</i>
	Similaridad.....	98
	Similaridad basada en el coseno .....	98
	Correlación de Pearson .....	99
	Similaridad basada en el coseno ajustado .....	99
	Predicción.....	99
	Suma Ponderada .....	100
	Regresión .....	100
	<i>Algoritmos basados en los usuarios.....</i>	<i>100</i>
3.	RECOMENDACIONES GRUPALES .....	102
	<i>Relevancia grupal.....</i>	<i>102</i>
	<i>Discrepancia grupal .....</i>	<i>102</i>
	<i>Función consenso .....</i>	<i>103</i>
4.	HERRAMIENTA DE EXTRACCIÓN DE DATOS: WEB HARVEST .....	104
5.	HERRAMIENTA DE RECOMENDACIÓN: APACHE MAHOUT .....	105
6.	SEGURIDAD CON APACHE SHIRO.....	107
7.	INTERFAZ DE USUARIO .....	109
8.	MANUAL DE INSTALACIÓN .....	114
9.	DETALLE DE OA EXTRAÍDOS DEL CURSO FUNDAMENTOS DE LA WEB SEMÁNTICA.....	117



# Índice de figuras

FIGURA 1 - MODELO DE DOMINIO DE MÓDULO DE EXTRACCIÓN.....	43
FIGURA 2 - MODELO DE CASOS DE USO DE MÓDULO DE EXTRACCIÓN .....	45
FIGURA 3 - DIAGRAMA DE FLUJO TÍPICO DE MÓDULO DE EXTRACCIÓN .....	46
FIGURA 4 - MODELO DE DOMINIO DE MÓDULO DE RECOMENDACIÓN .....	50
FIGURA 5 - MODELO DE CASOS DE USO DE MÓDULO DE RECOMENDACIÓN .....	51
FIGURA 6 - MODELO DE CASOS DE USO DE ADMINISTRACIÓN.....	53
FIGURA 7 - DIAGRAMA DE SECUENCIA DEL FLUJO PRINCIPAL DEL SISTEMA DE RECOMENDACIONES.....	54
FIGURA 8 - DIAGRAMA DE FLUJO TÍPICO DE ACCESO A MOODLE CON RECOMENDACIONES EMBEBIDAS.....	54
FIGURA 9 - DIAGRAMA DE FLUJO DE ACTIVIDADES DE ADMINISTRACIÓN.....	55
FIGURA 10 - DIAGRAMA DE CLASES DEL SISTEMA .....	58
FIGURA 11 - ARQUITECTURA DEL SISTEMA.....	59
FIGURA 12 - DIAGRAMA DE PAQUETES DE CAPA DE PRESENTACIÓN.....	60
FIGURA 13 - DIAGRAMA DE PAQUETES DE CAPA DE NEGOCIO .....	61
FIGURA 14 - DIAGRAMA DEL MODELO DE DATOS .....	64
FIGURA 15 - PANTALLA DE RECOMENDACIÓN INTEGRADA A MOODLE .....	67
FIGURA 16 - ZOOM DE PANTALLA DE RECOMENDACIÓN INTEGRADA A MOODLE.....	67
FIGURA 17 - GRÁFICA DE EVOLUCIÓN DE PUNTAJES EN EL TIEMPO .....	78
FIGURA 18 - GRÁFICA DE EVOLUCIÓN DE PUNTAJES DE UN OA EN EL TIEMPO .....	78
FIGURA 19 - PANTALLA DE INICIO .....	109
FIGURA 20 - PANTALLA DE BÚSQUEDA DE OA.....	110
FIGURA 21 - PANTALLA DE RECOMENDACIONES DE FILTRADO COLABORATIVO.....	110
FIGURA 22 - PANTALLA DE RECOMENDACIONES GRUPALES .....	111
FIGURA 23 - PANTALLA DE DETALLE DE OA.....	112
FIGURA 24 - PANTALLA DE CONTACTO .....	113
FIGURA 25 - CONFIGURACIÓN REENVÍO DE PUERTOS EN VIRTUALBOX.....	115
FIGURA 26 - AGREGAR BLOQUE HTML EN MOODLE .....	116
FIGURA 27 - EDITAR BLOQUE HTML EN MOODLE .....	116



# Glosario

**AJAX:** Acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML). Es una técnica de desarrollo web para crear aplicaciones interactivas.

**API:** Acrónimo de Application Programming Interface. Es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

**BD:** Base de datos.

**Bot:** Un bot es un programa informático que imita el comportamiento de un humano.

**DC:** Acrónimo de Dublin Core. Es un estándar de metadatos.

**EVA:** Entorno Virtual de Aprendizaje.

**Fing:** Facultad de Ingeniería.

**Framework:** Es una infraestructura digital, conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software. Típicamente puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

**HTML:** Acrónimo de HyperText Markup Language.

**IEEE:** Instituto de Ingeniería Eléctrica y Electrónica.

**INCO:** Instituto de Computación de la Facultad de Ingeniería.

**LOAR:** Término acuñado en este proyecto a las actividades y recursos de Moodle.

**LOM:** Acrónimo de Learning Object Metadata, estándar de metadatos.

**MAE:** Acrónimo de Mean Average Error.

**Moodle:** Acrónimo de Module Object-Oriented Dynamic Learning Environment. Es una aplicación web de tipo Ambiente Educativo Virtual, un sistema de gestión de cursos, de distribución libre, que ayuda a los educadores a crear comunidades de aprendizaje en línea.

**MoodleRecommender:** Nombre acuñado al sitio web desarrollado en este proyecto.

**OA:** Acrónimo de Objeto de Aprendizaje.

**PDF:** Es un formato de almacenamiento de documentos digitales independiente de plataformas de software o hardware.

**RMS:** Acrónimo de Root Mean Squareroot.

**ROA:** Acrónimo de Repositorio de Objetos de Aprendizaje.

**SCORM:** Acrónimo de Sharable Content Object Reference Model, estándar de metadatos.

**UdelaR:** Universidad de la República.

**URI:** Unidad de Recursos Informáticos de la Facultad de Ingeniería.

**XML:** Acrónimo de eXtensible Markup Language. Es un lenguaje de marcas utilizado para almacenar datos en forma legible.



# 1 Introducción

## 1.1 Área de estudio

El proyecto está orientado al área de sistemas de recuperación y recomendación de información. Dentro de este marco, el trabajo se enfoca en el área de la educación, más específicamente, en los entornos educativos virtuales.

## 1.2 Objetivos del proyecto

En el ámbito de los entornos educativos virtuales, la plataforma Moodle [1] es una de las más utilizadas ya que entre otras ventajas, es de libre acceso, posee mantenimiento continuo y además, ha sido probada en diversas universidades y centros educativos. En Uruguay, esta plataforma es utilizada por la Universidad de la República (UdelaR) [2] y se denomina EVA (Entorno Virtual de Aprendizaje) [3]. Este sistema permite a los docentes administrar cursos educativos donde pueden publicar materiales, ejercicios, presentaciones, videos, etc., facilitándoles el acceso a los estudiantes. Estos elementos, junto con la información que los describe, se denominan comúnmente como Objetos de Aprendizaje (OA) [4]. Los estudiantes pueden suscribirse a los cursos y participar de forma activa, ya sea a través de foros, chats, realizando ejercicios, encuestas y todo tipo de actividades. El docente puede realizar un seguimiento de las actividades del estudiante y eventualmente incentivarlo con nuevos materiales educativos. De esta forma se logra personalizar el aprendizaje sin importar las distancias.

La Facultad de Ingeniería (FING) [5] cuenta con este sistema desde principios del año 2008 y actualmente posee alrededor de 400 cursos, lo que implica que miles de estudiantes acceden constantemente a la plataforma para realizar diversas actividades. El hecho de que exista una gran cantidad de cursos, conlleva a que el número de OA existentes en la plataforma sea enorme, lo cual puede perjudicar el fácil y rápido acceso al material deseado por parte de estudiantes y docentes. Por lo tanto, para solucionar este problema se propone desarrollar un sistema de recomendación de OA para estudiantes y docentes vinculados a cursos de Moodle. Para lograr este cometido, se plantean cuatro objetivos principales. El primero es contar con un módulo de extracción que permita extraer OA de Moodle. El segundo objetivo es contar con un módulo de recomendación de OA que facilite el acceso a los mismos por parte de docentes y estudiantes. El tercer objetivo es contar con un motor de búsqueda de OA. Y por último, el cuarto objetivo es contar con un sitio web que integre los módulos mencionados, donde estudiantes y docentes puedan registrarse y acceder para interactuar con los mismos.

Otros objetivos que se buscan alcanzar en este trabajo son: investigar la posibilidad de extensión de la enseñanza aprovechando la plataforma Moodle, analizar la viabilidad de integrar a la plataforma las técnicas de recomendación, y experimentar con usuarios reales la

recomendación de OA en un curso de la Facultad de Ingeniería. También es un objetivo de este trabajo brindar propuestas de mejoras y extensiones de los módulos como trabajos futuros.

A continuación se describen los principales objetivos de este trabajo.

### 1.2.1 Extracción de OA

El primer objetivo de este proyecto es desarrollar un módulo de extracción. La extracción de OA de Moodle consiste en recuperar y almacenar en un repositorio propio la referencia al objeto en el Moodle junto con la mayor cantidad de metadatos disponibles. Se entiende por metadatos [6] a cierta información adicional del elemento educativo en sí (*datos sobre los datos*). Por ejemplo, si el recurso en cuestión es un video, los metadatos asociados al mismo podrían ser el título, autor, duración, temática, idioma, tamaño, etc. Muchos metadatos son comunes a todos los OA, es por eso que se definen estándares de metadatos, ver sección 2.1.3.

La extracción de OA es necesaria para el funcionamiento del módulo de recomendaciones ya que en un principio no se cuenta con un repositorio de los OA existentes en la plataforma, y por lo tanto, no se tienen elementos para recomendar. El módulo de extracción además permite recuperar información sobre los usuarios, la cual es necesaria para generar las recomendaciones personalizadas.

Debido a que el sistema utiliza datos personales tales como nombre, apellido y correo electrónico para generar las recomendaciones, previo a realizar la extracción se emitió un comunicado a los estudiantes con los cuales se iba poner a prueba el sistema, indicándoles que estos datos iban a ser utilizados únicamente con fines estadísticos y bajo ningún concepto iban a ser distribuidos a terceros.

Una vez puesto en funcionamiento el módulo de recomendaciones, se debe repetir el proceso de extracción con regularidad a medida que se van incorporando nuevos OA en la plataforma.

En este trabajo se analizan las distintas alternativas para realizar la extracción. En términos generales, las dos grandes posibilidades evaluadas son:

- A través de la web. Esta técnica consiste en descargar y procesar automáticamente el contenido HTML del sitio web de Moodle en busca de OA.
- Extraer la información directamente desde el servidor Moodle. Para ello se debe contar con acceso de lectura en el sistema de archivos y en la base de datos.

El acceso al módulo de extracción se restringe sólo a usuarios con rol administrador. Los OA extraídos de Moodle pueden visualizarse en este sitio junto con sus metadatos.

### 1.2.2 Recomendación de OA

Como segundo objetivo se tiene el desarrollo de un módulo de recomendación de OA y la integración de este módulo con la plataforma Moodle.

Los sistemas o módulos de recomendación se basan en un tipo específico de técnica de filtrado de información, los cuales tienen como objetivo presentar distintos temas o ítems de información como pueden ser películas, música, noticias, libros, etc., que son de interés para cada usuario en particular. Generalmente un sistema de recomendaciones compara el perfil del usuario con algunas características de referencia de los temas y busca predecir el ranking o ponderación que el usuario le daría a un ítem que aún no ha accedido. Para realizar recomendaciones existen diversas técnicas. Algunas de las que serán evaluadas en este proyecto son Filtrado Colaborativo (FC) [7], Friendsourcing [8] y Recomendaciones grupales [9]. Por más detalles ver la sección 2.3.

Hoy en día grandes empresas como Facebook [10], Amazon [11], Last.fm [12], MySpace [13], utilizan sistemas de esta índole para recomendar amigos, productos, canciones, personas, etc. También existen diversos trabajos de investigación relacionados con la recomendación de objetos en distintos contextos. Hasta el día de hoy no hemos encontrado estudios concretos realizados en un Entorno Educativo (EE) como Moodle, por lo cual este proyecto resulta novedoso y abre las puertas a futuros trabajos de investigación como realizar un análisis de impacto más profundo de obtener recomendaciones en un EE. Si bien éste no es el objetivo principal de este proyecto, se realiza un estudio con usuarios de un curso de la Facultad de Ingeniería obteniendo resultados muy positivos. Por más detalles ver la sección 7.

A continuación se analizan en profundidad las tres primeras técnicas para decidir cuales se pueden utilizar en EE y cuales pueden resultar de utilidad, considerando los datos disponibles y las posibilidades técnicas.

La técnica de Filtrado Colaborativo consiste en sugerir nuevos objetos o predecir la utilidad de cierto objeto para un usuario en particular, basándose en sus intereses y en las opiniones de otros usuarios con intereses similares. Esta técnica es una forma de Crowdsourcing [14], que se refiere a la colaboración abierta de varias personas para realizar una tarea. En este contexto, la colaboración sería puntuar o calificar los OA a los que accede cada usuario. De esta forma, si todos colaboran, esta técnica permite encontrar similitudes entre usuarios u objetos y así poder generar las recomendaciones.

A diferencia de Crowdsourcing, la técnica denominada Friendsourcing obtiene información adecuada a partir de un confiable y eventualmente reducido grupo de personas que están conectadas o relacionadas socialmente en una red. En Friendsourcing, la información no se obtiene a través de una multitud sino a través de un grupo de personas cuidadosamente seleccionado, por lo cual se convierte en una fuente de información de alta calidad. En Moodle la posibilidad de crear vínculos entre usuarios es nula. Si bien es posible tomar como vínculo la existencia de chats entre usuarios, o tomar las relaciones de amistad que los usuarios tienen en una red social como puede ser Facebook u otra, esto requiere autorización por parte de todos los usuarios lo cual parece ser inviable. Por lo tanto, en este trabajo se decide descartar esta técnica debido a la ausencia de información sobre las relaciones entre estudiantes.

Por último, la recomendación grupal es una técnica diseñada para contextos donde más de un usuario está involucrado en el proceso de recomendación. Es decir, se realizan recomendaciones a grupos de forma que la mayor parte de los integrantes queden conformes. Estas recomendaciones se hacen en base a la combinación de las preferencias individuales de los usuarios.

En este trabajo se estudian las distintas técnicas y se realiza un análisis comparativo focalizando en las ventajas y desventajas de cada una. Luego se determina que técnica es la que más se adecúa a la plataforma Moodle.

Para que las recomendaciones sean de utilidad, es necesario presentarlas en un lugar que sea de fácil acceso por parte de los estudiantes, por lo que resulta interesante realizar una integración entre la plataforma Moodle y el módulo de recomendaciones. Para lograr dicha integración, se realiza una investigación sobre las propiedades de la plataforma y se analizan distintas alternativas para la implementación e integración de las recomendaciones con Moodle:

- Desarrollar un módulo de Moodle. Esto es, aprovechar el diseño modular de Moodle, lo que permite realizar importantes modificaciones sobre la plataforma sin necesidad de cambiar el código fuente existente. Los principales tres tipos de módulos que se pueden desarrollar para Moodle son actividades, recursos y bloques. Estos tipos de módulos permiten dar contenido al curso. Por ejemplo, una actividad puede ser un ejercicio, un recurso puede ser un libro y un bloque puede ser un calendario en la página principal del curso.
- Utilizar una actividad, recurso o bloque predefinido por Moodle. Estos elementos son parte de las funcionalidades base de la plataforma. Los mismos consisten en una diversa selección de ítems configurables.

Por más detalles ver la sección 2.2.

Concluido dicho análisis, se decide aprovechar los elementos ya provistos por la plataforma Moodle y desarrollar un script que agregándolo dentro de un Bloque HTML (bloque existente que permite ingresar código HTML) se logra la integración de los sistemas, es decir, se presentan las recomendaciones en la página principal del curso de Moodle.

### **1.2.3 Búsqueda de OA**

El tercer objetivo de este trabajo es contar con un motor de búsqueda avanzado. El mismo permite realizar búsquedas utilizando la información que describe a los recursos, es decir, los metadatos. Para la organización de estos metadatos existen estándares. El propósito de los estándares de metadatos como por ejemplo Learning Object Metadata (LOM) o Dublin Core (ver sección 2.1.4) es facilitar la búsqueda, evaluación, adquisición y uso de los objetos de aprendizaje. Estas actividades son comúnmente realizadas por alumnos y docentes, pero debido al uso de estándares, también pueden ser llevadas a cabo por procesos automáticos. Además, permiten compartir OA con facilidad, así como el intercambio de los mismos. Esta interoperabilidad aumenta el valor de los repositorios de OA (ROA, ver sección 2.1.5) ya que al

colaborar entre sí, los ROA pueden mejorar los resultados de las consultas y así brindar un mejor servicio a sus usuarios.

#### 1.2.4 Sitio web

Por último, se desarrolla un sitio web denominado *MoodleRecommender*, en el cual se centralizan los módulos. Un usuario registrado puede tener acceso al motor de búsqueda de OA y a las recomendaciones personalizadas y grupales. Para las recomendaciones personalizadas, el usuario puede seleccionar con qué técnica desea recibir las mismas. Las técnicas pueden ser; recomendaciones basadas en usuarios con gustos similares, o recomendaciones basadas en OA similares a otros OA que el usuario calificó como útiles. Para las recomendaciones grupales, el usuario debe crear un grupo de usuarios o seleccionar uno ya existente, y luego podrá obtener recomendaciones para dicho grupo. El motor de búsqueda permite realizar dos tipos de búsqueda: búsqueda rápida y búsqueda avanzada. En la búsqueda rápida, el usuario ingresa un texto y el sistema busca correspondencias dentro de los metadatos de todos los OA. Mientras que la búsqueda avanzada, permite buscar OA por diferentes tipos de metadatos como: título, autor, fecha de creación, temática, año de publicación, etc.

#### 1.2.5 Resultados

El sistema fue puesto a prueba con estudiantes de un curso de la Facultad de Ingeniería. Se solicitó a los mismos que hagan uso del sistema y califiquen las recomendaciones recibidas. Aproximadamente 30 estudiantes participaron en el experimento.

El resultado fue muy positivo, se obtuvieron muchas calificaciones altas, y a medida que más usuarios calificaban, más precisas eran las recomendaciones. En la sección 7 se analiza en profundidad la eficacia de las recomendaciones. Los algoritmos de recomendación en general resultaron ser eficientes y precisos, brindando recomendaciones acertadas en más del 80% de los casos. Es por esto que se puede concluir que el uso de sistemas de recomendación en EE resulta muy beneficioso para los estudiantes y también para los docentes, ya que éstos últimos pueden encontrar OA de interés para sus cursos, como también analizar las preferencias de sus estudiantes mediante las recomendaciones grupales.

Como conclusión de este trabajo se puede decir que este tipo de sistema permite mejorar sustancialmente el uso de Moodle ya que los estudiantes pueden encontrar y acceder fácilmente a materiales de potencial interés e incluso puede introducirlos en áreas que desconocían.

### 1.3 Estructura del documento

Este documento se encuentra estructurado en ocho capítulos y un anexo compuesto por nueve secciones. El primer capítulo describe la realidad actual del área de estudio junto a los objetivos propuestos para resolver los problemas planteados. Luego, se presenta un resumen de la

solución junto a los resultados obtenidos. El segundo capítulo describe los conceptos teóricos manejados en el marco de este trabajo, utilizados en el desarrollo de la investigación y propuesta de la solución. Los capítulos 3 y 4 corresponden al estudio detallado del problema y posterior desarrollo de la solución. En el capítulo 5 se describen los pasos seguidos en el proceso de instalación de la aplicación desarrollada en el ambiente de producción, y se presentan los resultados de las pruebas de performance llevadas a cabo. En el capítulo 6 se realiza una introducción a las distintas métricas de evaluación y se detallan los procedimientos efectuados para la evaluación de los algoritmos. En el capítulo 7 se detallan los resultados obtenidos en la evaluación de algoritmos. El capítulo 8 se divide en dos secciones que especifican las conclusiones relevadas a lo largo del proyecto y las posibles extensiones denominadas como trabajos a futuro. Luego se detalla la bibliografía utilizada en el proyecto. Finalmente se presenta un anexo dividido en nueve secciones.

- Descripción de Actividades y Recursos que se pueden incluir en Moodle
- Proceso de Filtrado Colaborativo
- Recomendaciones grupales
- Herramienta de extracción de datos: Web Harvest
- Herramienta de recomendación: Apache Mahout
- Seguridad con Apache Shiro
- Interfaz de usuario
- Manual de instalación
- Detalle de OA extraídos del curso Fundamentos de la Web Semántica

## 2 Estado del arte

En este capítulo se realiza un estudio de los conceptos más importantes del área que abarca este proyecto. Se definen los términos utilizados en el ámbito de la educación virtual, se realiza un análisis en profundidad de la plataforma Moodle y las distintas alternativas para integrar un sistema de recomendación de recursos en el mismo. Finalmente se lleva a cabo un extenso estudio sobre técnicas de recomendación.

### 2.1 Objetos de aprendizaje

Según Wiley (2000) [4] los objetos de aprendizaje son elementos de enseñanza, basados en el paradigma de *orientación a objetos* de las ciencias de computación [15]. Dicho paradigma maneja componentes llamados "objetos" que pueden ser reutilizados en varios contextos. Esta es la idea fundamental detrás de los objetos de aprendizaje. Los docentes pueden crear pequeños componentes de enseñanza los cuales pueden ser reutilizados en diferentes contextos de aprendizaje.

Si bien son relativamente nuevos, los objetos de aprendizaje (OA) han sido definidos de varias formas según los diferentes autores. Por ejemplo, el Comité de Estandarización de Tecnología Educativa (Learning Technology Standards Committee - LTSC) del Instituto de Ingenieros Eléctricos y Electrónicos (Institute of Electrical and Electronics Engineers - IEEE) [16] define a un objeto de aprendizaje como "*cualquier entidad, digital o no digital, que puede ser utilizada para el aprendizaje, educación, o entrenamiento*". Mientras que Wiley (2000) los define como "*cualquier recurso digital que puede ser reutilizado para apoyar el aprendizaje*".

Mason, Weller y Pegler (2003) [17] los definen como "*una pieza digital de material de aprendizaje que direcciona a un tema claramente identificable o salida de aprendizaje y que tiene el potencial de ser reutilizado en diferentes contextos*". Otras definiciones interesantes son las de Morales y García (2005) [18]: "*Unidad de aprendizaje independiente y autónomo que está predispuesto a su reutilización en diversos contextos instruccionales*". Y las de The JORUM+ Project Teams (2004) [19] que los define como "*un objeto de aprendizaje es cualquier recurso que puede ser utilizado para facilitar la enseñanza y el aprendizaje y que ha sido descrito utilizando metadatos*".

#### 2.1.1 Definición

Si bien existen diversas definiciones, todas ellas muy similares pero con pequeñas diferencias entre sí, en el marco de este proyecto se asume como un objeto de aprendizaje a una entidad de carácter digital con fines educativos, la cual está compuesta por un conjunto de metadatos y contenido.

Los OA se caracterizan por tener las siguientes propiedades [20]:

- Reusabilidad: Los OA deben ser capaces de ser combinados con otros para generar nuevos OA más complejos.
- Accesibilidad: Los OA deben facilitar el acceso por parte de los usuarios a sus contenidos.
- Portabilidad: Los OA deben poder utilizarse en diferentes sistemas de Gestión de Aprendizaje.
- Durabilidad: La información que proveen los OA debe perdurar en el tiempo.
- Interoperabilidad: Los OA deben poder trabajar en varios sistemas.
- Escalabilidad: Los OA pueden ser integrados a estructuras más complejas dentro del dominio de aprendizaje para el que fueron creados, sin que esto implique un crecimiento proporcional de costos.

Dentro de los beneficios de los objetos de aprendizaje en un entorno educativo, se encuentran los siguientes:

- Brindan mayor flexibilidad para utilizarse en diversos contextos.
- El uso de metadatos provee una mejor administración del contenido.
- La adaptabilidad facilita la selección y composición de recursos.

### 2.1.2 Granularidad

El tamaño de un objeto de aprendizaje es variable, y se conoce como granularidad. La forma en la que los recursos se agregan o unen entre sí para formar un objeto más complejo puede ayudar a definir su granularidad. También puede ser su tamaño en relación al número de páginas, duración o tamaño del archivo. Sin embargo, el mejor criterio para definir la granularidad de un objeto es por sus propósitos u objetivos. El contenido de un objeto de aprendizaje puede variar desde un simple texto, hasta una imagen, un video, etc. A estos OA, junto con otros elementos simples de enseñanza, se los conoce como objetos de aprendizaje de *Granularidad Fina*. Mientras que a un conjunto de materiales de enseñanza más organizado, con cierta complejidad como puede ser un Tema, o incluso un Curso completo, se los denomina objetos de aprendizaje de *Granularidad Gruesa*.

### 2.1.3 Metadatos

Los metadatos se definen como “*datos de los datos*” y son un conjunto de atributos necesarios para describir un recurso. A través de ellos se tiene una noción general del objeto al cual está ligado el metadato, conociendo rápidamente sus características. Los bibliotecarios han utilizado esta idea desde hace cientos de años. Un catálogo de biblioteca permite a los bibliotecarios organizar sus libros para obtener mejores resultados en las búsquedas por autor, año, tema, etc. En general, los metadatos sirven para resumir el significado de los datos, facilitar búsquedas a los usuarios, determinar si efectivamente el dato seleccionado es el buscado, e indicar relaciones con otros datos. En particular, los metadatos son de gran utilidad en recursos que no son textuales y en los que su contenido no puede ser indexado por sistemas automáticos, como por ejemplo, contenidos multimedia, de audio o vídeo, imágenes, etc.

Existen tres tipos de metadatos:

- Metadatos descriptivos: Utilizados para el descubrimiento, la identificación y luego selección de los objetos.
- Metadatos administrativos: Contienen información sobre cuándo y cómo fue creado el recurso, quien es el responsable del acceso o de la actualización del contenido e información técnica.
- Metadatos estructurales: Sirven para identificar las partes que componen al recurso y definir la estructura.

En la educación, las instituciones y organizaciones utilizan metadatos para describir los recursos y así mejorar la gestión, la búsqueda y la reutilización de los objetos de aprendizaje.

### 2.1.4 Estándares de metadatos

Según la RAE (Real Academia Española) [21], un estándar se define como “*adj. Que sirve como tipo, modelo, norma, patrón o referencia*”. En este caso, cuando hablamos de “estándar de metadatos”, se trata de un modelo que especifica cómo organizar los metadatos de un OA.

El propósito de los estándares de metadatos es facilitar la búsqueda, evaluación, adquisición y uso de los objetos de aprendizaje. Estas actividades son comúnmente realizadas por alumnos y docentes, pero debido al uso de estándares, también pueden ser llevadas a cabo por procesos automáticos. Además, los estándares permiten compartir OA con facilidad, así como el intercambio de los mismos. Esta interoperabilidad aumenta el valor de los repositorios de OA (ROA) ya que al colaborar entre sí, los ROA pueden mejorar los resultados de las consultas y así brindar un mejor servicio a los usuarios.

Existen diversos estándares, a continuación se describen los más utilizados.

#### 2.1.4.1 Learning Object Metadata (LOM)

En el ámbito de e-learning, se ha desarrollado el estándar IEEE 1484.12.1:2002, especificado por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE). LOM [16] especifica la sintaxis y la semántica de los atributos necesarios para describir los objetos de aprendizaje. Es un modelo de datos generalmente codificado en XML [22], utilizado para describir objetos de aprendizaje. Su principal propósito es facilitar la reutilización de OA y su interoperabilidad.

Los *elementos de datos* (metadatos que describen al OA) se agrupan en nueve categorías denominadas: *general, life cycle, meta-metadata, educational, technical, educational, rights, relation, annotation, y classification*. Estas nueve categorías poseen un total de 62 campos entre todas, donde algunos de éstos utilizan un vocabulario predefinido. De esos 62 campos, 15 son obligatorios, con el objetivo de estandarizar cualquier búsqueda a nivel internacional. El modelo de datos LOM se describe como una jerarquía de elementos de datos, los cuales pueden ser *agregados* o *simples*. Los *agregados* se componen de otros elementos y no tienen valores, mientras que los *simples* son las hojas en la jerarquía. Dichas hojas tienen valores

individuales cuyos tipos pueden ser “*LangString*”, “*CharacterString*”, “*Vocabulary*”, “*Duration*”, “*DateTime*” o “*Undefined*”.

En resumen, el esquema básico de LOM define para cada elemento un *nombre* de referencia, una *explicación* del mismo, un *tamaño* o número de valores permitidos, un *orden* de relevancia de los valores (sólo aplicable en el caso de elementos de datos de valores múltiples), un *ejemplo* ilustrativo, y en el caso de los elementos simples también se define un *espacio de valores* y un *tipo*.

#### 2.1.4.2 Learning Object Metadata (LOM-ES)

El estándar Learning Object Metadata Español [23] es la versión oficial de LOM que tiene el objetivo de contemplar y satisfacer las necesidades la comunidad educativa de España. Fue desarrollado en conjunto por el Ministerio de Educación, el Ministerio de Industria, Turismo y Comercio y todas las Comunidades Autónomas en relación a los Programas institucionales para el desarrollo de la Sociedad de la Información y el Conocimiento. En líneas generales, las principales diferencias respecto al estándar original LOM se encuentra la definición específica de una arquitectura modular de jerarquía creciente basada en modelos de agregación, donde cada nivel de agregación es independiente de los demás, pero se construye basándose en los anteriores. También cobran mucha importancia los metadatos de origen didáctico, dado su relación con los procesos de diseño, utilización, evaluación y rediseño. Además se agregan nuevos metadatos, términos a los vocabularios, se alteran las cardinalidades y se incrementan las restricciones de obligatoriedad.

#### 2.1.4.3 Dublin Core (DC)

Dublin Core [24] es un estándar de metadatos desarrollado por la Dublin Core Metadata Initiative (DCMI), una organización enfocada en promover la adopción de estándares de metadatos para facilitar la interoperabilidad y el desarrollo de vocabularios especializados para describir los recursos. DC es un sistema que define 15 elementos, los cuales son opcionales, pueden repetirse y no importa el orden. Estos metadatos se pueden clasificar en tres categorías, las cuales indican la clase de información que almacenan.

- Metadatos que describen el contenido del objeto: *title*, *subject*, *description*, *source*, *type*, *relation*, *coverage*.
- Metadatos que describen la propiedad intelectual del objeto: *creator*, *contributor*, *publisher*, *rights*.
- Metadatos que describen a la instancia específica del objeto: *date*, *format*, *identifier*, *language*.

En éste estándar los metadatos se describen mediante la clave “*name*”, la cual se expresa en lenguaje natural y con una única palabra que transmite el significado semántico de los elementos. DC no especifica los tipos de datos de los elementos (a diferencia de LOM) y por defecto se asumen todos como Texto.

#### 2.1.4.4 Sharable Content Object Reference Model (SCORM)

SCORM [25] es un conjunto de estándares y especificaciones que permite crear objetos pedagógicos estructurados. Es producido por Advanced Distributed Learning (ADL) [26], un grupo de investigación patrocinado por el Departamento de Defensa de Estados Unidos. SCORM enseña a los programadores a escribir su código de forma que pueda inter-operar con otras aplicaciones e-learning. Es el estándar de facto para la interoperabilidad de e-learning.

SCORM se compone de tres sub-especificaciones:

- **Content Packaging:** Especifica cómo se empaqueta y describe el contenido. Se basa principalmente en XML.
- **Run-Time:** Especifica cómo deberá ponerse en marcha el contenido y cómo se comunica con el Learning Management System (LMS). Se basa principalmente en ECMAScript (JavaScript). Un LMS es un software instalado en un servidor web que se emplea para administrar, distribuir y controlar las actividades de formación no presencial (o aprendizaje electrónico) de una institución u organización.
- **Sequencing:** Especifica cómo el alumno puede navegar entre las partes del curso. Se define por un conjunto de reglas y atributos escritos en XML.

SCORM establece que el contenido debe ser empaquetado en un directorio auto contenido o en un archivo ZIP. Este paquete se llama *Package Interchange File* (PIF). El PIF siempre debe contener un archivo de manifiesto XML en la raíz denominado *imsmanifest.xml*. El archivo manifiesto contiene toda la información necesaria para que el LMS pueda recuperar el contenido. El manifiesto divide el contenido en una o más partes llamadas SCO (Sharable Content Object). Cada SCO se puede combinar en una estructura de árbol que representa el contenido completo (un curso entero por ejemplo). Dicho árbol es conocido como “árbol de actividad” (o *activity tree* en inglés). El manifiesto contiene una representación XML del árbol de actividad y opcionalmente metadatos sobre el contenido y sus partes.

#### 2.1.5 Repositorios de Objetos de Aprendizaje

Los repositorios de objetos de aprendizaje [27] son considerados a menudo como una biblioteca digital. Permite a los usuarios (profesores, alumnos, etc.) compartir, administrar y utilizar recursos educativos. Es un contenedor de recursos que implementa un estándar de metadatos. Como se ha visto, la reutilización de los objetos de aprendizaje es una de las principales características de los OA, por lo que es indispensable contar con un lugar destinado al almacenamiento de estos objetos para facilitar su mantenimiento, localización y posiblemente el intercambio de OA entre sistemas. Según CANARIE (2001) [28] los ROA “*son un catálogo electrónico/digital que facilita las búsquedas en Internet de objetos digitales para el aprendizaje*”. Daniel (2004) [29], a partir de los términos “repositorio digital”, “objeto de aprendizaje” y “metadato” dice que “*los repositorios de objetos de aprendizaje son bases de datos con búsquedas que alojan recursos digitales y/o metadatos que pueden ser utilizados para el aprendizaje mediado*”. El JORUM+ Project (2004) [19] adopta la siguiente definición:

*“Un ROA es una colección de OA que tienen información (metadatos) detallada que es accesible vía Internet. Además de alojar los OA, los ROA pueden almacenar las ubicaciones de aquellos objetos almacenados en otros sitios, tanto en línea como en ubicaciones locales”.*

Generalmente se identifican dos tipos de ROA. Un tipo de ROA son los que contienen los OA y sus metadatos dentro de un mismo sistema. Mientras que el otro tipo son los que sólo contienen los metadatos y se accede al objeto a través de una referencia. Es posible la combinación de los dos tipos.

También se pueden clasificar según la organización de los metadatos. Se dice que un ROA es centralizado cuando los metadatos están contenidos en un mismo servidor, en cambio, se dice que un ROA es distribuido cuando los metadatos se encuentran distribuidos en grupos, en varios servidores y se comunican entre ellos para intercambiarlos.

A continuación se citan algunos ejemplos de repositorios.

#### 2.1.5.1 GLOBE

Global Learning Objects Brokering Exchange [30] (GLOBE) es una comunidad mundial que promueve el uso de estándares abiertos entre sus miembros. También provee una serie de servicios y herramientas on-line para el intercambio de recursos de aprendizaje. Utiliza la API Simple Query Interface (SQI) para interactuar con los repositorios (enviar y recibir consultas) y el estándar LOM para almacenar los metadatos de los OA.

#### 2.1.5.2 MERLOT

Es el repositorio más conocido y reconocido, que está marcando la pauta para el desarrollo y tendencia de los ROA. Es un repositorio centralizado que contiene los metadatos y las referencias a objetos ubicados en sitios remotos. Provee servicios como búsqueda, importación y exportación de objetos. Cualquier usuario puede acceder a estos objetos y solo los miembros pueden agregar objetos. MERLOT [31] utiliza el estándar SCORM para empaquetar los OA, la API SQI para las búsquedas, y tiene un modelo propio de metadatos, el cual se puede mapear a LOM. Este repositorio es parte de la Alianza GLOBE.

#### 2.1.5.3 ARIADNE

La fundación ARIADNE [32] es una Asociación Europea abierta sin fines de lucro, que busca compartir conocimiento y el reuso de recursos. Para lograr este cometido, la fundación creó una infraestructura basada en estándares que permite la publicación y el manejo de recursos digitales de aprendizaje. Utiliza SQI para realizar la búsqueda, y LOM para describir los metadatos de los OA, pero además es compatible con otros estándares como por ejemplo Dublin Core. La fundación ARIADNE también es miembro de la Alianza GLOBE.

#### 2.1.5.4 CAREO

CAREO [33] es un repositorio centralizado de objetos de aprendizaje de profesores de Alberta, Canadá. Cualquier usuario puede acceder a los objetos, y los miembros poseen servicios adicionales.

## 2.2 Moodle

Moodle [34] [35] [36] es una aplicación web que se utiliza como ambiente educativo virtual. Es un sistema de gestión de cursos que ayuda a los educadores a crear comunidades de aprendizaje en línea. Este tipo de plataformas también se conoce en inglés como LMS (Learning Management Systems). Estas herramientas son de gran utilidad en ambientes educativos ya que permiten a los profesores gestionar cursos virtuales para sus alumnos y dar apoyo a los cursos semipresenciales.

Actualmente en la Facultad de Ingeniería tiene disponible la versión 2.3.7 de Moodle y cuenta con alrededor de 430 cursos, algunos de los cuales se dictan de forma no presencial y están basados exclusivamente en la plataforma.

Como ya se mencionó, uno de los principales objetivos de este proyecto es integrar con Moodle un módulo de recomendación de objetos de aprendizaje, utilizando distintos tipos de algoritmos. Las ventajas de contar con el sistema de recomendación de objetos de aprendizaje en Moodle son diversas. La principal ventaja es que los estudiantes pueden acceder rápidamente a los recursos que son utilizados más frecuentemente por otros estudiantes, o recursos que pueden resultar de interés. Otra ventaja es que el sistema permite valorar los objetos de aprendizaje, lo cual implica obtener una retroalimentación para los docentes facilitándoles la tarea de determinar cuáles elementos son útiles y cuáles no. El sistema también puede recomendar objetos de aprendizaje a un cierto grupo de estudiantes, esto resulta interesante por ejemplo cuando se tienen los estudiantes de un determinado curso, el sistema recomendaría al docente qué recursos pueden resultar interesantes a los alumnos de su curso.

En Moodle, las distintas herramientas que se pueden incluir en un curso se dividen en dos grandes categorías: Actividades y Recursos. Una Actividad puede ser cualquiera de los siguientes elementos:

- *Base de datos*
- *Chat*
- *Consulta*
- *Cuestionario*
- *Encuestas predefinidas*
- *Foro*
- *Glosario*
- *Herramienta Externa*

- *Lección*
- *Paquete SCORM*
- *Taller*
- *Tarea*
- *Wiki*

Mientras que un Recurso puede ser alguno de los siguientes:

- *Archivo*
- *Carpeta*
- *Etiqueta*
- *Libro*
- *Página*
- *Paquete de contenido IMS*
- *URL*

A continuación se detalla el Recurso Archivo debido a que fue seleccionado como el OA a recomendar por el sistema ya que se trata de la herramienta más utilizada en los ambientes de Moodle. El resto de las Actividades y Recursos se detallan en el anexo, sección *1. Descripción de Actividades y Recursos que se pueden incluir en Moodle*.

El Recurso Archivo permite a los profesores proveer un archivo como un recurso del curso. Cuando sea posible, el archivo se mostrará dentro de la interfaz gráfica del curso, si no es el caso, se les preguntará a los estudiantes si quieren descargarlo. El recurso Archivo puede incluir archivos de soporte, por ejemplo, una página HTML puede tener incrustadas imágenes u objetos Flash. Un Archivo puede utilizarse para:

- Compartir presentaciones utilizadas en clase
- Incluir una mini-web como recurso del curso
- Proveer a los estudiantes de borradores de archivos para que los editen y los envíen en sus tareas

Moodle permite agrupar Actividades o Recursos. Por ejemplo se podría dividir un curso en semanas, donde cada semana es un conjunto de Actividades y Recursos. También podría dividirse por temas. Para realizar las recomendaciones es necesario definir el nivel de granularidad con el cuál se harán, es decir, si se recomendará Actividades y Recursos individualmente, o se recomendará grupos de Actividades y Recursos, como puede ser el grupo de Actividades y Recursos que pertenezcan a cierta temática.

Otra herramienta que se puede incluir en los cursos de Moodle son los Bloques. Los bloques son ítems que pueden añadirse a la columna izquierda, derecha o central de cualquier página de Moodle. Existen varios tipos de bloques, por ejemplo bloque de últimas noticias, donde se muestran las publicaciones más recientes del foro de un curso. Otro ejemplo de bloque es el bloque calendario, donde se muestra un calendario en el lugar que se desee.

Cada curso puede tener un número indeterminado de alumnos y un número indeterminado de grupos que acoja a esos alumnos. Los alumnos tienen un perfil de usuario que pueden editar y opciones específicas por curso en el que están matriculados.

### 2.2.1 Integración de módulo de recomendaciones con Moodle

A continuación se realiza un análisis comparativo de las distintas alternativas para realizar la integración del módulo de recomendaciones con Moodle.

#### 2.2.1.1 Módulo Moodle

Moodle significa Modular Object-Oriented Dynamic Learning Environment (Entorno de enseñanza dinámico orientado a objetos modular). Desde la perspectiva del programador, la M de Moodle contiene un concepto muy importante, ya que Moodle, en su totalidad, está diseñado de forma modular. Esto permite al desarrollador realizar importantes modificaciones a la plataforma sin necesidad de cambiar el código fuente. La modularidad se convierte en una importante característica que permite reducir la cantidad de tiempo empleada en realizar modificaciones cuando se lanzan nuevas versiones de Moodle. Los principales tres tipos de módulos que se pueden desarrollar para Moodle son actividades, recursos y bloques. Por lo tanto una manera posible de brindar las recomendaciones al usuario es desarrollando un nuevo módulo para Moodle que realice esta tarea. La desventaja de esta alternativa es que para instalar o integrar el módulo con Moodle es necesario contar con privilegios de Administrador en la plataforma. Previo al desarrollo de un módulo es necesario conocer el esquema del mismo. Los módulos se almacenan en la carpeta moodle/mod, cada módulo se almacena en un directorio distinto con una estructura general de archivos y directorios que se resume a continuación:

- *mod\_form.php*: Formulario para crear o modificar una instancia de la actividad.
- *version.php*: para definir meta información, como por ejemplo la versión del módulo.
- *lang/*: directorio para almacenar los archivos de idioma del módulo. El módulo debe tener archivos de idioma que contenga las cadenas para ese módulo. Deberán ser al menos en inglés y traducidos a los idiomas de los usuarios finales que utilicen la actividad.
- *db/*: Directorio donde se almacenarán los ficheros con las tablas de las bases de datos necesarias para la actividad.
- *access.php*: Fichero de permisos del módulo. Los permisos no son obligatorios pero sí muy recomendables para garantizar qué usuarios pueden acceder a las distintas partes del módulo.
- *install.xml*: Fichero que describe la estructura de las tablas del módulo.
- *upgrade.php*: código de actualización, aquí es donde se deben de hacer las alteraciones de las tablas, si las hay, entre versiones.
- *index.php*: Este fichero sirve para mostrar todas las instancias de una actividad en un curso, es decir, una lista con todas las instancias del mismo módulo.
- *view.php*: Esta es la página que muestra una instancia de la actividad.

- *lib.php*: Librería de funciones del módulo. En este fichero se implementarán todas las funciones y procedimientos del módulo. Si el módulo se llama *ejemplo*, entonces las funciones mínimas y obligatorias que ha de tener la actividad tienen que ser de la forma:
  - *ejemplo\_install()*: Acciones a realizar al instalar el módulo.
  - *ejemplo\_add\_instance()*: código para añadir una nueva instancia.
  - *ejemplo\_update\_instance()*: función para actualizar una instancia existente.
  - *ejemplo\_delete\_instance()*: código para borrar una instancia.
  - *ejemplo\_user\_outline()*: da un resumen concreto de la actividad de un usuario.
  - *ejemplo\_user\_complete()*: devuelve un informe más detallado de la contribución de un usuario.
  - *ejemplo\_get\_view\_actions()*: Clasifica las acciones para el log. Se usa en el informe de participación.
  - Todas las funciones, procedimientos y constantes, creados en *lib.php*, tienen que comenzar con el nombre del módulo.
- *settings.php* (opcional): Formulario con las opciones generales del módulo.

En cuanto al idioma, como Moodle es una plataforma internacional, cada módulo ha de tener un archivo de idioma de forma que los textos que formen parte de la interfaz (por ejemplo: títulos de las páginas, nombre de los botones, etc.), se extraigan de un conjunto de archivos de idioma.

#### 2.2.1.2 Bloque HTML

Por otro lado, en Moodle existen bloques predefinidos como pueden ser calendarios, bloques con contenido HTML, canales RSS remotos y muchos otros más. El bloque HTML es otra posible alternativa, ya que en este bloque se podría embeber una página web donde se encontrarían las recomendaciones para el usuario. Para lograr esto se debe armar un link a determinada página web pasándole el identificador del usuario autenticado a Moodle para así poder realizar las recomendaciones personalizadas. El identificador del usuario se obtiene de la propia página del Moodle procesando la misma mediante código JavaScript.

#### 2.2.1.3 Recurso URL

Una tercera alternativa es utilizando el Recurso URL. El recurso URL permite que el docente pueda proporcionar un enlace de Internet como un recurso del curso. Todo aquello que esté disponible en línea, como documentos o imágenes, puede ser vinculado. La URL no tiene por qué ser la página principal de un sitio web. Moodle permite pasar parámetros a través de la URL como pueden ser el identificador del usuario, identificador del curso, etc. La desventaja de esta alternativa es que sólo es visible el link de acceso a la página y no la página embebida en sí, por lo que es necesario ingresar al link para poder observar las recomendaciones.

#### 2.2.1.4 Herramienta externa

Otra alternativa es la de utilizar el recurso Herramienta externa. La desventaja de utilizar este recurso es que a la hora de presentar las recomendaciones, no es posible que se vean

directamente en la página principal de Moodle, sino que se presenta un link para acceder a la herramienta externa, esto quita practicidad a la hora de acceder a las recomendaciones rápidamente y de forma más transparente para el usuario.

### 2.2.2 Extracción de metadatos de OA

Como ya se mencionó anteriormente, los Objetos de Aprendizaje están conformados por una entidad de carácter digital, más un conjunto de metadatos que pueden cumplir con algún estándar. La extracción de metadatos de los Objetos de Aprendizaje [37] [38] es necesaria ya que estos son almacenados en un repositorio y utilizados por algoritmos de recomendación o por motores de búsquedas avanzados.

Existen estudios realizados por distintos investigadores para realizar la extracción automática de Objetos de Aprendizaje. Uno de ellos es un prototipo denominado LookIng4LO [39]. Este prototipo se desarrolló en la Facultad de Ingeniería, UdelaR. A partir de documentos no estructurados, como pueden ser pdfs, textos, HTML y paquetes SCORM extrae información según un área temática y un conjunto de componentes pedagógicos (definiciones, ejemplos, ejercicios, etc.), empaquetándola en Objetos Digitales de Aprendizaje (ODAs). La temática se define a través de una ontología y los componentes pedagógicos son modelados con reglas para definir patrones de búsqueda. Además, el sistema genera metadatos que describen el contenido extraído y el origen de dicha información. Los ODAs generados son empaquetados utilizando el estándar SCORM. Otro sistema dedicado a la extracción automática de metadatos es SAXEF (System for Automatic eXtraction of E-learning object Features) [40], el cual extrae automáticamente indicadores didácticos de cualquier página Web. El sistema SAXEF ha sido pensado para la extracción de atributos texto/multimedia desde cada página Web, considerada como un objeto de aprendizaje, o un grupo de páginas Web, que representan un curso completo. En la práctica, dado un curso o simplemente un objeto de aprendizaje.

Otra solución para la extracción de metadatos es el sistema TWYS5 [41], desarrollado por Tang Way Yuen, dentro del departamento de Ciencias de la Computación en la Ciudad Universitaria de Hong Kong. TWYS permite extraer metadatos de objetos de aprendizajes desde páginas HTML. Se basa en el estándar IEEE LOM. Con TWYS se puede adquirir fácilmente recursos de aprendizaje relevantes haciendo que los motores de búsqueda adopten y soporten el estándar IEEE LOM por medio de la extracción automática desde páginas webs.

## 2.3 Técnicas de recomendación

En esta sección se describen las técnicas de recomendación [42]. Para entenderlas mejor, es necesario definir los conceptos de Crowdsourcing y Friendsourcing

### 2.3.1 Crowdsourcing

El término proviene del inglés *Crowd* (multitud) y *Outsourcing* (tercerización) [14], el cual se podría traducir como *colaboración abierta distribuida*, la cual consiste en externalizar tareas que

tradicionalmente realizaba un empleado a un numeroso grupo de personas a través de una convocatoria abierta, donde cada individuo realiza pequeñas contribuciones agregando gran valor a nivel global. Uno de los ejemplos más conocidos es la enciclopedia en línea conocida como Wikipedia. Se trata de un sitio web en el cual los usuarios pueden crear y editar documentos, generando una gran fuente de conocimiento.

Estellés y González (2012) [43], tras estudiar más de 40 definiciones de Crowdsourcing, proponen una nueva definición integradora: *"El Crowdsourcing es un tipo de actividad en línea participativa en la que una persona, institución, organización sin ánimo de lucro o empresa, propone a un grupo de individuos mediante una convocatoria abierta flexible la realización libre y voluntaria de una tarea. La realización de la tarea, de complejidad y modularidad variable, y en la que la multitud debe participar aportando su trabajo, dinero, conocimiento y/o experiencia, siempre implica un beneficio mutuo. El usuario recibirá la satisfacción de una necesidad concreta, ya sea esta económica, de reconocimiento social, de autoestima, o de desarrollo de aptitudes personales, mientras que el crowdsourcer obtendrá y utilizará en su beneficio la aportación del usuario, cuya forma dependerá del tipo de actividad realizada."*

### 2.3.2 Friendsourcing

Friendsourcing [8] es una forma de Crowdsourcing, pero con la diferencia de que el conocimiento se obtiene a partir de un confiable y eventualmente reducido grupo de personas que están conectadas socialmente en una red. En Friendsourcing la información no se obtiene a través de la multitud sino a través de un grupo de personas cuidadosamente seleccionado, lo cual se convierte en una fuente de información de alta calidad. Un ejemplo de Friendsourcing es la red social Facebook, que brinda a los usuarios recomendaciones para unirse a ciertos grupos o participar en eventos. Una desventaja de Friendsourcing es que cuando se tienen pocas relaciones sociales o cuando los amigos no han hecho muchas evaluaciones de recursos, las recomendaciones pueden no ser relevantes para el usuario.

Los sistemas de recomendación se basan en (1) información propia o base, que existe en el sistema antes de comenzar el proceso de recomendación, (2) datos de entrada (input) ingresados por el usuario objetivo para que el sistema genere la recomendación y (3) el algoritmo que combina los datos base del sistema con los datos de entrada para generar las recomendaciones. Existen diversas clasificaciones de las técnicas de recomendación que son utilizadas en los sistemas de recomendaciones. A continuación se detallan las cinco más importantes.

### 2.3.3 Filtrado colaborativo

El objetivo de un algoritmo de filtrado colaborativo [7] es sugerir nuevos objetos o predecir la utilidad de cierto objeto para un usuario en particular basado en sus intereses y opiniones de otros usuarios con intereses similares. Esta técnica entra en la categoría de los algoritmos que se basan en Crowdsourcing. Investigadores han desarrollado una gran cantidad de algoritmos basados en filtrado colaborativo y los han clasificado en dos categorías: basados en memoria y basados en el modelo.

Los algoritmos **basados en memoria** utilizan toda la información que se tiene de los usuarios y objetos para generar la predicción. Estos sistemas utilizan técnicas estáticas para encontrar el conjunto de usuarios, conocidos como *vecinos*, que tienen intereses similares al usuario al cual se le quiere realizar la predicción, por ejemplo usuarios que han clasificado o puntuado ciertos objetos de la misma forma que el usuario objetivo. Una vez encontrados estos vecinos, se utiliza un algoritmo que combina la preferencia de los usuarios para generar una predicción o un conjunto de predicciones (top N) para el usuario objetivo. Estas técnicas conocidas como *vecino más cercano* o *basadas en el usuario* son las más populares y más usadas en la práctica.

Los algoritmos **basados en el modelo** recomiendan objetos a través de la construcción de un modelo de las puntuaciones de los usuarios. Los modelos se desarrollan a través de un proceso que intenta descubrir patrones en grandes volúmenes de conjuntos de datos, por ejemplos algoritmos de aprendizaje automático. Estos se utilizan para realizar predicciones de datos reales. Existe una gran variedad de algoritmos de filtrado colaborativo basado en modelos, algunos son: redes bayesianas, modelos de clustering, modelos de semántica latente y otros. Estos algoritmos tienen varias ventajas, por ejemplo manejan los datos dispersos mejor que los basados en memoria lo que permite mayor escalabilidad con grandes conjuntos de datos. También mejora el rendimiento de la predicción y da un fundamento intuitivo para las recomendaciones. Las desventajas de este método es el costo de la construcción del modelo. Se debe tener un compromiso entre el rendimiento de la predicción y la escalabilidad. Se puede perder información útil debido a la reducción del modelo.

A continuación se describen distintos algoritmos de filtrado colaborativo.

#### 2.3.3.1 Basados en contenido

Un Sistema de Recomendación basado en contenido permite sugerir al usuario nuevos ítems en función de su similitud con el contenido o descripción de otros ítems que éste ha juzgado anteriormente, lo que se denomina comúnmente: correlación ítem a ítem. Los ítems de interés son definidos por sus características, como por ejemplo las palabras en un documento de texto. Para determinar los perfiles se utilizan métodos de aprendizaje, como por ejemplo árboles de decisión o redes neuronales. Los perfiles son modelos a largo plazo y se actualizan a medida que se observan más preferencias del usuario. En los sistemas basados en contenido los datos base son las características de los ítems, los datos de entrada son las puntuaciones de ítems hechas por el usuario al que se le quiere brindar las recomendaciones, y el algoritmo de recomendación consiste en generar el perfil del usuario que se ajuste al comportamiento en las puntuaciones de ítems y luego se utiliza dicho perfil para determinar las preferencias. Estos sistemas tienen el problema de que necesitan muchas puntuaciones de ítems para poder generar perfiles confiables. Otra desventaja es que están limitados por las características de los ítems que recomienda.

### 2.3.3.2 Basados en la demografía de los usuarios

En los sistemas de recomendaciones demográficos los usuarios se categorizan por atributos personales y se realizan las recomendaciones de ítems en base a clases demográficas. Se utilizan las puntuaciones de ítems realizadas por los usuarios e información demográfica. El objetivo del algoritmo es identificar aquellos usuarios que son demográficamente similares al usuario objetivo y se extrapolan las preferencias a partir de las puntuaciones de ítems. Una ventaja de esta técnica es que no se requiere de un historial de puntuaciones del usuario como en las técnicas de filtrado colaborativo y en las basadas en contenido.

### 2.3.3.3 Basados en la utilidad

En los sistemas de recomendaciones basados en utilidad se realizan recomendaciones de ítems basadas en la computación de la utilidad de los ítems para el usuario. La mayor dificultad de esta técnica es la definición de la función de utilidad para cada usuario. Al algoritmo se le brinda la función de utilidad sobre los ítems que describe la preferencia del usuario al que se le quiere brindar recomendaciones, y en base a las características de los ítems, se recomiendan aquellos con mayor utilidad para el usuario objetivo. La función de utilidad puede ser definida por el administrador del sistema, ya que debe poseer conocimientos acerca de los intereses de los usuarios. Una ventaja de esta técnica es que no necesita de puntuaciones de ítems para generar las recomendaciones. Una desventaja que presentan estos sistemas es que no son flexibles en dominios complejos y subjetivos. Otra desventaja es que la habilidad para recomendar es estática, ya que no tienen la capacidad de aprender.

### 2.3.3.4 Basados en el conocimiento

En los sistemas de recomendaciones basados en conocimiento se realizan recomendaciones de ítems a través de inferencias acerca de las necesidades y preferencias del usuario. En estos sistemas hay tres tipos de conocimiento:

- Conocimiento del catálogo, que es el conocimiento de los ítems que son recomendados y sus características.
- Conocimiento funcional que es el conocimiento para mapear las necesidades del usuario y el ítem que podría satisfacer estas necesidades.
- Conocimiento del usuario, que es el conocimiento que el sistema debe tener sobre el usuario para poder brindar buenas recomendaciones.

Los datos base son las características de los ítems y el conocimiento de cómo estos ítems satisfacen las necesidades de los usuarios. Los datos de entrada son las descripciones de las necesidades o intereses del usuario al que se le quiere brindar las recomendaciones, y el algoritmo de recomendación consiste en inferir aquellos ítems que satisfagan las necesidades o sean de interés para el usuario objetivo. Una ventaja de estos sistemas es que son apropiados para exploración casual. Una desventaja que presentan estos sistemas es que necesitan adquirir el conocimiento.

En la Tabla 1 se resumen las técnicas especificadas anteriormente, donde se asume que  $I$  es el conjunto de ítems a recomendar,  $U$  es el conjunto de usuarios con preferencias conocidas por el sistema,  $u$  es el usuario al que se le realiza la recomendación, e  $i$  es el ítem al cual se quiere predecir la puntuación.

Técnica	Datos propios	Datos de entrada	Proceso
Filtrado Colaborativo	Puntaje de los usuarios de $U$ sobre los ítems de $I$ .	Puntaje del usuario $u$ sobre los ítems de $I$ .	Identificar a los usuarios en $U$ similares a $u$ , y extrapolar con sus puntajes de $i$ .
Contenido	Características de los ítems de $I$ .	Puntaje del usuario $u$ sobre los ítems de $I$ .	Generar un perfil que modele el comportamiento de puntuación del usuario $u$ , y aplicarlo sobre ítem $i$ .
Demográficos	Información demográfica de $U$ y los puntajes de los ítems de $I$ .	Información demográfica de $u$ .	Identificar a los usuarios que son demográficamente similares a $u$ , y extrapolar con sus puntajes de $i$ .
Utilidad	Características de los ítems de $I$ .	Función de utilidad que aplicada a los ítems de $I$ describe las preferencias de $u$ .	Aplicar la función sobre los ítems y determinar el puntaje de $i$ .
Conocimiento	Características de los ítems de $I$ . Conocimiento sobre cómo los ítems satisfacen las necesidades de los usuarios.	Descripción de las necesidades e intereses de $u$ .	Inferir un correspondencia entre $i$ y las necesidades de $u$ .

Tabla 1 - Resumen de técnicas de recomendación

### 2.3.3.5 Comparaciones entre las técnicas de recomendación

Las técnicas mencionadas anteriormente tienen cierta madurez en el área de los sistemas de recomendación, y por lo tanto ya son conocidos los principales problemas y beneficios que éstas conllevan. A continuación se realiza una breve comparación de las diferentes técnicas [42].

Los sistemas de recomendación basados en *filtrado colaborativo* no dan buenos resultados en casos de:

- Usuarios nuevos: Para la recomendación se realiza una comparación entre el usuario objetivo y los demás usuarios basándose únicamente en los puntajes otorgados. Un usuario con pocos puntajes resulta difícil de categorizar.

- Ítems nuevos: Al igual que con los nuevos usuarios, un nuevo elemento con pocos puntajes resulta difícil de clasificar. Este problema se presenta en casos de recomendación de noticias, donde se presenta un flujo constante de artículos y los usuarios solamente puntúan unos pocos.
- O cuando el espacio de puntajes es disperso, es decir, cuando sólo unos pocos usuarios han puntuado los mismos elementos. Si bien es posible disminuir la dispersión utilizando FC basado en el Modelo, éste continúa siendo un problema en el filtrado de noticias donde, como se mencionó anteriormente, la cantidad de ítems es muy grande y a menos que la información de los usuarios también lo sea, la probabilidad de que los usuarios compartan una gran cantidad de elementos puntuados es baja.

Estos problemas sugieren que las técnicas de filtrado colaborativo son más eficientes en casos de que la densidad de interés de los usuarios sea relativamente alta respecto a un pequeño (y constante) conjunto de ítems. El uso de FC es más recomendado para casos en que los usuarios compartan los gustos entre sí. Esta técnica no funciona muy bien cuando el usuario objetivo tiene gustos particulares, diferentes a los demás usuarios del sistema. Este problema, llamado “Gray-Sheep” [44], también se presenta en sistemas de recomendación basados en datos demográficos cuando intentan clasificar a los usuarios de acuerdo a sus características personales. Por otro lado, los sistemas demográficos no presenta el problema de los usuarios nuevos ya que no requieren los puntajes de dicho usuario. Sin embargo, si se presenta el problema de recolectar la información demográfica lo cual afecta el crecimiento de este tipo de sistemas.

Las técnicas *basadas en contenido* tienen el principal problema de recolectar una cantidad suficiente de puntajes del usuario objetivo para construir un perfil confiable. Al igual que en FC, las técnicas basadas en contenido tienen el problema de estar limitadas por las características que están explícitamente relacionadas con los objetos que recomiendan. Por ejemplo, cuando se recomienda música, sólo se puede utilizar la información adicional sobre la canción como nombre, álbum, año, intérprete(s), etc. ya que la canción en sí no es analizada por el sistema. Esto deja a las técnicas a la merced de la información adicional que el sistema maneje. La gran ventaja del FC sobre las técnicas basadas en contenido es la posibilidad de asociar géneros, o también conocido como recomendaciones “*outside the box*”. Es posible que a personas fanáticas de las películas de suspenso, también les gusten las de terror, pero un sistema de recomendación que utilice técnicas basadas en contenido entrenado en las características del cine de suspenso (actores, director, trama, sinopsis, etc.) no será capaz de recomendar una película de terror ya que ninguno de los datos adicionales tienen por qué coincidir.

Las técnicas de recomendación *basadas en utilidad* y *basadas en conocimiento* carecen de problemas de dispersión, ya que no basan sus recomendaciones en análisis de estadísticas. Las técnicas basadas en utilidad requieren de una función (llamada *función de utilidad*) que procese todas las características de los ítems a recomendar. El principal beneficio de esta técnica es que permite incorporar diferentes factores que contribuyen a la valoración del elemento, más allá de sus características específicas. La flexibilidad que permiten estos sistemas es también un problema. El usuario debe construir la función de preferencia y valorar

la relevancia de las características de los elementos. Esto puede ser factible para ítems con pocas cualidades, pero no para elementos más complejos como contenidos multimedia o noticias. También se debe tener en cuenta el perfil de los usuarios, ya que para usuarios expertos puede resultar una ventaja, para el usuario promedio puede ser más inapropiado.

En el anexo, bajo el título 2. *Proceso de Filtrado Colaborativo*, se presentan con más detalle los conceptos comentados anteriormente.

### 2.3.4 Recomendaciones grupales

En los últimos años se ha presenciado un crecimiento exponencial en las actividades de los usuarios en internet, el cual se refleja en el uso de las redes sociales tales como Facebook, Twitter y otros sitios que promueven los vínculos entre usuarios, y el intercambio de contenidos (fotos, música, etc.). Si bien ha habido un avance en el desarrollo de técnicas de recomendación de información para usuarios individuales basadas en filtrado colaborativo y Friendsourcing (entre otras), aún no se ha realizado mucho en el área de las recomendaciones para todo un grupo de usuarios relacionados. Este problema se conoce como Recomendaciones Grupales. Esta sección se basa en [9] [45] [46]. Un grupo de usuarios puede ser un grupo de amigos que van al cine, compañeros de trabajo que van a almorzar juntos, o un grupo de personas que (sin conocerse) realizan un viaje en excursión.

La dificultad yace en que un ítem puede tener diferente relevancia para los distintos usuarios dentro de un grupo. Por ejemplo, dado un grupo de amigos, no todos pueden opinar de la misma forma respecto a la elección de una película, o un grupo de compañeros de trabajo puede estar en desacuerdo sobre donde ir a comer.

El objetivo de las recomendaciones grupales es determinar un puntaje grupal para cada ítem que refleje el interés y preferencia de todos los miembros del grupo. En general, los miembros de un grupo no tienen la misma preferencia para cierto ítem, por lo que se debe diseñar una forma inteligente de calcular la puntuación de cada ítem en base a la preferencia de cada miembro del grupo. Se han desarrollado una gran cantidad de estrategias de decisión grupal en los últimos años. Estas estrategias buscan comprender cómo los miembros de un grupo llegan a un consenso, es decir, dado un conjunto de preferencias individuales para un cierto ítem, cuál es la preferencia grupal para ese ítem.

Hay dos aspectos principales para determinar el puntaje grupal de cierto ítem. Primero, el puntaje grupal debe determinar el grado de *concordancia* que hay sobre un ítem, cuanto más miembros prefieran un ítem, mayor deberá ser el puntaje de concordancia. A su vez, el puntaje grupal debe reflejar el grado de *desacuerdo* que hay entre los miembros del grupo sobre cada ítem. El primer aspecto se denomina *group relevance* (relevancia grupal), el segundo aspecto se denomina *group disagreement* (discrepancia grupal). En el anexo 3. *Recomendaciones grupales* se detallan estos aspectos.



## 3 Análisis

Actualmente la Facultad de Ingeniería cuenta con la plataforma de aprendizaje Moodle. La misma posee alrededor de 400 cursos, lo que implica que miles de estudiantes acceden constantemente al sistema para realizar diversas actividades. El hecho de que exista una gran cantidad de cursos, conlleva a que el número de OA existentes en la plataforma sea enorme, lo cual puede perjudicar el fácil y rápido acceso al material deseado por parte de estudiantes y docentes.

En el marco de este proyecto se presenta la tarea de realizar un sistema que facilite el acceso a los recursos educativos publicados en el curso de Fundamentos de la Web Semántica de la Facultad de Ingeniería que se encuentra en la plataforma Moodle en su versión 2.3.7.

En esta sección se realiza el análisis del problema a resolver estudiando distintas alternativas para la solución y evaluando las ventajas y desventajas de cada una. Específicamente se tratan los temas de extracción y recomendación de Objetos de Aprendizaje en Moodle.

### 3.1 Extracción de Objetos de Aprendizaje en Moodle

La extracción de OA del Moodle se refiere a la tarea de recuperar y almacenar en una base de datos la referencia al objeto en Moodle junto con la mayor cantidad de metadatos disponibles.

Se evaluaron dos alternativas para llevar a cabo el proceso de extracción de OA.

Una forma es extraer la información de las páginas HTML mediante un Web Scraper. Web Scraping [47] es una técnica de computación que consiste en extraer información desde sitios web mediante herramientas de software. Usualmente estos programas simulan la navegación que haría un humano sobre un sitio web a través de un navegador, utilizando el protocolo HTTP. El Web Scraping es una técnica universal utilizada por la mayoría de los motores de búsqueda y se basa en la indexación de la web, extrayendo y agrupando la información mediante el uso de un *bot* (*Bot: Programa informático que realiza funciones muy diversas, imitando el comportamiento humano*) [48]. El Web Scraping se enfoca en la estructuración de información de la web, como el texto HTML, para un posterior análisis. Esta técnica se utiliza frecuentemente para hacer copias de sitios enteros, descargando la información en un equipo local y así poder utilizarlos sin la necesidad de conexión a internet. También puede ser utilizada para la comparación de precios entre diferentes tiendas, o para la integración automática de datos en diferentes sitios web.

Otra alternativa es extraer la información directamente desde el Moodle. Para ello, se debe contar con acceso de lectura sobre el sistema de archivos (directorio *moodledata* dentro del servidor) y sobre la base de datos del sistema, donde se persiste toda la información del mismo.

### 3.1.1 Alternativas para la extracción

A continuación se detalla el estudio comparativo realizado entre las dos alternativas y al final una tabla que resume las ventajas y desventajas de cada una.

#### 3.1.1.1 Extracción web

La extracción desde la web posee muchas ventajas. Quizás la más importante es que no depende de cómo Moodle mantiene los OA, es decir, no es necesario invertir recursos en profundizar en el funcionamiento interno de la plataforma, en cómo maneja los archivos u objetos de aprendizaje, que estructuras utiliza, ni como se relacionan. Realizando la extracción desde la web sólo es necesario acceder a los OA desde la interfaz gráfica, como cualquier usuario desde un navegador, y luego mediante una tarea automática se extrae la información requerida de los objetos.

Otra de las principales ventajas es que no es necesario contar con permisos sobre el servidor de aplicaciones (*donde por servidor de aplicaciones se entiende como el dispositivo de software que gestiona la mayor parte de las funciones de la lógica de negocio y de acceso a los datos de la aplicación*) ni sobre la base de datos, ya que la información se extrae a nivel de capa de presentación (*donde por capa de presentación o interfaz gráfica, se entiende como la porción del sistema que comunica y presenta la información al usuario*), para lo cual sólo es necesario tener permisos mínimos, como acceso de lectura sobre los cursos. Incluso esto último a veces no es necesario, ya que algunos cursos permiten acceso libre como usuario invitado. Sin embargo, este método requiere más procesamiento de datos para la obtención de los objetos de aprendizaje, esto es, se deben procesar todas las páginas que se pueden acceder desde un curso en busca de OA. Este procesamiento implica procesar código HTML, con mucha información innecesaria, como por ejemplo: etiquetas, referencias, JavaScript, etc. Si bien esto puede verse como una desventaja, en realidad los potenciales problemas de baja performance no afectan mucho al usuario final ya que los procesos de extracción son procesos que no se ejecutan muy a menudo. Otra desventaja es que los programas de extracción dependen de la estructura de la capa de presentación del Moodle, por lo tanto, si ésta cambia en una nueva versión o actualización de funcionalidades, es probable que afecte el funcionamiento de los programas y haya que modificarlos.

#### 3.1.1.2 Extracción directa

La principal ventaja de la extracción directa contando con acceso al servidor y a la base de datos es que se tiene acceso a toda la información existente del Moodle. No sólo que se tiene acceso a todo, sino que la extracción no depende del ancho de banda disponible en la conexión a internet, o de la sobrecarga de información que tiene el lenguaje HTML, donde se debe procesar información innecesaria, tales como imágenes o scripts, propios de la capa de presentación. Además, la capa de persistencia (*donde por capa de persistencia se entiende como la porción del sistema donde residen los datos y es la encargada de acceder a los mismos*) generalmente posee mayor estabilidad en comparación con la capa de presentación, lo cual permite tener programas de extracción más durables en el tiempo. Sin embargo, el proceso de extracción puede llegar a ser incluso más complejo que desde la web, ya que esto

implica tener conocimiento de todas estructuras que utiliza Moodle para el almacenamiento de información, y cómo están relacionadas entre sí. Entender el funcionamiento interno, es decir, entender la capa de negocios de Moodle puede llegar a ser muy complejo. Se entiende por capa o lógica de negocio como la porción del sistema donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. No sólo la complejidad es una gran desventaja de este método, sino que el acceso a la información no es trivial, se requieren permisos a nivel de capa de persistencia para poder acceder. Esta es una desventaja muy importante de esta estrategia, ya que genera una dependencia muy fuerte con el Moodle, dado que si se quisiera realizar una extracción sobre una plataforma de Moodle en particular (p.e. de otra Universidad en otro país), se debería solicitar al administrador de ese sistema para que otorgue los permisos pertinentes. El bajo acoplamiento al sistema Moodle es una de las características principales que se desea en el marco de este proyecto, dado que si bien se desarrolla un sistema para una versión específica de la plataforma, es deseable poder extenderlo a futuro. El Moodle provee un conjunto de operaciones a través de una API (*Application Programming Interface*), que permite el acceso a la estructura de datos, que si bien puede facilitar la extracción de información, el uso de la misma no es trivial e insume tiempo de aprendizaje.

En la Tabla 2 se comparan las dos alternativas para la extracción de OA analizadas anteriormente.

Estudiando las ventajas y desventajas que presentan estas dos alternativas, se llega a la conclusión de que el método de extracción información desde la web es el más conveniente para este proyecto. Si bien la principal desventaja es la posible necesidad de una reescritura del proceso en caso de alteraciones en la capa de presentación, esta contrariedad también está presente en el caso de la extracción accediendo a la base de datos. Analizando el historial de cambios de las distintas versiones del sistema Moodle [49], se encuentran pocos cambios en la capa de presentación que afecten el proceso de extracción, mientras que se pueden apreciar varias modificaciones en las capas de persistencia y negocio. A modo de ejemplo se destaca un gran cambio respecto a la persistencia de los recursos (archivos) desde la versión 1.x a la 2.x. En las versiones anteriores, se mantenía en la base de datos la ubicación de los documentos en el sistema de archivos, mientras en la actual (2.0 en adelante) los archivos se almacenan en un directorio común, compartido por todo el sistema con el fin de optimizar el reúso de los OA. Dicho almacenamiento se realiza en el directorio *moodledata/filedir*, con una estructura de directorios basados en los identificadores numéricos de los elementos, resultando complejo para el entendimiento por parte de personas. Otro detalle no menor a tener en cuenta es el de los permisos requeridos para realizar la extracción. Si bien desde la web se requieren permisos mínimos, existen cursos que exigen una contraseña (seleccionada por los docentes) para acceder a la información. Es por lo tanto un prerrequisito para la extracción de OA que los cursos permitan acceso de invitados, o que los docentes proporcionen dicha clave.

Desde la WEB	Desde el sistema Moodle
Se requieren menos privilegios para ejecutar el proceso de extracción.	Se debe contar con acceso tanto al servidor como a la base de datos
Cierta complejidad en la estructura de las páginas HTML.	Estructura de almacenamiento de OA compleja (se almacenan en el sistema de archivos y se guardan las referencias en la BD).
Cantidad de metadatos limitada (por usuario de Moodle cuando ingresa el elemento).	Con tiempo suficiente, se puede obtener toda la información existente (metadatos).
Menor estabilidad en capas superiores puede afectar funcionamiento del proceso de extracción.	Mayor estabilidad en capas inferiores permite durabilidad en el proceso de extracción.
Extracción desde la web menos eficiente en cuanto a la performance, hay más overhead, información inútil en código HTML, etc.	Extracción directa, es más rápido. Sólo se extrae lo que se quiere.
Mayor complejidad y procesamiento para determinar si hubo una actualización de OA en el Moodle.	Más simple de saber si un curso fue actualizado con algún archivo u OA, mediante consultas SQL, es más directo.
En cursos que necesitan matrícula para acceder no es posible la extracción. Es necesario tener un usuario especial matriculado en todos los cursos para poder acceder con el mismo.	No se tiene este problema ya que al contar con acceso a la base y sistema de archivos, se puede acceder a toda la información.
No es necesario conocer cómo Moodle almacena los objetos de aprendizaje.	Es necesario adquirir conocimientos del funcionamiento de Moodle respecto al almacenamiento de la información, es decir, que tablas se utilizan para persistir los archivos, como se relacionan, etc.
Existen diversas herramientas automatizadas que facilitan el acceso a la información desde la web.	Moodle provee una API para el acceso a la estructura de datos.

Tabla 2 - Comparación de alternativas de extracción

### 3.1.2 Evaluación de herramientas de extracción de datos

Un paso fundamental luego de determinar que la extracción de datos desde la web puede ser el camino a seguir, es realizar un estudio de factibilidad, es decir, comprobar que realmente es posible lograr el cometido mediante esta estrategia. Para esto se evaluaron distintas herramientas.

Web Harvest [50] es una herramienta libre de extracción de datos desde la web, desarrollada en Java [51]. Permite obtener las páginas web deseadas y extraer la información requerida de ellas utilizando técnicas de manipulación tales como XSLT [52], XQuery [53] y expresiones regulares. La herramienta ejecuta scripts escritos en un lenguaje de programación basado en la utilización de etiquetas o *tags* [22]. Para sacar provecho de la herramienta, es imprescindible tener conocimientos de XPath [54], lenguaje que permite construir expresiones que recorren y procesan un documento XML. En el anexo, bajo el título 4. *Herramienta de extracción de datos: Web Harvest*, se detalla el uso de la misma.

Si bien esta herramienta presenta un potencial por sus facilidades, también hay que tener en cuenta que se encuentra discontinuada y que su última actualización fue realizada en Febrero de 2010. Debido a los posibles inconvenientes que esto puede conllevar, como por ejemplo, errores sin corregir, falta de soporte por parte de los desarrolladores, etc., se decide buscar otras herramientas.

Continuando con la investigación, se encuentra una herramienta con muy buenas referencias en la web: JSoup [55]. JSoup es una librería de Java para trabajar con elementos HTML. Provee una API para la extracción y manipulación de datos utilizando DOM [56], CSS [57] y JQuery [58]. JSoup no sólo permite la extracción de información, sino también el llenado de formularios y fácil manejo de Cookies [59], lo cual es necesario para el acceso a la información de la plataforma Moodle ya que la misma puede llegar a solicitar autenticación por parte del usuario.

La documentación es completa y provee ejemplos claros que sugieren que esta herramienta es la apropiada para la tarea de extracción. La API contiene un amplio set de operaciones que permite la búsqueda de elementos HTML por Id, por clase (CSS class), o por etiqueta (HTML Tag). También provee un selector utilizando expresiones regulares.

A continuación se muestra un ejemplo de código de JSoup en el cual, mediante las dos sentencias señaladas debajo, es posible extraer el listado de cursos disponibles en la página principal del Moodle de la Facultad de Ingeniería.

```
Document doc = Jsoup.connect("www.eva.fing.edu.uy/").get();
Elements listaDeCursos = doc.getElementsByClass("name");
```

Además de estas facilidades, también permite sin mayores complicaciones navegar por los cursos y extraer la información considerada relevante.

Para confirmar la viabilidad de esta alternativa, se implementó un prototipo utilizando JSoup que permite acceder al sistema ingresando las credenciales de un usuario existente en el mismo (manejo de Cookies), obtener un listado de los cursos disponibles, y acceder a los OA publicados en ellos.

### 3.1.3 Descripción del problema

Para comenzar a trabajar en el proceso de extracción se debe definir la granularidad mínima (ver sección 2.1.2) de los OA. Se opta por tomar como elemento mínimo a una actividad o un recurso de Moodle. Como se detalló en la sección 2.2, dentro de las actividades de Moodle se encuentran; *Base de datos, Chat, Consulta, Cuestionario, Encuestas predefinidas, Foro, Glosario, Herramienta Externa, Lección, Paquete SCORM, Taller, Tarea, Wiki*. Mientras que un recurso puede ser; *Archivo, Carpeta, Etiqueta, Libro, Página, Paquete de contenido IMS, URL*.

De acuerdo a la definición de OA (ver sección 2.1.1), es posible integrar OA entre sí, con el fin de formar otro OA de mayor granularidad. De esta forma, si se unen distintas actividades/recursos que se llevan a cabo en una misma semana, o dentro de un mismo tema, es posible obtener otro OA. Siguiendo la línea de razonamiento, con un conjunto de semanas se puede obtener un curso entero. En principio se define la entidad LOAR (*Learning Object Activity-Resource*), con estructura jerárquica, donde puede representar una Actividad o un Recurso. Y a su vez, la actividad puede representar un Chat, Consulta, Cuestionario, etc., y el recurso un Archivo, Carpeta, etc.

Se decide tomar la granularidad más fina ya que permite generar sugerencias más específicas a los usuarios que en un futuro utilicen la herramienta de recomendación. Por ejemplo, si se tomara un curso como granularidad más fina, quizás una sugerencia posible puede ser todo un curso completo, cuando en realidad al usuario sólo le interesa un tema específico de dicho curso. Otra alternativa para definir la granularidad, es tomar las semanas o meses de los cursos, los temas, o cualquier agrupación que permita la plataforma.

En la etapa de análisis, se plantea el modelo de dominio preliminar de la realidad planteada descrita en la Figura 1.

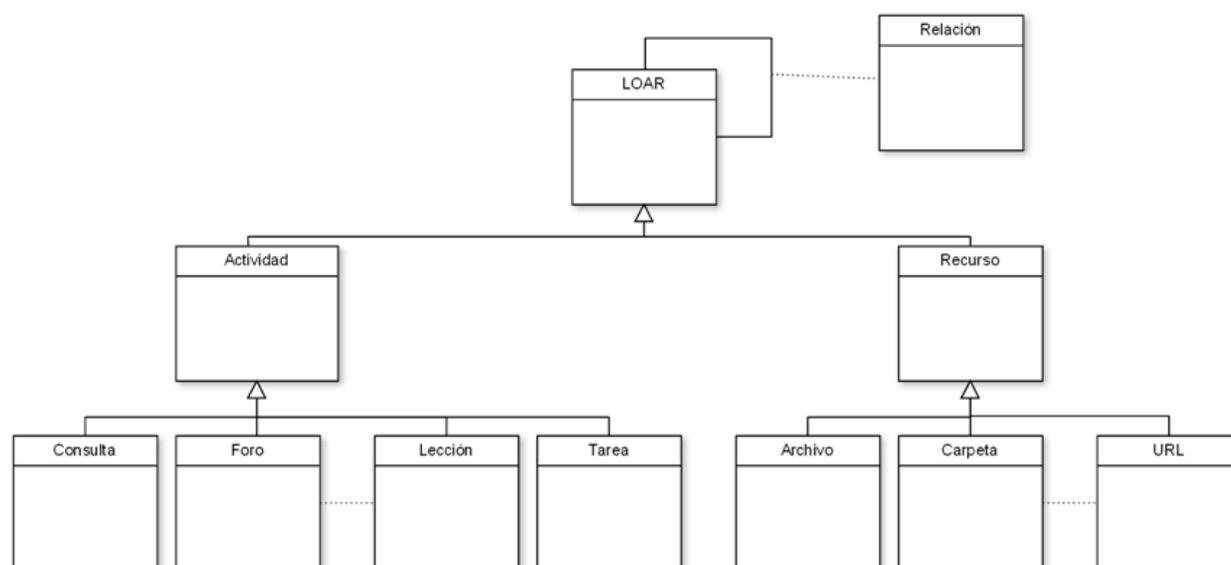


Figura 1 - Modelo de dominio de módulo de extracción

El mismo resume la estructura y representa algunas de las entidades antes mencionadas.

Dada la relevancia de los cursos en el contexto de Moodle y la importancia de que los Elementos estén asociados a un curso, en una primera instancia se pone en discusión la posibilidad de contar con la entidad Curso de forma diferenciada. Luego, se concluye que es más genérico (y por lo tanto más adecuado para este proyecto) mantener una entidad Relación la cual permita relaciones entre LOARs, como por ejemplo, los que pertenecen a una misma semana, a un mismo tema, o a un mismo curso.

En el modelo anterior, se presenta un conjunto de elementos con estructura jerárquica que representan los tipos de OA disponibles en la plataforma. En esta etapa de análisis, de acuerdo a las entidades existentes en Moodle, se discriminan los OA por Actividad y Recurso, y a su vez estos se separan en los diferentes tipos de elementos (Foro, Archivo, etc., detallados anteriormente). Dicha discriminación se presenta de acuerdo a los diferentes atributos que tienen estos tipos de elemento. Un primer análisis permite detectar que un OA de tipo Archivo posee un conjunto específico de metadatos, como por ejemplo la extensión (pdf, ppt, etc.), la cual carece de sentido en el OA de tipo Foro. Explicado esto, también cabe destacar las coincidencias existentes en estos elementos. Es aquí que toma importancia la estructura jerárquica y la herencia de atributos, ya que todos los elementos disponibles poseen un nombre, fecha de creado, etc.

### 3.1.4 Solución propuesta para la extracción

Como parte del trabajo de este proyecto, se desarrolla una aplicación web que provee las funcionalidades necesarias para realizar la extracción de objetos de aprendizaje de la plataforma Moodle. La misma contiene una interfaz web, la cual permite a un usuario con los permisos necesarios, configurar parámetros básicos y realizar el proceso de extracción.

En la Figura 2 se presenta el diagrama del modelo de casos de uso, conteniendo las operaciones básicas del módulo de extracción detalladas a continuación.

**Login:** Este caso de uso permite identificar al usuario en el sistema mediante el ingreso de credenciales.

**Configurar extracción:** Mediante este caso de uso, un usuario previamente autenticado y con los privilegios necesarios, puede configurar los parámetros del módulo utilizados para la posterior extracción de información. Los parámetros son los siguientes:

- *Moodle Server:* Indica la URL del servidor donde se encuentra instalada la plataforma Moodle. (p.e. *www.eva.fing.edu.uy*).
- *Moodle Session Name:* Especifica el nombre del atributo de sesión (o *cookie de sesión*) de la aplicación web. La plataforma Moodle permite parametrizar este campo, por lo que su nombre podría variar de sistema en sistema. Es necesario para el manejo de sesiones mediante el proceso de extracción.
- *Usuario:* Indica el nombre de usuario de Moodle utilizado por el proceso extractor para

acceder a los cursos. Solamente se podrá extraer información de los cursos públicos (que no requieren autenticación) o de los que el usuario tenga acceso.

- *Contraseña*: Es la contraseña de acceso del usuario explicado en el punto anterior.

**Extraer información:** Una vez autenticado y habiendo configurado los parámetros necesarios para la extracción, un usuario con los privilegios necesarios puede comenzar el proceso de extracción de información. El mismo consiste en conectarse a la plataforma Moodle (configurada) y mediante sucesivos pedidos HTTP (iterando por las páginas de los cursos) se obtienen los distintos archivos HTML, de los cuales mediante técnicas avanzadas de parseo, se va realizando la extracción de atributos para luego almacenarlos en la base de datos de la aplicación.

**Logout:** Se elimina la información de la sesión del usuario autenticado.

El flujo de ejecución de los casos de uso queda representado en la Figura 3.

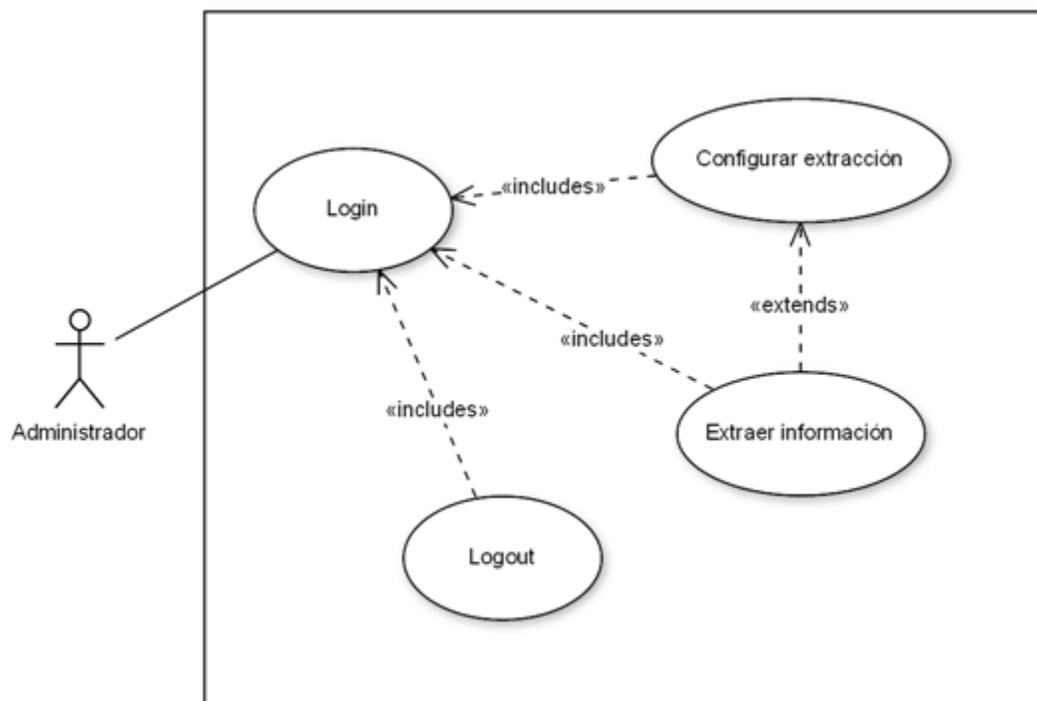


Figura 2 - Modelo de casos de uso de módulo de extracción

## 3.2 Recomendación

En cuanto la recomendación de información, existen diversas técnicas utilizadas en la actualidad. Sin embargo, al día de hoy no hemos encontrado demasiados proyectos aplicados a la recomendación de objetos de aprendizaje. En 2010 se realizó un estudio sobre la efectividad del uso de técnicas de filtrado colaborativo basado en usuarios en el repositorio

MERLOT obteniendo resultados positivos [60] y en 2012 se desarrolló un framework de recomendaciones híbridas para repositorios de OA, denominado Delphos [61]. Es por ello que entra dentro de los objetivos de este proyecto el desarrollo de un sistema web que provea las funcionalidades necesarias para la recomendación de elementos educativos tanto a estudiantes como docentes.

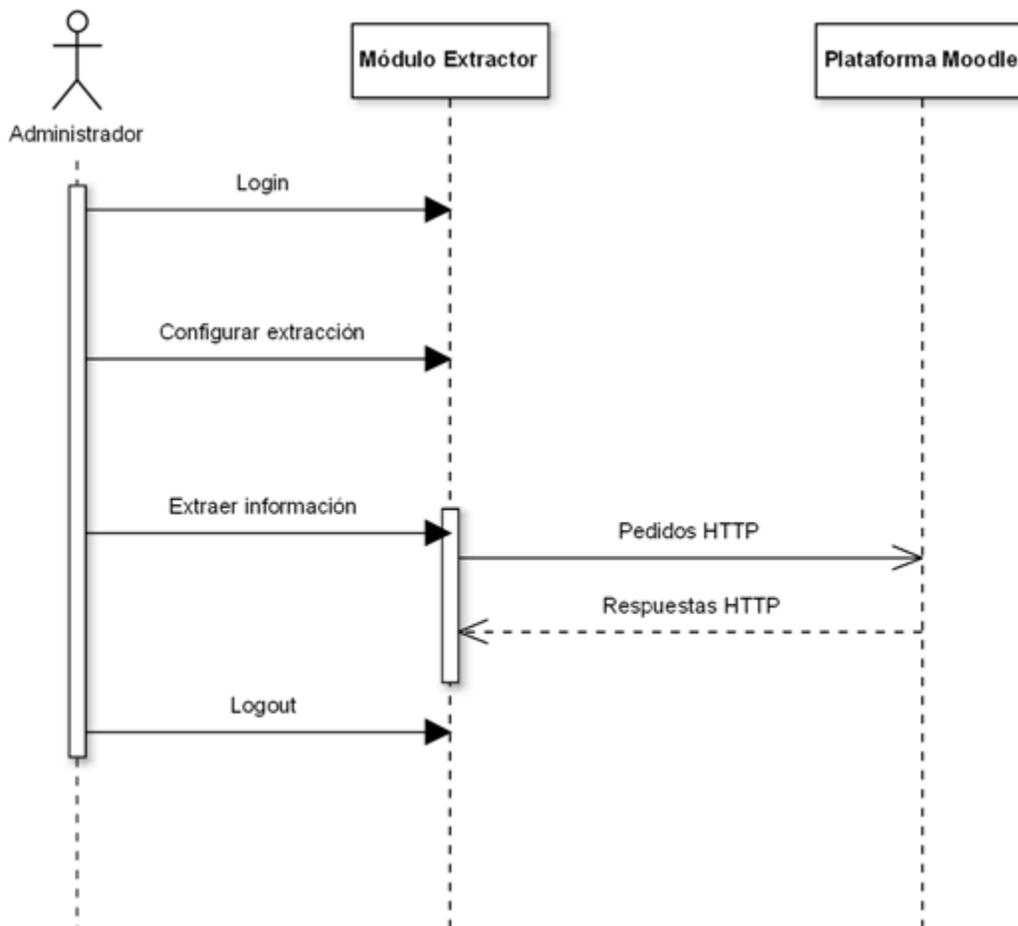


Figura 3 - Diagrama de flujo típico de módulo de extracción

### 3.2.1 Evaluación de técnicas de recomendación

Dentro de las técnicas estudiadas (ver sección 2.3), se evalúa cuáles son más apropiadas en el contexto de recomendación de OA extraídos de la plataforma Moodle.

#### 3.2.1.1 Friendsourcing

Si bien es una técnica muy utilizada en diversos ámbitos, en el área de los objetos de aprendizaje y Moodle puede no ser tan exitosa además de ser compleja de implementar. Una alternativa es intentar aprovechar las relaciones de amistad existentes en las redes sociales, como por ejemplo en Facebook. Para ello se deberían integrar ambas aplicaciones mediante el desarrollo de un módulo de Moodle, o implementar una tercera aplicación que se encargue de

dicha comunicación para obtener las relaciones. Además de los altos costos de desarrollo, los principales problemas de esta alternativa son los siguientes:

- Poca relación entre los amigos de una red social y el interés mutuo entre los contenidos académicos.
- Gran dificultad de obtener la confianza necesaria por parte de los usuarios para que ingresen las credenciales de acceso a su cuenta de la red social.

Por otra parte, Moodle permite mantener conversaciones entre usuarios (mediante *chats*), lo cual que se podría tomar como una relación de amistad. Sin embargo, para acceder a esta información se requieren permisos adicionales a la hora de realizar la extracción y se debería contar con autorización por parte de los usuarios para utilizar estos datos. También es importante tener en cuenta la relación costo/beneficio de realizar este trabajo, ya que por lo indagado en la realización de este proyecto, se detecta muy poco uso de las conversaciones entre usuarios por lo que no sería mucha la información a extraer.

Es por esto que se concluye en que en el marco de este proyecto no es conveniente utilizar las recomendaciones basadas en Friendsourcing.

### 3.2.1.2 Crowdsourcing

La técnica de Crowdsourcing es muy amplia e incluye diversos algoritmos. Los más utilizados son los conocidos como Filtrado Colaborativo. Dentro de esta línea de trabajo se puede discriminar entre distintas alternativas, como por ejemplo: Filtrado colaborativo basado en usuarios o basado en ítems. En cuanto a la decisión de cual seleccionar, hay que tener en cuenta los siguientes factores: Si la lista de ítems no varía mucho, es conveniente realizar un análisis basado en ítems ya que la relación ítem a ítem se hará más precisa con el tiempo. En caso contrario, si la lista de ítems es inestable, un análisis basado en los usuarios sería más conveniente debido a que no se tendrá información respecto a las relaciones que involucren a los nuevos ítems. Otro factor que influye en la decisión de optar cual algoritmo es más conveniente usar, es la dimensión del espacio de usuarios e ítems. En un escenario donde existen millones de usuarios y relativamente pocos ítems, es más sencillo realizar un análisis ítem a ítem [62]. Se ha demostrado que los algoritmos basados en ítems son computacionalmente más rápidos de implementar y proveen mejores resultados [63].

En Moodle, generalmente la cantidad de ítems dentro de un curso tiende a ser constante, es decir, a medida que pasa el tiempo, el crecimiento de ítems tiende a ser menor. Esto es debido a que los cursos educativos no sufren frecuentemente grandes modificaciones en su estructuración. La mayoría de los cursos en Moodle ya cuentan con una estructura básica, con ciertas actividades y recursos que son indispensables para el mismo y luego se van adaptando de acuerdo a sus necesidades.

En este caso se tienen muchos usuarios en relación a la baja cantidad de objetos de aprendizaje, por lo que el algoritmo basado en ítems es el mejor camino a seguir. De cualquier forma, se realizan recomendaciones utilizando el algoritmo basado en usuarios con el fin de realizar comparaciones entre las técnicas, ya que en un futuro, la situación podría cambiar y se

podría tener un aumento significativo de ítems, por ejemplo, si se quiere recomendar elementos de todos los cursos de la facultad, o de otras facultades y/o universidades.

### 3.2.1.3 Recomendaciones grupales

El objetivo de las recomendaciones grupales es determinar un puntaje grupal para cada ítem que refleje el interés y preferencia de todos los miembros del grupo. En general, los miembros de un grupo no tienen la misma preferencia para cierto ítem, por lo que se debe diseñar una forma inteligente de calcular la puntuación de cada ítem en base a la preferencia de cada miembro del grupo por el mismo. Se han desarrollado una gran cantidad de estrategias de decisión grupal en los últimos años. Estas estrategias buscan comprender cómo los miembros de un grupo llegan a un consenso, es decir, dado un conjunto de preferencias individuales para un cierto ítem, cuál es la preferencia grupal para ese ítem.

Hay dos aspectos principales para determinar el puntaje grupal de cierto ítem. Primero, el puntaje grupal debe determinar el grado de *concordancia* que hay sobre un ítem, cuanto más miembros prefieran un ítem, mayor deberá ser el puntaje de concordancia. A su vez, el puntaje grupal debe reflejar el grado de *desacuerdo* que hay entre los miembros del grupo sobre cada ítem. El primer aspecto se denomina *group relevance* (relevancia grupal), el segundo aspecto se denomina *group disagreement* (discrepancia grupal). A partir de estos dos aspectos se obtiene la función consenso, la cual determina el puntaje del grupo para cierto ítem.

Para determinar el puntaje grupal, se diseñó un algoritmo que considera los aspectos mencionados anteriormente. El algoritmo recibe como entrada el grupo de usuarios y la cantidad de recomendaciones que se desean. Para obtener las recomendaciones, se calcula un valor para cada ítem que no fue puntuado por ninguno de los usuarios, y este valor es determinado por la función consenso. Para este cálculo, la función consenso utiliza la relevancia grupal y la discrepancia grupal, dando la posibilidad de ponderar estos campos, es decir, se le puede dar más importancia a la relevancia que a la discrepancia o viceversa, o la misma importancia. Los usuarios pertenecientes al grupo que no hayan realizado ninguna puntuación no son tenidos en cuenta por el algoritmo, es decir, no afectan en el cálculo para la obtención de recomendaciones.

### 3.2.1.4 Conclusión de la evaluación de las técnicas de recomendación

En resumen, se decide utilizar las técnicas de Filtrado Colaborativo basado en usuarios y en ítems, y Recomendaciones grupales. Se descarta para este proyecto la implementación de Friendsourcing y de técnicas híbridas (o mixtas) que utilicen metadatos de los OA para generar las recomendaciones, ya que la extracción de metadatos (de calidad) en principio no es la prioridad del proyecto. Sin embargo, éstas técnicas son tenidas en cuenta como trabajos a futuro. Ver sección 8.2.

### 3.2.2 Evaluación de herramientas de recomendación

Apache Mahout es una destacada herramienta en el área de la recomendación, desarrollada en un proyecto perteneciente a la Fundación Apache [64]. Su objetivo es la construcción de una librería de aprendizaje automático escalable, donde con escalable se refiere a trabajar con grandes conjuntos de datos.

Mahout actualmente provee principalmente tres funcionalidades:

- Recomendación de elementos de acuerdo al comportamiento de los usuarios, intentando encontrar los objetos que éstos quisieran.
- Agrupación o *clustering* de elementos, por ejemplo, dado un conjunto de documentos de texto, se pueden asociar en subgrupos de acuerdo a la temática de los mismos.
- Clasificación automática de documentos de acuerdo a ciertas categorías existentes teniendo en cuenta las similitudes entre los documentos.

Los algoritmos fundamentales para la agrupación, clasificación y filtrado colaborativo se implementan basándose fuertemente en Apache Hadoop [65] usando el paradigma Map Reduce [66]. El software de Apache Hadoop es un framework que permite el proceso de grandes conjuntos de datos de forma distribuida. A su vez, el paradigma de Map Reduce es un modelo de programación para el proceso y generación de grandes conjuntos de datos. Sin embargo, el uso de la librería no está restringido a ambientes que se ejecutan en un solo nodo, además de que las bibliotecas del núcleo están altamente optimizadas para permitir un buen rendimiento también para los algoritmos no distribuidos.

En cuanto a las recomendaciones, Mahout provee un conjunto flexible de operaciones de filtrado colaborativo junto a diversas implementaciones de algoritmos con todo lo necesario para construir un sistema de recomendaciones. Está diseñado para realizar un procesamiento eficiente, ser fácil de utilizar y permitir integraciones con diferentes conjuntos de datos (también conocidos como *datasets*) [67]. Dado que el filtrado colaborativo se basa en las interacciones entre usuarios e ítems, Mahout se apega a dicho concepto modelando las preferencias como la tripleta (*usuario, ítem, valor*). Para mejorar el rendimiento de memoria, sólo se permiten identificadores numéricos.

Dentro de filtrado colaborativo, las técnicas denominadas “Vecino más cercano” son las más utilizadas. Estos métodos computan las recomendaciones buscando usuarios con gustos similares, o ítems que han sido valorados de forma similar. Mahout proporciona implementaciones para ambos enfoques: Basados en usuarios y basados en ítems.

En el anexo 5. *Herramienta de recomendación: Apache Mahout* se detalla el uso de esta herramienta.

### 3.2.3 Descripción del problema

Una vez analizadas las técnicas de recomendación, que en principio se determinan como factibles de implementar y de utilidad para aplicar de acuerdo a la realidad, junto a las

herramientas a utilizar, se puede diseñar la solución en detalle del problema a resolver. A las entidades manejadas en el módulo de extracción (LOAR: Objetos de aprendizaje extraídos) se le suman las necesarias para generar las recomendaciones de elementos. A primera vista se destacan los Usuarios y Grupos. En una primera etapa de análisis, se plantea en la Figura 4 el modelo de dominio preliminar de la realidad planteada.

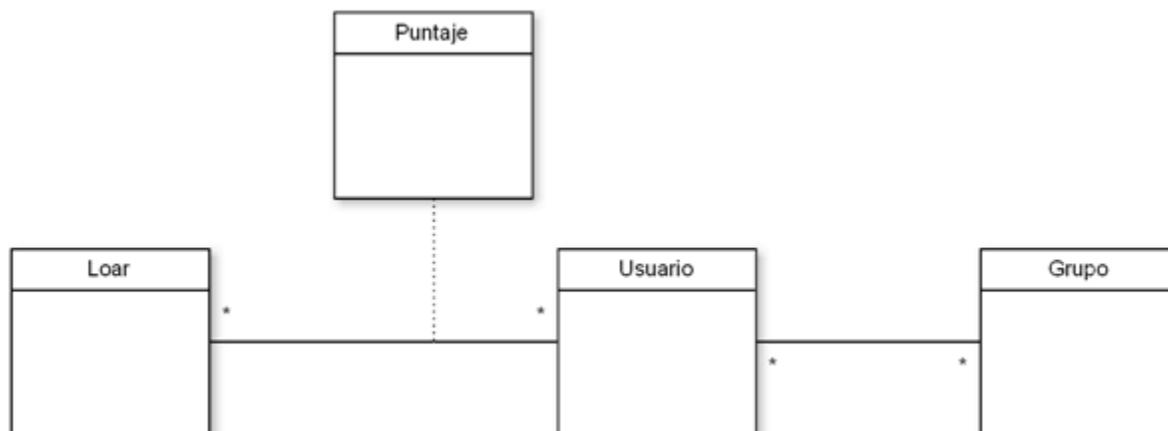


Figura 4 - Modelo de dominio de módulo de recomendación

En la Figura 4 se representan las entidades relevadas junto a sus relaciones. Se muestra la clase asociativa Puntaje que se liga a la relación Usuario-Loar, lo que representa el puntaje que otorga cada usuario a los elementos recomendados. A su vez, los usuarios también pueden pertenecer a varios grupos. Esta información es posteriormente utilizada para obtener las recomendaciones grupales.

### 3.2.4 Solución propuesta para las recomendaciones y búsquedas

Se desarrolla una aplicación web que provee las funcionalidades básicas de un sistema de recomendaciones, utilizando los objetos de aprendizaje extraídos mediante el módulo de extracción especificado anteriormente. También es de suma importancia facilitar el acceso de los usuarios a las recomendaciones, por lo que se plantea la posibilidad de incorporar las recomendaciones generadas directamente en la plataforma Moodle.

Otro punto a tener en cuenta es la gestión del sistema, y por ello se incluyen funcionalidades para el alta, baja y modificación de usuarios y grupos.

En las Figuras 5 y 6 se presentan los diagramas del modelo de casos de uso, conteniendo las operaciones básicas del módulo de recomendación y gestión junto a una breve descripción de las mismas. Las operaciones referentes a la autenticación (login, logout) fueron explicadas en la sección 3.1.4)

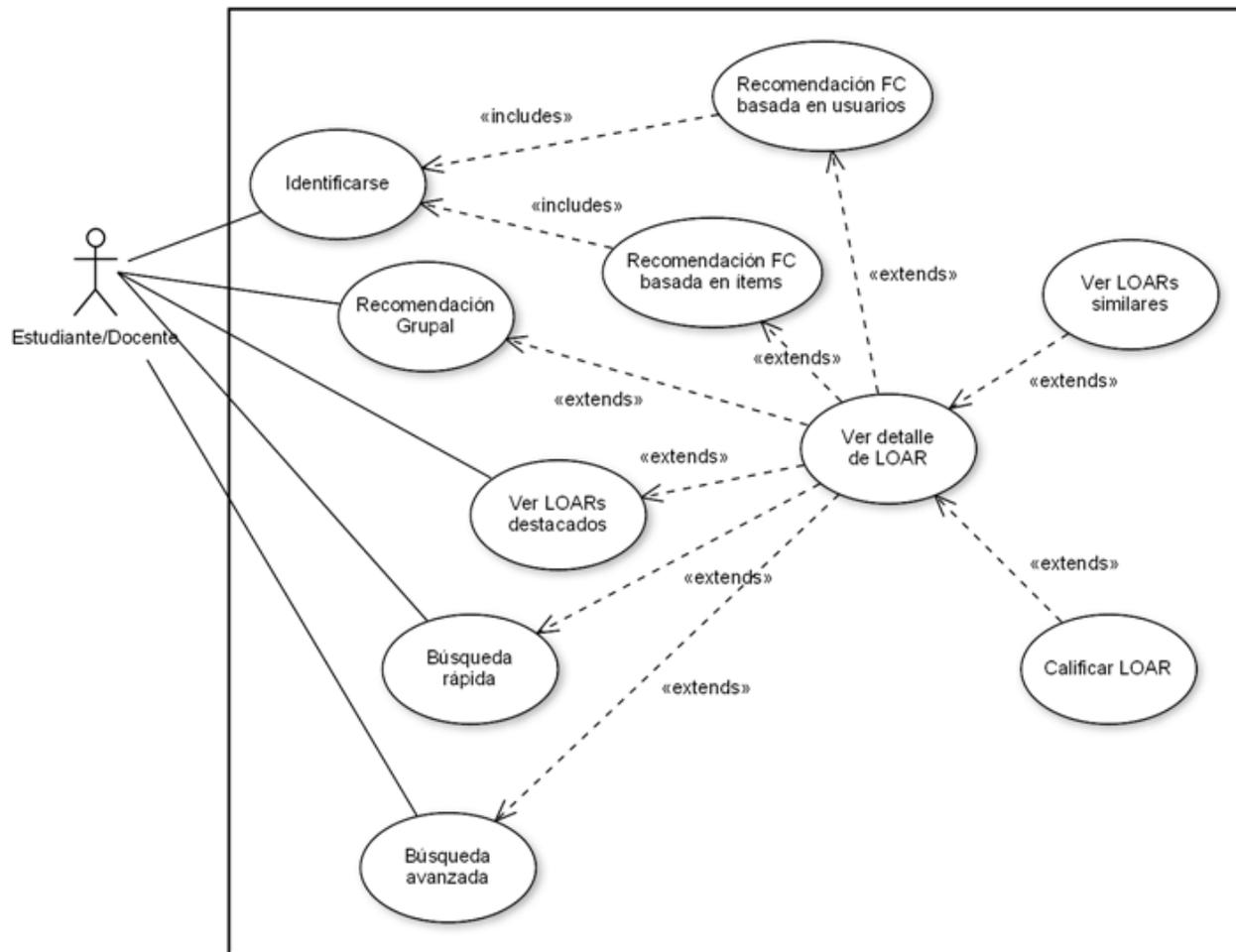


Figura 5 - Modelo de casos de uso de módulo de recomendación

**Recomendación de Filtrado Colaborativo basado en usuarios:** Este caso de uso permite a un usuario identificado obtener un listado de elementos basándose en técnicas de recomendación de filtrado colaborativo basado en usuarios.

**Recomendación de Filtrado Colaborativo basado en ítems:** Un usuario identificado puede obtener un listado de elementos basándose en técnicas de recomendación de filtrado colaborativo basado en ítems.

**Recomendación grupal:** Un usuario puede obtener un listado de elementos basándose en técnicas de recomendaciones grupales. El sistema despliega un listado de grupos donde el usuario puede filtrarlos por nombre. El usuario selecciona el grupo y el sistema despliega los resultados.

**Ver detalle de LOAR:** Cuando se accede a un elemento, se registra el pedido del usuario, tanto en la tabla de accesos (aumentando la cuenta) como en la tabla de logs de acceso (se registra la fecha y tipo de recomendación que llevó al usuario a ese elemento). También se chequea si el acceso se realiza desde el Moodle, en cuyo caso se agrega al usuario al grupo

que representa al curso. En caso contrario, se le solicita al usuario que se identifique en el sistema mediante el ingreso de su email. Finalmente, se retorna el elemento deseado.

**Calificar un LOAR:** Un usuario identificado (ingresó desde el Moodle, especificó su email o se logueó) puede otorgar un puntaje a un elemento seleccionado. Dicha acción se registra tanto en la tabla de puntajes (en caso de no existir, se crea la tupla, en caso de existir, se actualiza con el valor ingresado), como en la tabla de logs de puntajes (se registra la fecha y tipo de recomendación que llevó al usuario a ese elemento).

**Ver LOARs similares:** Cuando un usuario accede al detalle de un elemento, se puede acceder a un listado de elementos similares. Se entiende por elementos similares cuando varios usuarios los puntuaron de forma similar.

**Ver LOARs destacados:** Se listan los elementos con mayor puntaje dentro del sistema.

**Búsqueda rápida de LOARs:** Un usuario puede realizar una búsqueda rápida de elementos desde cualquier página dentro del sistema. El usuario ingresa un valor descriptivo del elemento y el mismo se matchea contra los metadatos básicos de un LOAR.

**Búsqueda avanzada de LOARs:** Un usuario puede acceder a un formulario específico que permite filtrar por diversos campos. El sistema retorna los elementos que se corresponden a los campos ingresados.

**Identificarse:** Un usuario no identificado puede ingresar su correo para acceder a recomendaciones personalizadas y acceder al detalle de LOARs.

**Crear grupo:** Un usuario autenticado y con los privilegios necesarios puede crear un nuevo grupo para luego obtener recomendaciones grupales. El usuario ingresa el nombre del grupo, y luego selecciona los usuarios que formarán parte del mismo.

**Ver detalle de grupo:** Un usuario puede acceder al detalle de un grupo. El sistema despliega el nombre del grupo junto al listado de usuarios que lo integran.

**Editar grupo:** Un usuario autenticado y con los privilegios necesarios puede editar un grupo existente. El usuario actualiza el nombre del grupo, y luego selecciona los usuarios que formarán parte del mismo.

**Crear usuario:** Este caso de uso permite a un usuario autenticado y con los privilegios necesarios crear un nuevo usuario en el sistema.

**Ver detalle de usuario:** Este caso de uso permite acceder al detalle de un usuario. El sistema despliega los datos del mismo.

**Editar usuario:** Un usuario autenticado y con los privilegios necesarios puede actualizar los datos de un usuario existente.

**Enviar formulario de contacto:** Un usuario puede enviar un mensaje al administrador del sistema, con el fin de reportar un error, solicitar un aumento de privilegios dentro del sistema, o cualquier inquietud que éste posea.

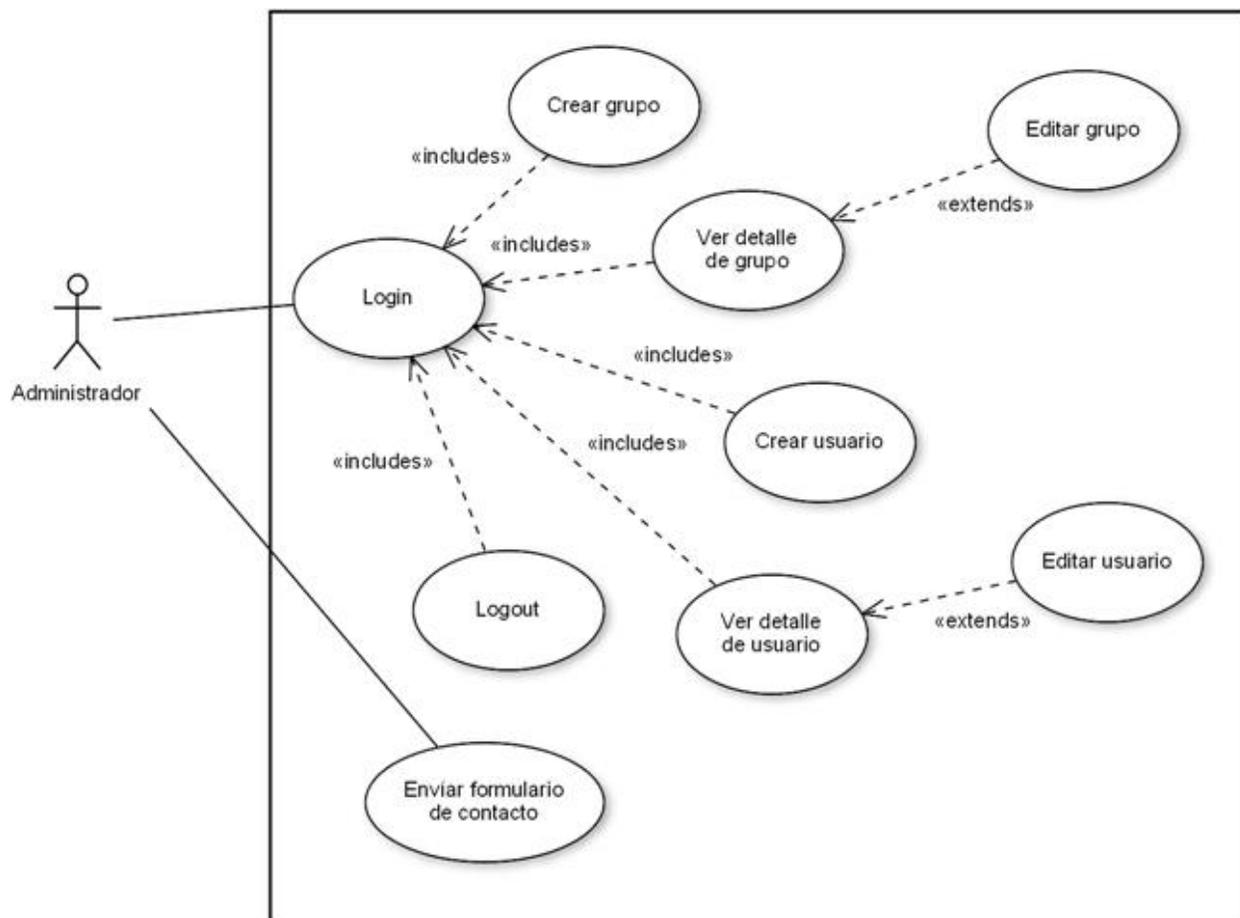


Figura 6 - Modelo de casos de uso de administración

En la Figura 7 se muestra el flujo típico de acceso al sistema de recomendaciones. El flujo comienza cuando el usuario accede al sistema de recomendaciones, el mismo debe identificarse y a partir de allí puede acceder a las recomendaciones personalizadas. Luego de obtener el listado con los elementos recomendados, se permite acceder al detalle del Loar de interés y finalmente valorar dicho elemento.

El diagrama de la Figura 8 describe otro flujo de acción posible. Para facilitar el acceso por parte de los usuarios a las recomendaciones, las mismas se integran a la plataforma. Esto sucede cuando el estudiante accede al Moodle, la plataforma envía un pedido al Sistema de recomendaciones, identificando al usuario y retornando el listado de elementos. Desde allí, el

estudiante puede acceder al detalle del Loar de interés en el mismo sistema de recomendaciones donde es posible valorar dicho elemento.

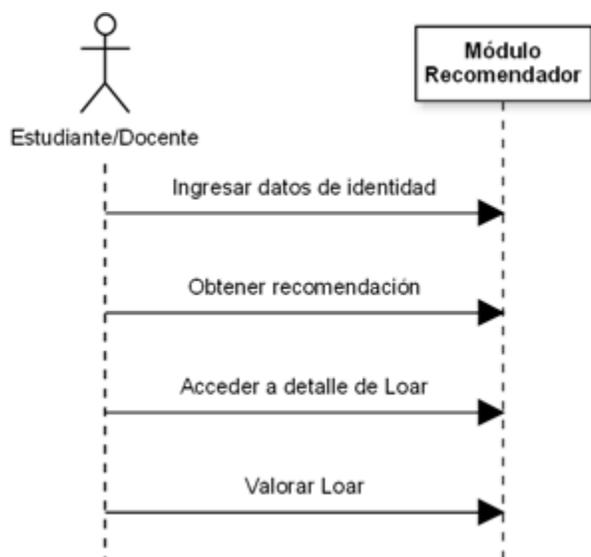


Figura 7 - Diagrama de secuencia del flujo principal del sistema de recomendaciones

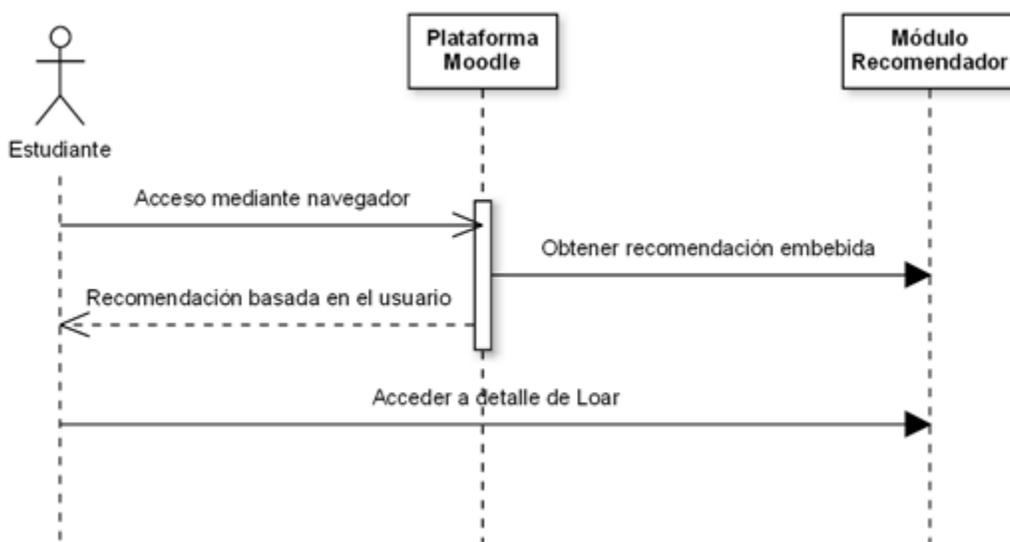


Figura 8 - Diagrama de flujo típico de acceso a Moodle con recomendaciones embebidas

En cuanto a las actividades de gestión, se destacan las altas de grupo y usuario, así como la inclusión de un usuario a un grupo. Una posible acción a seguir luego de organizado un grupo, es la de obtener recomendaciones grupales. La Figura 9 muestra algunas de estas actividades.

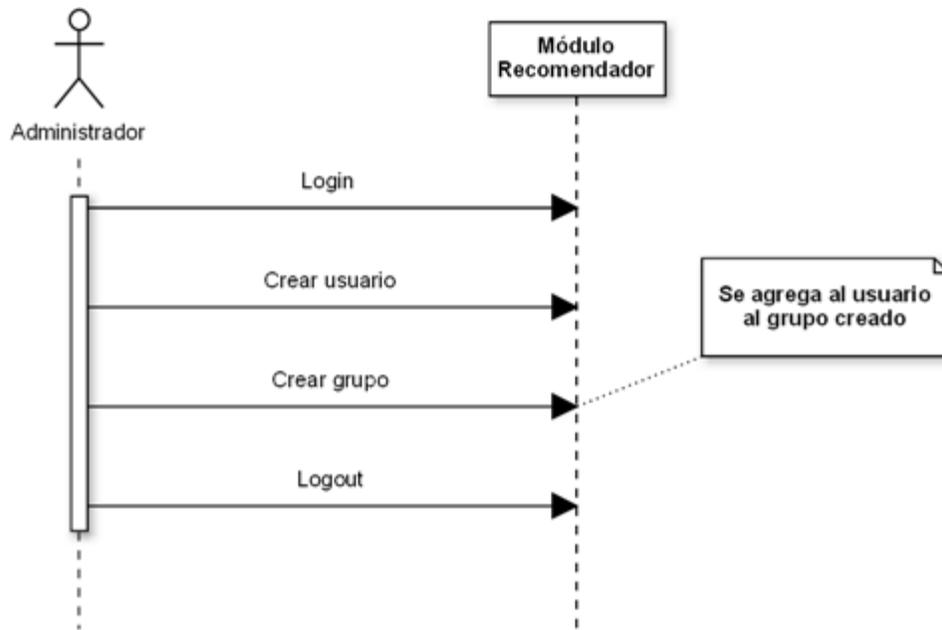


Figura 9 - Diagrama de flujo de actividades de administración

## 4 Diseño

Para resolver el problema planteado, se decide desarrollar una aplicación web denominada *MoodleRecommender*, que contenga un conjunto de funcionalidades básicas para la extracción y recomendación de OA. Las mismas fueron analizadas y detalladas en las secciones 3.1.4 y 3.2.4. A continuación se describe el diseño de la solución basándose en la información relevada en la etapa de análisis.

Desde el punto de vista de la extracción de objetos de aprendizaje, hay varias consideraciones a tener en cuenta. Respecto a la información a extraer, como se explicó anteriormente, existen diversos estándares para almacenar metadatos que proveen ciertas ventajas desde el punto de vista de la interoperabilidad. De acuerdo a los objetivos propuestos y teniendo en cuenta la baja cantidad y calidad de los metadatos existentes en los elementos de la plataforma, queda fuera del alcance la implementación de alguno de estos estándares, ya que la mayoría necesitan obligatoriamente un mínimo de atributos. En cambio, se opta por realizar un breve estudio sobre la calidad de los metadatos existentes en los objetos de la plataforma, más específicamente en el curso de Fundamentos de la Web Semántica, con el fin de extraer los datos que provean mejores resultados a la hora de recuperar la información. Otro punto importante a tener en cuenta, son los distintos tipos de objetos existentes. Nuevamente, centrándose en el curso objetivo de las pruebas, se destacan como buen punto de partida los elementos de tipo Archivo, tanto por su facilidad de procesar como por su abundancia con respecto a otros, como por ejemplo los Foros, que requerirían extensas técnicas de procesamiento de lenguaje natural. Es por todo esto que se decide comenzar a trabajar con el tipo Archivo para el sistema en producción. A continuación se presenta un listado con los metadatos seleccionados dentro de los presentes, discriminando entre los que son comunes a las demás actividades.

Los atributos comunes a todos los elementos son:

- Nombre: Es el campo ingresado como *nombre* del elemento por el usuario que lo subió a la plataforma. Se trata del dato más importante ya que describe brevemente al objeto y además se encuentra presente en todos los elementos por ser un dato obligatorio para Moodle.
- Referencia: Es la URL donde se encuentra ubicado el detalle del elemento en la plataforma Moodle.
- Subido Por: Identifica al usuario de Moodle que subió el contenido al sitio.
- Fecha Subido: Fecha en que el elemento fue subido al sitio.

Existen dos clases de atributos de los elementos de tipo Archivo: Los que son extraídos de la plataforma, es decir, los metadatos que fueron ingresados por el usuario que sube el documento a Moodle. Y los que son extraídos del archivo, es decir, los metadatos generados por el usuario que crea el documento. Es importante remarcar que estos dos conjuntos de metadatos no tienen relación entre sí, ya que los mismos pueden ser generados por personas

distintas en momentos distintos. A continuación se destacan dichos atributos:

- Descripción: Este atributo se extrae desde la plataforma, y es la descripción que ingresa el usuario a publicar el archivo en el sitio.
- Título: Es el metadato *'title'* específico del archivo.
- Autor: Es el metadato *'author'* específico del archivo.
- Creador: Es el metadato *'creator'* específico del archivo.
- Tema: Es el metadato *'subject'* específico del archivo. Describe el la temática que abarca el archivo.
- Fecha de creación: Es el metadato *'creationDate'* específico del archivo.
- Fecha de modificación: Es el metadato *'modDate'* específico del archivo. Indica la fecha última modificación del archivo.
- Nombre de archivo: Es el nombre del archivo en cuestión. Se extrae de la URL de acceso al elemento en la plataforma.
- Extensión: Indica la extensión y por lo tanto el tipo de archivo (p.e. pdf, ppt, doc, etc.).
- Tamaño: Indica el tamaño del archivo en megabytes. Se extrae desde la plataforma web en los casos que se encuentra disponible.

Se seleccionan estos atributos como conjunto de metadatos teniendo en cuenta la utilidad de los mismos a la hora de la recuperación de información. Por ejemplo, un escenario típico consiste en realizar una búsqueda por *nombre* o *descripción*, pero también podría interesar un elemento subido a la plataforma en la semana actual, por lo que hay que analizar el atributo *fecha de subida*. Otro ejemplo es que se quiera recuperar elementos subidos por cierto docente, referente en un área (atributo *subido por*). Siguiendo esa línea de razonamiento se puede encontrar la utilidad a los demás metadatos específicos de los archivos, sumándoles la facilidad de extracción de esa información ya que no se requiere procesamiento de texto HTML, sino que se accede mediante funcionalidades provistas por la librería iText [68] de JAVA una vez descargado el archivo.

Desde el punto de vista de las recomendaciones, surge la necesidad de registrar los históricos tanto de los accesos como de los puntajes de los recursos, para mejorar la evaluación de efectividad de los algoritmos en un futuro.

En la Figura 10 se presenta el diagrama de clases de diseño, el cual describe las entidades manejadas en el proceso de desarrollo y las relaciones entre ellas. Las entidades Log Accesos y Log Puntajes registran cada acceso y puntaje sobre un Archivo respectivamente. Cabe destacar el atributo *tipoAcceso* el cual permite determinar la forma en que el usuario llegó al Objeto. Es decir, si fue mediante una recomendación de filtrado colaborativo basado en usuarios, en ítems, mediante recomendaciones grupales, a través del buscador, por el listado de elementos destacados o por elementos similares a un recurso seleccionado. Este atributo puede ser utilizado para realizar estudios en profundidad y estadísticos sobre el uso de la herramienta de recomendación y recuperación de información. De esta forma, se puede analizar rápidamente qué mecanismo resulta más efectivo para los usuarios, observando los logs de acceso y de puntaje. En cambio, la entidad Puntaje registra el último puntaje otorgado

por un usuario sobre un recurso, y a diferencia con las anteriores (*Log Accesos* y *Log Puntajes*), sólo puede existir una instancia de Puntaje por cada par Usuario-Loar.

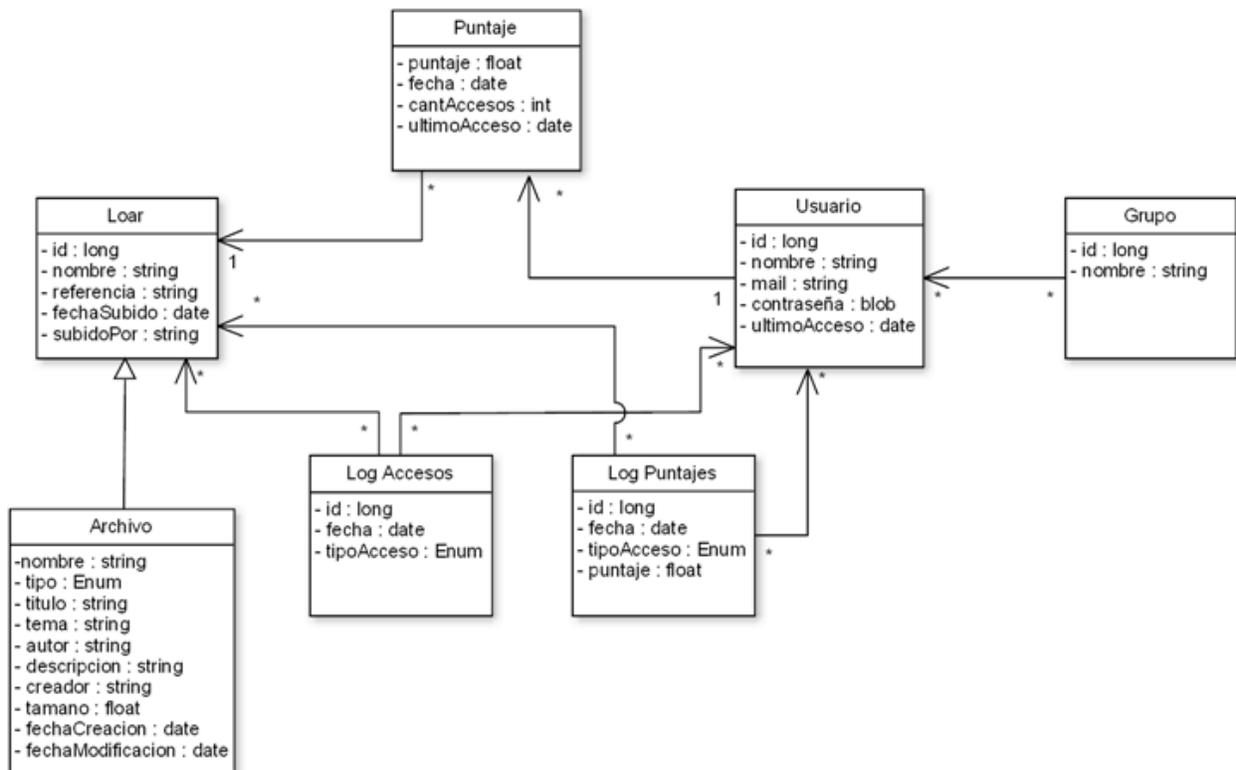


Figura 10 - Diagrama de clases del sistema

## 4.1 Arquitectura

Se desarrolla una aplicación utilizando una arquitectura en capas [69], destacando 3 niveles;

- Capa de persistencia, que contiene el modelo de datos necesario para persistir la información del sistema.
- Capa de negocio, que contiene la lógica del sistema, donde se incluyen los algoritmos de extracción y recomendación.
- Capa de presentación, que contiene las vistas que despliegan la información a los usuarios.

En la Figura 11 se puede observar la estructura de las capas antes mencionadas.

### 4.1.1 Capa de presentación

Para la capa de presentación se implementa el patrón de arquitectura de software denominado *Model-View-Controller* (o MVC) [70], el cual separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las

comunicaciones. Como controladores se utilizan Java Servlets [71] para acceder a la capa inferior y páginas JSP (*Java Server Pages*) [72] como generadores de las vistas que presentan la información a los usuarios. En la sección 7. *Interfaz de usuario* del anexo, se presentan algunos ejemplos de la interfaz. También se hace uso de hojas de estilo CSS (*Cascading Style Sheets*) para enriquecer el formato de los datos desplegados.

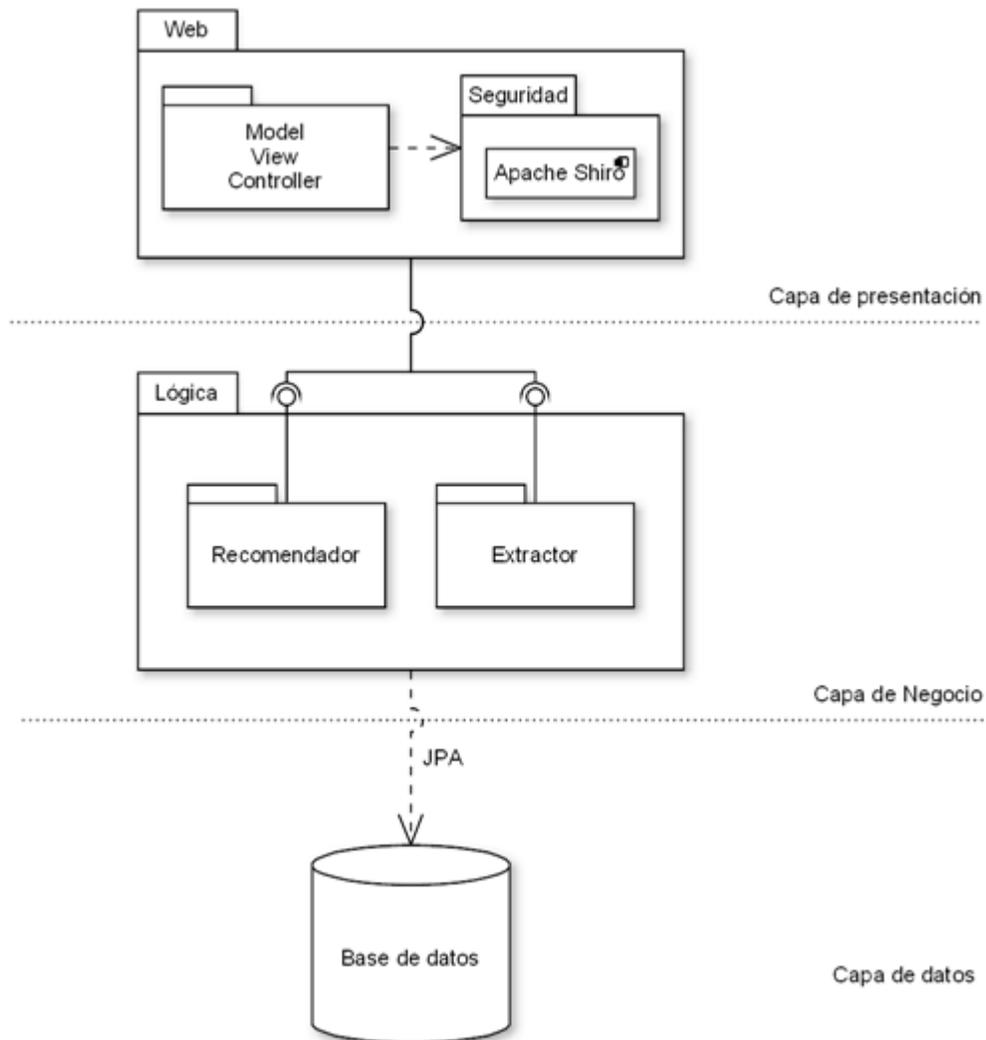


Figura 11 - Arquitectura del sistema

#### 4.1.1.1 Seguridad

En cuanto a la seguridad del sistema, se define una política de control de acceso, es decir, autenticación y autorización, a partir de un modelo basado en roles [73]. A cada usuario se le asigna uno o más roles, los cuales tienen asignados uno o más permisos. A su vez, cada operación del sistema, está asociada a un permiso. Lo que se resume en que para que un usuario pueda acceder a una funcionalidad, el mismo debe estar identificado (autenticación) y tener los privilegios correspondientes (autorización). Para la implementación de estas políticas se utiliza un framework de seguridad llamado Apache Shiro. Apache Shiro [74] es una

herramienta desarrollada por la Fundación Apache, cuyo objetivo es implementar mecanismos de seguridad en aplicaciones Java, proporcionando facilidades para la autenticación, autorización, cifrado criptográfico y manejo de sesiones.

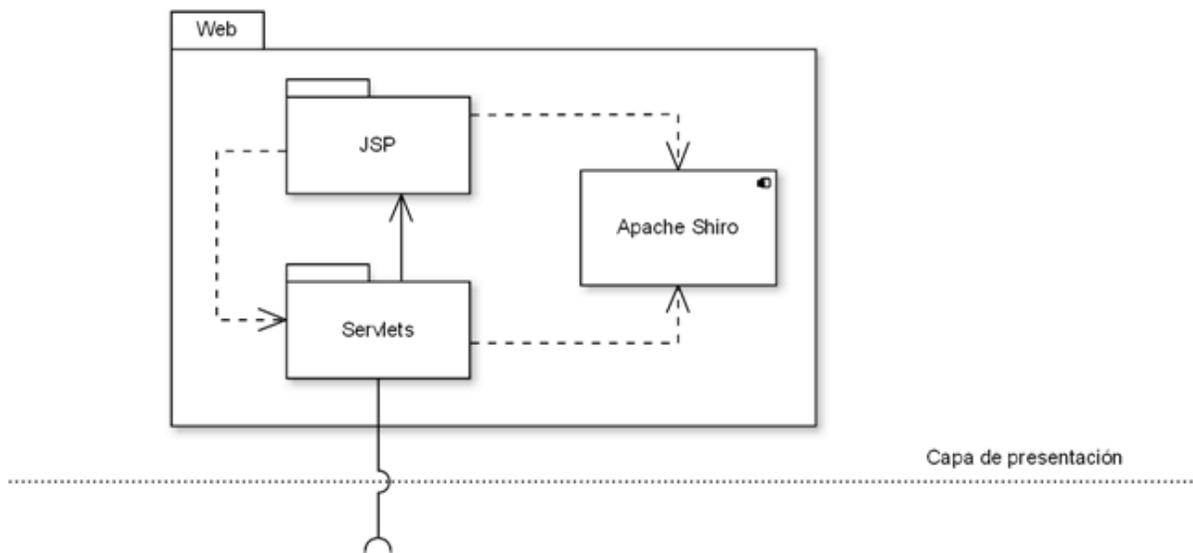


Figura 12 - Diagrama de paquetes de capa de presentación

Mediante una serie de simples configuraciones (conexiones a la base de datos, descripción del esquema de datos, etc.) el framework provee un conjunto de operaciones que facilitan las tareas de autenticación (caso de uso Login) y de decisión sobre que usuarios pueden invocar determinadas acciones del sistema. También es posible seleccionar el mecanismo de cifrado utilizado con el que se almacenan las credenciales en la base de datos.

En el anexo 6. *Seguridad con Apache Shiro*, se presentan ejemplos de configuración.

#### 4.1.2 Capa de negocio

En la capa de negocio se implementa la lógica del sistema, incluyendo las clases que se mapean con las entidades del dominio y los algoritmos de extracción y recomendación. Se definen interfaces para las operaciones del sistema y se utiliza el patrón de diseño denominado Factory [70] para disminuir el acoplamiento de los pedidos de capas superiores.

Además se implementa un manejador de configuraciones que permite acceder a un archivo de propiedades que contiene valores parametrizados utilizados por los algoritmos.

En la Figura 13 se describe la estructura de la capa y las relaciones entre los componentes.

Dentro de los componentes existentes, es importante destacar los Controladores, ya que son los encargados de realizar la extracción y generar las recomendaciones de elementos.

#### 4.1.2.1 Extracción

La extracción de objetos de aprendizaje se lleva a cabo haciendo uso de la herramienta JSoup detallada anteriormente (ver sección 3.1.2). La metodología consiste en descargar la página HTML de la plataforma Moodle que contiene el listado de cursos y procesar el contenido. Para cada enlace a un curso, se accede y se descarga la página HTML conteniendo las actividades y recursos disponibles. Nuevamente iterando en los elementos, se descarga y procesa la información de los objetos, almacenándola en la base de datos del sistema. En el proceso de extracción se descargan los cursos (como grupos), los OA y los usuarios pertenecientes a los cursos (quedando éstos asociados a los cursos). Es importante destacar que todos los elementos extraídos de la plataforma se almacenan con el mismo identificador numérico existente en ella de forma de mantener un mapeo entre elementos de Moodle y el repositorio del sistema de recomendaciones.

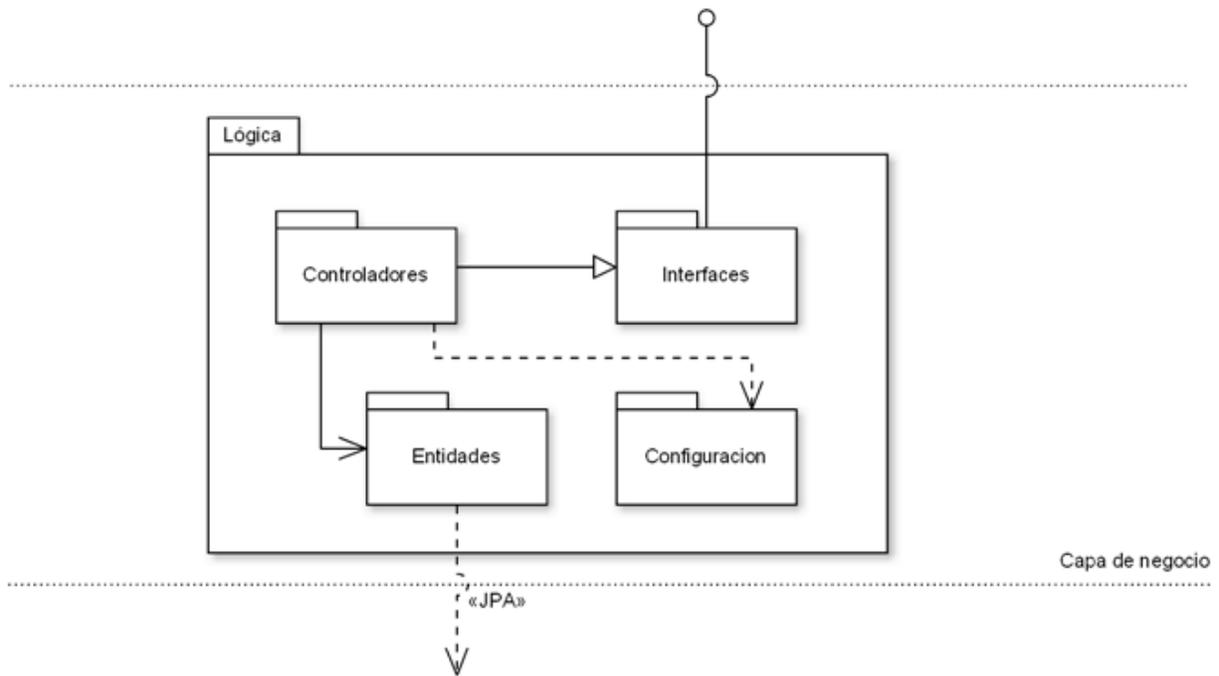


Figura 13 - Diagrama de paquetes de capa de negocio

A continuación se detalla el pseudocódigo del algoritmo de extracción:

```
Descargar página principal de Moodle
Obtener listado de cursos
Para cada curso c en listado
    Descargar página del curso c
    Obtener identificador y nombre del curso c
    Crear un nuevo grupo con los datos del curso c
    Descargar página de estudiantes matriculados al curso
    Obtener listado de estudiantes
```

```

Para cada estudiante e
    Obtener identificador, nombre y correo electrónico del estudiante e
    Almacenar un nuevo usuario con los datos del estudiante e
Obtener listado de Actividades y Recursos
Para cada elemento oa
    Obtener identificador, URL de referencia a Moodle y metadatos
    Almacenar nuevo LOAR con los datos extraídos

```

FIN

#### 4.1.2.2 Recomendación

Para las recomendaciones se implementan tres técnicas: Filtrado Colaborativo basado en usuarios e ítems, y recomendaciones grupales. Como ya se mencionó, para el filtrado colaborativo se utiliza la librería Apache Mahout, la cual provee las operaciones necesarias para llevar a cabo la tarea, simplemente es necesario configurar los algoritmos que se desean aplicar.

En el caso del **filtrado colaborativo basado en usuarios**, se utiliza el algoritmo de “Vecinos más cercanos”. Este algoritmo obtiene los usuarios que tienen patrones similares de puntaje con el usuario activo, y a partir de estos puntajes se hace una predicción. Los usuarios que tienen patrones parecidos de puntaje son denominados vecinos, y se puede asumir que tienen gustos parecidos. Para obtener los vecinos más cercanos, es decir, los usuarios con puntajes más similares, se utiliza la correlación de Pearson. La correlación de Pearson, dado dos usuarios, calcula un número entre 0 y 1, que define la similaridad entre ambos usuarios. Luego se toman N usuarios con mayor correlación para realizar las predicciones. Para más detalles ver la sección 2.3.3.

La implementación de la correlación de Pearson en la librería Mahout es la siguiente:

$$PersonCorrelation(u, v) = \frac{\sum_{i \in I} puntaje(u, i) * puntaje(v, i)}{\sqrt{\sum_{i \in I} puntaje(u, i)^2 * \sum_{i \in I} puntaje(v, i)^2}}$$

Se decide dejar parametrizada la cantidad de “vecinos” a los efectos de poder facilitar la realización de pruebas en un futuro. Por defecto se toma el valor en 3.

En los algoritmos de **filtrado colaborativo basados en los ítems**, se examinan los elementos puntuados por el usuario y se determinan qué tan similares son al ítem objetivo. Luego se recuperan los  $n$  ítems más similares, mientras que al mismo tiempo se calcula el grado de similaridad de cada uno. Para obtener la similaridad entre ítems, al igual que la similaridad entre usuarios, se utiliza la correlación de Pearson. El cálculo es análogo al de similaridad entre usuarios.

Para el diseño del algoritmo de **recomendaciones grupales** se utilizó la técnica de *Divide and Conquer*, técnica que consiste en dividir el problema en varios sub-problemas para facilitar la solución de la misma.

Como ya se vio anteriormente (ver sección 2.3.4) para obtener el conjunto de recomendaciones, se debe calcular para cada ítem que tan útil resultará para el grupo. Para calcular este valor, se tiene en cuenta dos variables: el grado de concordancia y el grado de discrepancia sobre un ítem. El primer aspecto se denomina *group relevance* (relevancia grupal) y el segundo aspecto se denomina *group disagreement* (discrepancia grupal). Cuantos más miembros prefieran un ítem, mayor deberá ser el puntaje de concordancia. En caso contrario, cuantos menos miembros prefieran un ítem, el grado de discrepancia será mayor. A partir de estas dos variables, se calcula la función consenso. La función consenso resulta de combinar estas dos variables ponderadas:

$$F(G, i) = w_1 \times rel(G, i) + w_2 \times (1 - dis(G, i)), w_1 + w_2 = 1$$

Donde G es el grupo de usuarios e i es un ítem.

Para calcular la relevancia grupal se analizaron dos alternativas posibles, una es el promedio de las relevancias individuales, y otra es tomando la relevancia mínima. Se decidió tomar el promedio ya que en nuestro contexto tomar la relevancia mínima resulta muy drástico. Por ejemplo si un ítem es extremadamente relevante para la mayoría de los usuarios de un determinado grupo excepto para uno, queremos que ese ítem sea relevante para el grupo, por lo tanto debemos tomar el promedio como factor de relevancia y no el mínimo. Se define una función *relevanciaGrupal* para un determinado grupo e ítem.

Se consideran los siguientes métodos para determinar esta discrepancia. Un método se conoce como discrepancia par a par promedio y el otro como varianza de discrepancia. En nuestro caso se toma la discrepancia par a par promedio. Queda fuera de alcance y se considera como posible trabajo a futuro evaluar los resultados utilizando la varianza de discrepancia.

Una vez calculada la relevancia grupal y la discrepancia grupal para un cierto ítem, se calcula la función consenso. Al no poseer información, en un principio se decide ponderar de igual forma la relevancia y discrepancia grupal, es decir,  $w_1=w_2=0.5$ . Queda fuera del alcance y se considera como posible trabajo a futuro experimentar con diferentes valores de forma de mejorar los resultados.

Para los cálculos de la relevancia y discrepancia grupal, es necesario calcular para cada usuario perteneciente al grupo la relevancia de cada ítem. Para obtener esta relevancia se utiliza el algoritmo de filtrado colaborativo Slope One [75].

### 4.1.3 Capa de persistencia

Para el desarrollo de la capa de persistencia se utiliza una base de datos relacional implementada con el motor MySQL [76] generada a partir de las entidades del negocio utilizando JPA [77] de EclipseLink [78]. Debido a ello, las tablas presentes en el modelo de

datos se corresponden directamente con las entidades definidas en la capa de negocio (ver sección 4.1.2)

En la Figura 14 se presenta el diagrama que representa dicho modelo.

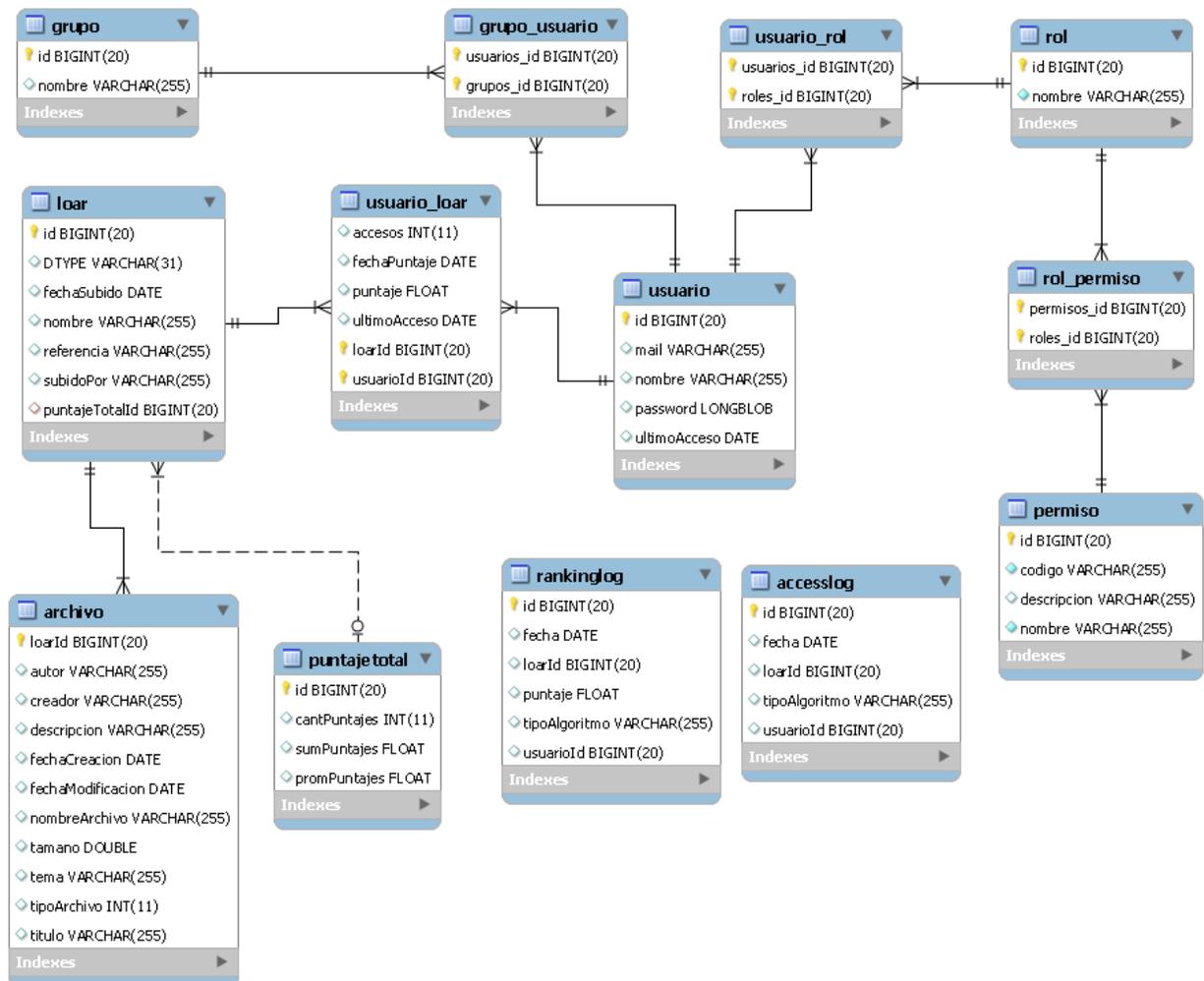


Figura 14 - Diagrama del modelo de datos

En este diagrama se destacan las tablas *Rol*, *Usuario\_Rol*, *Rol\_Permission* y *Permiso* las cuales almacenan la estructura de roles y permisos de un determinado usuario detallada en la sección 4.1.1.1. Se pretende representar una relación N a N entre usuarios y permisos, de forma de un usuario pueda tener asociados N permisos y a su vez, un permiso ser asignado a N usuarios. Los permisos tienen un *código* identificador con el cual son consultados desde la aplicación. Otra tabla con información relevante al sistema de recomendación es *PuntajeTotal*, con ella se pretende llevar la cuenta de todas las valoraciones (o puntajes) que recibe un OA. Si bien esta información es calculable procesando las tuplas de la tabla *RankingLog*, de esta forma se mejora la performance del sistema evitando excesivos cálculos y se permite obtener el puntaje promedio de un OA de manera eficiente.

## 4.2 Recomendaciones integradas a la plataforma Moodle

Con el objetivo de facilitar el acceso de los usuarios a las recomendaciones provistas por el sistema, se desea integrar los resultados de los algoritmos de recomendación implementados en la plataforma Moodle, de forma que los estudiantes pueden recibir sugerencias de objetos de aprendizaje mientras realicen sus actividades de dentro del sitio del curso.

### 4.2.1 Alternativas para la integración

Como ya se explicó antes, para llevar a cabo esta tarea se evaluaron diversas alternativas de acuerdo a las facilidades provistas por Moodle.

Como primera opción, se maneja el desarrollo de un nuevo módulo que permita la integración de ambos sistemas. Si bien Moodle mantiene una estructura modular simplificando la expansión de funcionalidades, se debe implementar el mecanismo de interoperabilidad de acuerdo a los estándares de desarrollo y seguridad [36] además de las consideraciones que se deben tener en cuenta a la hora de desarrollar e instalar un nuevo módulo en un sistema en producción. Por lo tanto se opta por analizar otras alternativas.

Otro camino más interesante desde el punto de vista práctico, es aprovechar las funcionalidades ya provistas por la plataforma, como es el caso del Recurso URL y el Bloque HTML. Por más detalles de estos elementos ver anexo 1. *Descripción de Actividades y Recursos que se pueden incluir en Moodle.*

El Recurso URL permite publicar en el sitio un enlace definido por el docente. El mismo puede contener parámetros, como por ejemplo el identificador del usuario en la plataforma. Esta alternativa permite que el docente ingrese una url que redirija al sistema de recomendaciones, de forma que cuando el estudiante acceda al enlace, el mismo sea llevado a una página que presente las recomendaciones personalizadas. Si bien esta alternativa resuelve fácilmente el problema, se requiere que el estudiante acceda al enlace para recién ahí poder ver las recomendaciones, proceso que en muchos casos lleva a la pérdida del interés de los usuarios por falta de practicidad.

Finalmente, el Bloque HTML permite ingresar un campo de código HTML, que el navegador interpreta como tal. Incluyendo una rutina en JavaScript, se puede extraer el identificador del usuario, realizar un pedido al sistema de recomendaciones *MoodleRecommender* y cargar el listado resultante con los objetos de aprendizaje recomendados dentro del bloque. Al cargarse la página del curso en la plataforma Moodle, se ejecuta dicho código JavaScript y se obtienen las recomendaciones directamente en la pantalla, sin necesidad de ninguna acción por parte del estudiante.

El Bloque HTML es la alternativa más atractiva si se tiene en cuenta la facilidad en contraste con el desarrollo de un nuevo módulo y a la usabilidad respecto al Recurso URL. Sin embargo, el usuario administrador (o docente) que ingrese el bloque debe contar con privilegios específicos que provee la plataforma Moodle para ingresar contenido de tipo JavaScript en los

bloques HTML, o de lo contrario, el sistema no podrá procesar dicho texto como tal y el mismo no será ejecutado, imposibilitando la obtención de recomendaciones.

#### 4.2.2 Solución

Una vez analizados los caminos posibles, se opta por utilizar el Bloque HTML. Para ello resta desarrollar el código JavaScript que obtenga y presente en la pantalla las recomendaciones.

La tecnología propuesta, permite resolver el problema mediante dos vías. Una opción es desarrollar una rutina JavaScript que realice pedidos en segundo plano en forma asíncrona al servidor donde se encuentra instalado el sistema de recomendaciones. Esta técnica se conoce como AJAX (Asynchronous JavaScript And XML) [79]. Dicho script, envía la solicitud de recomendación pasando como parámetro el identificador del usuario, y al obtener la respuesta del sistema de recomendaciones, genera texto HTML conteniendo un listado de los resultados. Con esta alternativa, en el Bloque HTML de Moodle, se debe ingresar tanto el código JavaScript como la estructura de lista vacía, para que luego de obtenidos los resultados la misma se carga de forma dinámica.

La otra opción es utilizar lo que se conoce como un iframe, esto es, un marco que contiene una página de otro sitio. Un iframe es un componente HTML que recibe como parámetros una URL de un sitio y las dimensiones deseadas (entre otros) y presenta el contenido cargado desde la URL ingresada. Entonces, se implementa una rutina JavaScript que obtenga el identificador del usuario y genere una URL hacia una página codificada en el servidor donde se encuentra el sistema de recomendaciones, la cual se encarga de resolver la consulta y desplegar los resultados.

Si bien ambas alternativas son viables, se opta por la segunda ya que minimiza el procesamiento del lado del cliente debido a que la rutina JavaScript (código que se ejecuta en el navegador del usuario) simplemente debe generar la URL, mientras que la primera opción requiere realizar un pedido y procesar la respuesta del sistema de recomendaciones.

La Figura 15 - Pantalla de recomendación integrada a Moodle se muestra la página del curso Fundamentos de la Web Semántica en el sitio del EVA. En la parte superior derecha se encuentran las recomendaciones generadas por el módulo de recomendaciones. Si se selecciona uno de los OA recomendados, se abre una nueva pestaña en el navegador mostrando el detalle del OA (ver Figura 23) en el sitio *MoodleRecommender*.

En la Figura 16 se muestra específicamente el módulo de recomendaciones integrado a la plataforma Moodle. Allí se puede ver que el módulo generó tres recomendaciones al estudiante, a las que se puede acceder mediante el link.

**Facultad de Ingeniería**  
 Área de Tecnologías y Ciencias de la Naturaleza y el Habitat  
**ProEVA**  
 Programa de Entornos Virtuales de Aprendizaje  
 Universidad de la República Uruguay

Usted se ha identificado como Juan Gabriel Ibañez Gonnet (Salir)

Página Principal » Mis cursos » Institutos » Computación » fuwebsem

**Administración**  
 Administración del curso  
 Darme de baja en fuwebsem  
 Calificaciones  
 Ajustes de mi perfil

**Navegación**  
 Página Principal  
 Área personal  
 Páginas del sitio  
 Mi perfil  
 Curso actual  
**fuwebsem**  
 Participantes  
 BIENVENIDOS AL CURSO FUNDAMENTOS DE LA WEB SEMANTICA  
 Problemas con la web tradicional

**BIENVENIDOS AL CURSO FUNDAMENTOS DE LA WEB SEMANTICA**  
**ATENCIÓN:**  
 Este curso no tiene clases presenciales tradicionales.  
 Los estudiantes interesados en realizar este curso deben automatizarse para poder participar de las actividades del curso en esta plataforma.  
 Por consultas dirigir correo a:  
 Regina Motz [rmotz@fing.edu.uy](mailto:rmotz@fing.edu.uy)  
 Edelweis Rohrer [erohrer@fing.edu.uy](mailto:erohrer@fing.edu.uy)  
 Descripción de la propuesta  
**Noticias:**  
 19 de agosto: inician actividades en el espacio de trabajo EVA.  
 RESULTADOS FINALES  
**FOROS:**  
 Novedades

**Recomendaciones**  
 - [Fundamentos de la lógica descriptiva](#)  
 - [Introducción a RDF](#)  
 - [Artículo sobre conceptos básicos de metadatos](#)

**Últimas noticias**  
 Regina Motz 7 de mar, 14:47 RESULTADOS FINALES DEL CURSO  
 Regina Motz 5 de mar, 05:37 Notas finales  
 Libertad Tansini 19 de dic, 19:22 Recomendación de materiales  
 Edelweis Rohrer 17 de dic, 08:30 PRUEBA FINAL DEL CURSO  
 Edelweis Rohrer 14 de dic, 15:38 Transparencias Encuentro de Cierre  
 Temas antiguos ...

Figura 15 - Pantalla de recomendación integrada a Moodle

**Recomendaciones**  
 - [Fundamentos de la lógica descriptiva](#)  
 - [Introducción a RDF](#)  
 - [Artículo sobre conceptos básicos de metadatos](#)

**Últimas noticias**  
 Regina Motz 7 de mar, 14:47 RESULTADOS FINALES DEL CURSO  
 Regina Motz 5 de mar, 05:37 Notas finales  
 Libertad Tansini 19 de dic, 19:22 Recomendación de materiales  
 Edelweis Rohrer 17 de dic, 08:30 PRUEBA FINAL DEL CURSO  
 Edelweis Rohrer 14 de dic, 15:38 Transparencias Encuentro de Cierre  
 Temas antiguos ...

Figura 16 - Zoom de pantalla de recomendación integrada a Moodle

## 5 Puesta en producción

### 5.1 Instalación

Una vez finalizado el proceso de desarrollo se procede a instalar la aplicación en un servidor, con el fin de brindar los servicios a los usuarios a través de internet. Como primer alternativa se estudia la posibilidad de contratar un servidor pago en el mercado o de utilizar algún servicio gratuito disponible en la web. Luego, surge la posibilidad de utilizar como servidor un pc dentro del Instituto de Computación (INCO) y así poder aprovechar la infraestructura de red de la Facultad de Ingeniería (FIng) con los mecanismos de seguridad que esta provee. En coordinación con la Unidad de Recursos Informáticos (URI) de la FIng se agrega el servidor a la red interna del INCO y luego mediante configuraciones en el firewall de la Facultad se realiza la redirección de los accesos de usuarios externos (estudiantes o docentes) hacia el servidor (en la red interna).

Luego de configurada la conexión que permite el acceso al sistema a través de internet, se procede a instalar el servicio. Para ello se crea una máquina virtual utilizando la plataforma de virtualización VirtualBox de Oracle [80]. Como software de base se utiliza el sistema operativo libre Debian GNU/Linux [81] en su versión i386. Como servidor de base de datos se utiliza MySQL versión 5.6 el cual se instala localmente en el mismo host. Y como servidor de aplicaciones se utiliza Apache Tomcat versión 7.0.42 [82].

Una vez operativo el sistema se procede a realizar la extracción de OA del curso de Fundamentos de la Web Semántica. Previo a la extracción se les comunica a los estudiantes del curso que se almacenará en la base de datos del sistema información personal como nombre, apellido y correo electrónico, pero solamente con la finalidad de realizar las recomendaciones personalizadas.

Para dar por finalizada la instalación del sistema, falta ingresar el script en el Bloque HTML que permite obtener las recomendaciones integradas con la plataforma Moodle (como se explicó en la sección 4.2.2). Cabe destacar que este último paso lo debe realizar un usuario que cuente con los privilegios necesarios para ingresar contenido de este tipo.

### 5.2 Pruebas de performance

Luego de instalar la aplicación en el servidor, pero previo a poner en funcionamiento el sistema con usuarios reales, es conveniente realizar pruebas de performance. Las pruebas de performance son importantes ya que validan que el sistema funcione correctamente en condiciones extremas. También pueden servir para validar y verificar otros atributos del sistema, tales como escalabilidad, fiabilidad y uso de los recursos.

Hay distintos tipos de pruebas: pruebas de carga, pruebas de estrés, prueba de estabilidad y pruebas de picos. Las pruebas de carga es el tipo más sencillo de hacer y consiste en observar el comportamiento de una aplicación bajo una cantidad de peticiones esperada. Esta carga puede ser el número esperado de usuarios concurrentes utilizando la aplicación y que realizan un número específico de transacciones durante el tiempo que dura la carga. Las pruebas de estrés se utilizan normalmente para hacer que la aplicación falle. Se va duplicando el número de usuarios que se agregan a la aplicación y se ejecuta una prueba de carga hasta que se rompe. Este tipo de pruebas se realizan para determinar la robustez de la aplicación en los momentos de carga extrema. Las pruebas de estabilidad son realizadas para determinar si la aplicación puede soportar una carga esperada en forma continua. Las pruebas de picos tratan de observar el comportamiento del sistema variando el número de usuarios, tanto cuando bajan, como cuando tiene cambios drásticos en su carga.

Para realizar estas pruebas se utiliza la herramienta JMeter [83] . Esta herramienta es utilizada para analizar y medir el desempeño de una variedad de servicios, con énfasis en aplicaciones web. Para crear un plan de pruebas, JMeter permite simular la navegación de un sitio web de dos formas posibles. La primera es crear la navegación manualmente, es decir, crear las peticiones HTTP que se realizan en una determinada transacción. La otra alternativa es configurar JMeter para grabar una navegación realizada a través de un *browser* (navegador web). Se optó por la segunda alternativa por ser más sencilla.

El escenario que se escogió para ser grabado con JMeter es el de ingreso de un usuario a la aplicación, luego realizar una búsqueda y finalmente ingresar a la página de recomendaciones personalizadas. Esta transacción se reprodujo con 250 usuarios concurrentes ya que es la cantidad aproximada de estudiantes matriculados al curso de Moodle.

Los resultados obtenidos se detallan en la Tabla 3.

<b>Acción</b>	<b>Media(ms)</b>	<b>Min(ms)</b>	<b>Max(ms)</b>	<b>% Error</b>	<b>Rendimiento</b>	<b>Kb/seg</b>
Ingreso al sitio	275	9	3271	0.0	17.92	160.24
Realizar búsqueda	5718	124	12474	0.0	9.74	134.92
Obtener recomendaciones	17891	2305	34993	0.0	6.89	28.61
<b>Total</b>	<b>5405</b>	<b>9</b>	<b>34993</b>	<b>0.0</b>	<b>26.36</b>	<b>209.80</b>

Tabla 3 - Resultados de pruebas de performance

La columna *Acción* describe la acción realizada en el sitio. Como se mencionó anteriormente, el escenario escogido fue: ingresar al sitio, realizar una búsqueda y luego obtener recomendaciones. La columna *Media* indica el tiempo medio en milisegundos invertido por la petición. La columna *Min* y *Max* indican el tiempo mínimo y máximo en milisegundos invertido en la petición. La columna *%Error* indica el porcentaje de las peticiones que fallaron, ya sea por error de conexión o lentitud de respuesta por parte del servidor. La columna *Rendimiento* y

*Kb/seg* indican la cantidad de peticiones y *Kb* respectivamente que es capaz de procesar el servidor por unidad de tiempo (segundos).

La aplicación tuvo un rendimiento aceptable ante la prueba de carga realizada simulando 250 usuarios utilizando la misma concurrentemente. Donde se notó más demora en las respuestas fue a la hora de obtener las recomendaciones, con una media de 34993 milisegundos, esto se debe a que el servidor debe ejecutar los algoritmos de recomendación que manejan mucha información, lo cual resulta razonable que el tiempo de respuesta sea más elevado que las demás peticiones. Lo importante de estos resultados es que la aplicación fue capaz de soportar 250 usuarios concurrentes con un porcentaje de error de 0%, es decir, todas las peticiones fueron respondidas exitosamente.



## 6 Evaluación de algoritmos

En esta sección se analizan diferentes métricas para evaluar la eficacia de los algoritmos de recomendación y se exponen los resultados obtenidos en un experimento con estudiantes de la Facultad de Ingeniería.

### 6.1 Métricas

Una vez desarrollado el sistema de recomendaciones, es necesario realizar pruebas para validar la efectividad de los algoritmos generadores de las recomendaciones.

Existen diversas técnicas de evaluación [84], las cuales tienen en cuenta el propósito de dicha recomendación (venta de productos, sugerencia de películas, etc.), fundamental a la hora de concluir los resultados. De acuerdo a la literatura clásica, los mecanismos de evaluación denominados como evaluación de algoritmos “offline” (o fuera de línea) son los más importantes. Los mismos son muy utilizados para verificar el comportamiento de los algoritmos antes de implantarlos en producción con usuarios reales y para minimizar los costos de resolver problemas ocasionados por posibles fallas en el futuro. La evaluación offline también es muy beneficiosa para obtener comparaciones objetivas de distintos algoritmos fácilmente reproducibles. También se encuentran las evaluaciones denominadas “online” que consisten en hacer un experimento con usuarios activos. Se brinda a un determinado grupo de usuarios un sistema y se monitorea su comportamiento. Se puede evaluar la performance, satisfacción, participación y otras métricas.

Antes de comenzar a detallar las técnicas de evaluación offline, es necesario manejar el concepto de “data set” o conjunto de datos. Un data set es como bien dice su nombre, es una colección de datos utilizada para realizar la recomendación. Dentro de estos datos se encuentran los puntajes de los usuarios sobre los ítems, y opcionalmente pueden contar con información adicional de los usuarios y/o los ítems.

Para medir la eficacia de los algoritmos de recomendación se utilizan distintas métricas. En términos generales, las métricas se clasifican en tres grupos:

- Predictive accuracy
- Classification accuracy
- Rank accuracy

#### 6.1.1 Predictive accuracy

Estas métricas miden qué tan cerca está la predicción del sistema de recomendaciones con el puntaje real del usuario y son importantes para evaluar sistemas donde la predicción de cierto objeto se muestra al usuario.

Una métrica de este tipo se conoce como Mean Absolute Error (MAE) y computa la desviación promedio absoluta entre la predicción y el puntaje real.

$$|E| = \frac{\sum_{i=1}^N |p_i - r_i|}{N}$$

Otras métricas relacionadas con MAE son: Mean Squared Error (MSE), Root Mean Squared Error (RMSE) y Normalized Mean Absolute Error (NMAE). Las primeras dos variantes elevan al cuadrado el error antes de sumarlo. Root Mean Squared Error se calcula como:

$$|E| = \sqrt{\frac{1}{N} \sum_{i=1}^N (p_i - r_i)^2}$$

El objetivo es penalizar más cuando mayor es la diferencia entre la predicción y el puntaje real. Por ejemplo un error de 1 incrementa la sumatoria en 1, mientras que una diferencia de 2, incrementa la sumatoria en 4. La métrica NMAE es igual a MAE pero normalizando el rango de puntajes.

### 6.1.2 Classification accuracy

Este tipo de métrica mide la frecuencia en la que un sistema de recomendaciones toma una decisión correcta sobre la utilidad de un objeto y es apropiada en sistemas donde la preferencia de los usuarios sobre un determinado objeto es binaria, es decir, le resulta útil o no. Las métricas más populares de este estilo son: Precision y Recall.

Classification accuracy consiste en separar el conjunto de ítems en dos clases; relevantes y no relevantes. En caso que la puntuación de los ítems no sea binaria, por ejemplo sea de una escala del 1 al 5, es necesario dividir esta escala en dos, por ejemplo los ítems puntuados del 1 a 3 serán los no relevantes y los puntuados en 4 o 5 serán los relevantes. También es necesario separar el conjunto de ítems en dos, el conjunto de ítems retornados por el algoritmo de recomendación y el conjunto complemento del conjunto anterior. Se define entonces la Precisión como el cociente del conjunto de los ítems relevantes retornados y el conjunto retornado. Y se define el Recall como el cociente del conjunto de los ítems relevantes retornados y el conjunto de los ítems relevantes.

$$precision = \frac{|\{itemsRelevantes\} \cap \{itemsRecomendados\}|}{|\{itemsRecomendados\}|}$$

$$recall = \frac{|\{itemsRelevantes\} \cap \{itemsRecomendados\}|}{|\{itemsRelevantes\}|}$$

### 6.1.3 Rank accuracy

Este tipo de métricas miden que tan certero es un algoritmo de recomendación a la hora de recomendar una lista de objetos ordenados según la preferencia de un determinado usuario. Al contrario de los tipos de métricas vistos anteriormente, las métricas de ranqueo son más apropiadas para evaluar algoritmos donde se presenta al usuario una lista ranqueada, en dominios donde la preferencia del usuario no es binaria.

Rank accuracy puede resultar no del todo adecuado en dominios donde el usuario sólo califique los objetos como relevantes o no relevantes (puntuación binaria), ya que al usuario no le va a interesar una lista con los objetos relevantes primero, y luego los no relevantes, ya que por ejemplo aunque el sistema recomiende el top 10 con todos los objetos relevantes para el usuario, quizás el más relevante para el usuario esté en el puesto 10. En estos casos se dice que la métrica resulta demasiado sensible.

## 6.2 Selección de datos para la evaluación

Hay decisiones claves que se deben tomar a la hora de realizar la evaluación de las recomendaciones. Es importante saber si las evaluaciones pueden ser llevadas a cabo offline, a través de usuarios activos o una combinación de ambas.

Cuando se quiere evaluar un algoritmo en el contexto de recomendar nuevos objetos, sería inapropiado realizar únicamente un análisis offline ya que al recomendar un objeto que el usuario no ha puntuado, no hay forma de evaluar si la recomendación fue buena o mala. En las evaluaciones offline, el algoritmo es utilizado para predecir ciertos valores de un conjunto de datos y los resultados son analizados usando una o más métricas vistas anteriormente. Estas evaluaciones tienen la ventaja de que son rápidas y prácticas a la hora de realizar en grandes conjuntos de datos, en diferentes conjuntos de datos, o con distintos algoritmos. Cuando el conjunto de datos incluye timestamps, es posible hacer una reproducción de la serie de puntajes y recomendaciones offline, como si fuera un uso real, es decir, cada vez que se hace una puntuación, primero se hace una predicción del ítem con todas las puntuaciones obtenidas hasta el momento, ese valor predicho se compara con la puntuación real y se hace la evaluación, y así sucesivamente. El análisis offline tiene una debilidad importante y es que la esparsidad de los puntajes de los ítems limita el conjunto de los ítems que pueden ser evaluados. No se puede evaluar la efectividad de la recomendación de un ítem para un usuario si no se tiene un puntaje de ese usuario para ese ítem en el conjunto de datos.

## 6.3 Evaluación de algoritmos con Apache Mahout

Mahout permite realizar una evaluación de las recomendaciones generadas [85]. Para realizar la evaluación, se toma un subconjunto de las preferencias de los usuarios como datos de prueba y toma el complemento de este subconjunto como datos de entrenamiento. Generalmente se toma un porcentaje grande como datos de entrenamiento, aproximadamente entre el 80-90%, ya que cuantos más datos se tenga sobre las preferencias de los usuarios,

más precisas resultan las recomendaciones. Con estos datos, Mahout genera preferencias para elementos pertenecientes al conjunto de datos de prueba y los compara con la preferencia real. Para realizar las comparaciones, Mahout provee dos tipos de métricas: Average Absolute Difference Evaluator y Root Mean Square Evaluator.

La métrica Average Absolute Difference calcula la diferencia promedio entre la preferencia real y la obtenida por el algoritmo de recomendación. Cuanto más bajo sea este valor, mejores son las recomendaciones. Un valor bajo significa que la preferencia estimada difiere muy poco de la preferencia real. Si el valor fuera 0, la preferencia estimada y la real son iguales, lo que significa que las recomendaciones son perfectas.

La métrica Root Mean Square calcula la diferencia como la raíz cuadrada del promedio de las diferencias entre la preferencia real y la estimada al cuadrado. Al igual que la métrica anterior, un valor relativamente bajo, indica que las recomendaciones son buenas.

## 7 Resultados

El sistema fue puesto a prueba con 27 estudiantes que cursaban la asignatura Fundamentos de la Web Semántica de la carrera Ingeniería en Computación de la Facultad de Ingeniería. Esta asignatura sólo puede ser cursada por estudiantes avanzados en la carrera ya que requiere tener aprobadas ciertas materias. De este curso, se extrajo la totalidad de archivos con formato pdf, resultando en una extracción de 18 documentos. Los mismos contienen diapositivas, papers, apuntes, y artículos. Estos materiales fueron revisados y publicados por el docente de la asignatura, por lo que se puede afirmar que la calidad del contenido es buena. Sin embargo, en cuanto a la calidad de los metadatos, no se puede realizar tal afirmación. En un breve relevamiento de la información extraída, se detecta que la mayoría de los documentos no posee ningún metadato o sólo cuentan con autor y descripción del archivo. Esto puede afectar negativamente la eficacia de la búsqueda de OA a través de metadatos. En el Anexo 9. *Detalle de OA extraídos del curso Fundamentos de la Web Semántica* se detallan los recursos extraídos junto con sus metadatos. Además de los metadatos incluidos en la tabla, el sistema soportaba otros metadatos como temática, fecha que se subió el recurso y por quién se subió, pero no fueron incluidos en la tabla ya que ningún recurso poseía estos metadatos. La calidad de los metadatos no es buena ya que hay muchos recursos que no disponen de algunos metadatos, y otros cuyos metadatos están completos pero su contenido es de mala cantidad.

En esta situación, el motor de búsqueda, en un ambiente más amplio, puede que no sea aprovechado de forma óptima. Para lograr un uso adecuado del motor, es necesario que docentes y estudiantes suban material a Moodle incluyendo la mayor cantidad de metadatos posibles, así como contribuyendo al aporte de metadatos en los recursos ya existentes en Moodle.

En cuanto a las recomendaciones, cada estudiante puntuó en promedio 4 ítems. Los ítems podían ser calificados con un puntaje de 1 a 5 según la preferencia del estudiante. Se obtuvieron 83 calificaciones de diferentes usuarios sobre distintos ítems, donde sólo 19 calificaciones fueron menores o iguales a 3. Lo que significa que más del 75% de las calificaciones fueron de 4 y 5, lo cual es muy positivo. Analizando las recomendaciones con las métricas provistas por Mahout, se obtuvieron los resultados expuestos en la Tabla 4.

Métrica \ Algoritmo	Basado en ítems	Basado en usuarios
<i>Precisión</i>	0.83	1.0
<i>Recall</i>	0.66	0.66
<i>MAE</i>	1.78	<b>0.15</b>
<i>RMS</i>	<b>1.84</b>	1.18

Tabla 4 - Resultados obtenidos con las distintas métricas

Para el algoritmo de filtrado colaborativo basado en ítems, se obtuvo una precisión de 0.83 y un Recall de 0.66. Lo que indica que el 83% de los ítems recomendados por el algoritmo resultan relevantes para el usuario, y el 66% de los ítems calificados por el usuario resultaron relevantes para el mismo. En una postura conservadora, los ítems calificados entre 1 y 3 son considerados como ítems no relevantes y los ítems calificados entre 4 y 5 como relevantes.

Utilizando la métrica de Mean Average Error, la evaluación dio 1.78. Lo que significa que en promedio, este algoritmo de recomendación predice el puntaje de un cierto usuario a un cierto ítem con una precisión de  $\pm 1.78$ , lo cual no resulta muy favorable en un entorno donde el rango de puntajes va de 1 a 5. Utilizando la métrica Root Mean Square, la evaluación resultó ser 1.84. Esta métrica, como se vio anteriormente, cuanto más amplia sea la diferencia entre el puntaje estimado y el real, mayor es la penalización, por lo que es razonable que se obtenga un valor mayor. Para el algoritmo de filtrado colaborativo basado en usuarios, se obtuvo una Precisión de 1.0, lo que indica que el 100% de los ítems recomendados al usuario, resultaron ser relevantes. El Recall resultó ser 0.66. Utilizando la métrica de Mean Average Error, la evaluación dio 0.15 y utilizando la métrica Root Mean Square, la evaluación resultó ser 1.18.

Según estos resultados, el algoritmo de recomendación basado en usuarios tuvo mejor rendimiento con todas las métricas propuestas que el algoritmo basado en ítems. La diferencia de eficacia es más notoria con la métrica MAE, con esta métrica el algoritmo basado en ítems predijo puntuaciones con una diferencia en promedio de 1.78 de la puntuación real, mientras que el basado en usuarios esta diferencia es de 0.15, lo que significa que las predicciones son casi perfectas. El peor resultado se obtuvo con la métrica Root Mean Square en el algoritmo basado en ítems, donde se obtuvo un resultado de 1.84, si bien es una diferencia importante, hay que tener en cuenta que esta métrica penaliza más que la métrica MAE la diferencia entre el puntaje real y el predicho.

A pesar de que los números indiquen que el algoritmo basado en usuarios sea más adecuado que el basado en ítems, no se puede garantizar que este algoritmo tenga mejor performance en todos los escenarios posibles en un ambiente educativo, ya que el estudio se realizó en un escenario reducido, con una cantidad muy acotada de usuarios y OA. Se deja como trabajo futuro, ya que escapa del alcance del proyecto, la utilización de este sistema en un ambiente más extenso, con más usuarios y OA, de forma de poder determinar si el algoritmo de filtrado colaborativo basado en usuarios sigue comportándose más eficientemente que el basado en ítems.

Continuando con el análisis, en la Figura 17 se representa gráficamente la evolución de los puntajes otorgados por los usuarios a los OA del sistema, agrupados por fecha. Si bien al comienzo los valores son un poco erráticos (puntajes 2, 3, 4 y 5), a medida que fue pasando el tiempo y los usuarios continuaron calificando elementos, se puede apreciar un crecimiento en los puntajes otorgados, tendiendo los valores máximos, 4 y 5. De esto se puede concluir que a medida que más usuarios ingresan calificaciones, más precisas son las recomendaciones generadas.

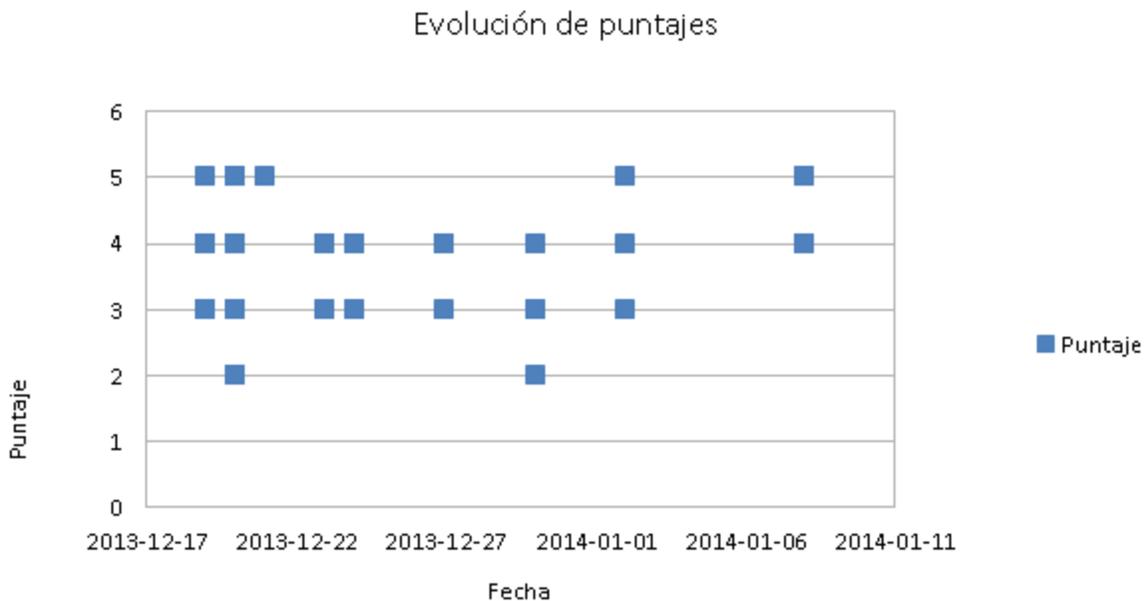


Figura 17 - Gráfica de evolución de puntajes en el tiempo

En la Figura 18 se representa gráficamente la evolución de los puntajes en función del tiempo, pero de un OA determinado. Para este análisis se seleccionó el OA con mayor cantidad de puntajes con el fin de obtener resultados más precisos. Aquí también se puede apreciar el mismo patrón que antes. Al comienzo, los valores son dispersos pero se nota una tendencia creciente a medida que avanza el tiempo y los estudiantes continúan ingresando calificaciones.

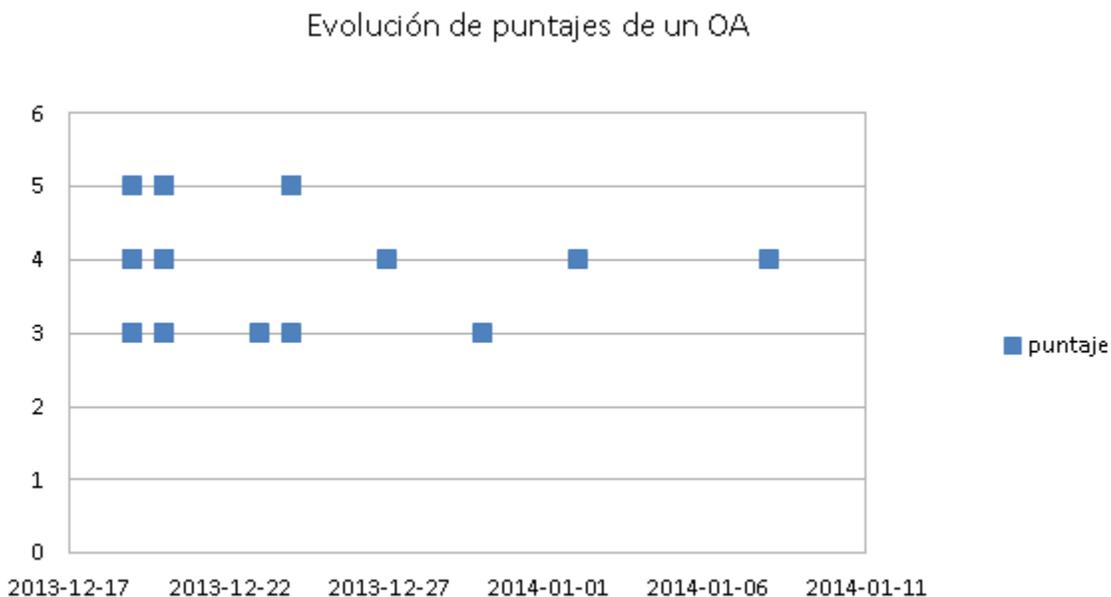


Figura 18 - Gráfica de evolución de puntajes de un OA en el tiempo



## 8 Conclusiones y trabajos a futuro

En este capítulo se presentan las conclusiones del proyecto y finalmente se mencionan posibles trabajos a futuro que pueden complementar este trabajo.

### 8.1 Conclusiones

Resulta evidente afirmar que la plataforma es una herramienta de gran utilidad para la educación. Desde 2008 se encuentra operativa y la cantidad de cursos ha ido creciendo exponencialmente a tal punto que la cantidad de OA en la plataforma alcanza un número elevado. Como consecuencia de esto, muchas veces no resulta sencillo encontrar el material deseado, lo que implica una pérdida de tiempo por parte de los estudiantes en la búsqueda.

El objetivo de este trabajo fue contar con un sistema que mediante la integración con Moodle mitigue el problema antes mencionado, es decir, que facilite el acceso a los recursos y que a su vez el sistema recomiende elementos a los estudiantes, contribuyendo con la mejora del proceso de aprendizaje. Para lograr este objetivo, se llevaron a cabo cuatro tareas. La primera tarea fue desarrollar un módulo de extracción de OA que permita extraer el material publicado en el curso y los usuarios inscriptos al mismo, ya que esta información se utiliza como datos de entrada para los algoritmos de recomendación. Luego se desarrolló un módulo de recomendación de OA, encargado de brindar las recomendaciones a los estudiantes en base a sus perfiles. La tercer tarea consistió en el desarrollo de un motor de búsqueda de OA, el cual permite buscar material utilizando los metadatos de los mismos, y por último, un sitio web que integre estos tres módulos, donde el estudiante puede registrarse e ingresar al sitio para buscar y obtener recomendaciones.

Estas tareas se concretaron satisfactoriamente, logrando así brindar recomendaciones personalizadas de forma efectiva a los estudiantes y docentes. El sistema fue puesto a prueba con alumnos del curso Fundamentos de la Web Semántica de la Facultad de Ingeniería. Más del 80% de las recomendaciones realizadas fueron puntuadas con un puntaje entre 4 y 5, donde 5 era el máximo posible, lo que significa que el sistema tuvo un impacto muy positivo en un ambiente acotado, donde participaron de las pruebas 27 estudiantes.

Se utilizaron diferentes métricas de evaluación para medir la eficacia de los algoritmos. Como se vio en la sección 7, el algoritmo de FC basado en usuarios tuvo mejor rendimiento que el basado en ítems, debido a que la cantidad de usuarios e ítems es similar. En ambientes donde la cantidad de usuarios es mucho mayor a la de ítems, es de esperar un comportamiento opuesto, obteniendo mejores resultados mediante FC basado en ítems.

Utilizando la métrica MAE se dio la diferencia más notoria. Con esta métrica, el algoritmo basado en ítems obtuvo un valor de 1.78, mientras que el algoritmo basado en usuarios obtuvo un valor de 0.15. Esto significa que la diferencia promedio entre el puntaje real y el puntaje

predicho, es de 1.78 en el algoritmo basado en ítems y 0.15 en el algoritmo basado en usuarios. Esta última es una diferencia extremadamente pequeña, lo que significa que las puntuaciones predichas por el algoritmo de FC basado en usuarios son muy acertadas, es decir, cercanas a las reales. En general, el resultado de la evaluación de las recomendaciones se puede considerar muy satisfactorio. A pesar de que un algoritmo sea más eficiente que otro, no se puede garantizar que este tenga mejor performance en todos los escenarios posibles en un ambiente educativo, ya que el estudio se realizó en un escenario acotado, con pocos OA y usuarios.

Vistos los resultados de este trabajo, se concluye que es totalmente viable el uso de un sistema de recomendación en un entorno de EVA más amplio, con más cursos, usuarios y OA. Contar con recomendaciones suficientemente precisas en el EVA contribuye favorablemente el proceso de aprendizaje ya que las mismas permiten a los estudiantes acceder fácilmente a materiales que les resulten de interés y que al momento desconocieran.

El sistema es fácilmente integrable con Moodle, cualquier docente, sin necesidad de contar con conocimientos informáticos puede incluir esta herramienta a su curso virtual. El mismo puede llegar a aportar grandes beneficios a los estudiantes que lo utilicen, como el rápido acceso al material deseado y la sugerencia de recursos que pueden resultar de interés.

## 8.2 Trabajos futuros

A lo largo de este trabajo surgieron muchas alternativas para mejorar algunos aspectos del sistema y mejoras que pueden complementar el mismo pero por cuestiones de tiempo y recursos quedan fuera del alcance del proyecto y se describen en esta sección como trabajos futuros.

Un posible trabajo a futuro es extender el módulo de extracción para descargar todos los tipos actividades de Moodle, tales como ejercicios, libros, cuestionarios, foros, etc. ya que actualmente el sistema sólo extrae archivos de tipo PDF. También se puede mejorar la extracción de metadatos tanto en cantidad como en calidad. De acuerdo a lo detallado en la sección 2.2.2, se han realizado trabajos enfocados en esa área con resultados positivos, por lo que se podría reutilizar dichos proyectos para aprovechar la mejora en la extracción de metadatos.

En cuanto a las recomendaciones, en una primera instancia se evalúa el uso de técnicas basadas en Friendsourcing, las que luego fueron descartadas debido a que estas requieren de algún tipo de relación de amistad entre los estudiantes, inexistente en la plataforma Moodle. Es por ello que se considera como un trabajo a futuro la integración del módulo de recomendación con alguna red social, por ejemplo Facebook, para obtener dicha información de amistad.

También resulta interesante utilizar alguna de las técnicas detalladas en la sección 2.3.3, como los basados en la información demográfica de los usuarios, en el contenido, en la utilidad o en el conocimiento.

Otro trabajo a futuro de potencial interés, es realizar un análisis más detallado sobre los valores asignados a los parámetros que reciben los algoritmos de recomendación implementados, tales como “*N vecinos más cercanos*” en FC o la “*función de consenso*” en las recomendaciones grupales, ambos detallados en la sección 4.1.2.2. Experimentando con diferentes valores para cada parámetro es posible obtener mejores resultados en las recomendaciones.

Por último, realizar pruebas de los algoritmos en un entorno con mayor cantidad de usuarios y OA permitiría mejorar las conclusiones a la hora de determinar mediante que técnicas (o combinación de técnicas) es posible obtener mejores recomendaciones en un ámbito educativo.



# Bibliografía

1. Moodle. [En línea] [Citado el: 05 de 06 de 2014.] <https://moodle.org>.
2. Universidad de la República. [En línea] [Citado el: 03 de 05 de 2014.] <http://www.universidad.edu.uy/>.
3. Entorno Virtual de Aprendizaje. [En línea] [Citado el: 03 de 05 de 2014.] <https://eva.fing.edu.uy/>.
4. **Wiley, D. A.** *Connecting learning objects to instructional design theory*. Utah, Estados Unidos : Utah State University, 2000.
5. Facultad de Ingeniería. [En línea] [Citado el: 03 de 05 de 2014.] <http://www.fing.edu.uy/>.
6. **National Information Standards Organization.** *Understanding Metadata*. Bethesda, Estados Unidos : NISO Press, 2004. ISBN: 1-880124-62-9.
7. **Sarwar, B., y otros.** *Item-Based Collaborative Filtering Recommendation Algorithms*. Minnesota, Estados Unidos : Proceeding WWW '01 Proceedings of the 10th international conference on World Wide Web, 2001.
8. **Bernstein, M. S., y otros.** *Personalization via Friendsourcing*. Nueva York, Estados Unidos : Journal ACM Transactions on Computer-Human Interaction (TOCHI), 2010.
9. **Gartrell, M., y otros.** *Enhancing Group Recommendation by Incorporating Social Relationship Interactions*. Colorado, Estados Unidos : Proceeding GROUP '10 Proceedings of the 16th ACM international conference on Supporting group work, 2010.
10. Facebook. [En línea] [Citado el: 08 de 05 de 2014.] <https://www.facebook.com/>.
11. Amazon. [En línea] [Citado el: 08 de 05 de 2014.] <http://www.amazon.com/>.
12. LastFM. [En línea] [Citado el: 08 de 05 de 2014.] <http://www.lastfm.es/>.
13. MySpace. [En línea] [Citado el: 08 de 05 de 2014.] <https://myspace.com/>.
14. **Howe, J.** *The rise of crowdsourcing*. Nueva York, Estados Unidos : Wired Magazine, 2006.
15. **Dahl, J. y Nygaard, K.** *SIMULA: an ALGOL-based simulation language*. Oslo, Noruega : Magazine Communications of the ACM, 1966.

16. **Learning Technology Standards Committee of the IEEE.** *Draft Standard for Learning Object Metadata.* Nueva York, Estados Unidos : IEEE Standards Department, 2002.
17. **Weller, M., Pegler, C. y Mason, R.** *Putting the pieces together: What working with learning objects means for the educator.* Edinburgo, Escocia : Proceedings of the Second eLearnInternational World Summit, Edinburgh International Conference Centre, 2003.
18. **Morales, E. y García, F.** *Valoración de la calidad de unidades de aprendizaje.* Murcia, España : Revista de Educación a Distancia, Universidad de Murcia, 2005.
19. **The JORUM+ Project Teams at EDINA and MIMAS.** *The JISC Online Repository for [learning and teaching] Materials.* 2004.
20. **García, L.** *¿Por qué va ganando la educación a distancia?* Madrid, España : UNED, 2009. ISBN: 978-84-362-5879-0.
21. Real Academia Española. [En línea] [Citado el: 08 de 05 de 2014.] <http://www.rae.es/>.
22. XML. W3C. [En línea] [Citado el: 06 de 05 de 2014.] <http://www.w3.org/standards/xml/core>.
23. LOM-ES. [En línea] [Citado el: 05 de 05 de 2014.] <http://educalab.es/intef/tecnologia/recursos-digitales/lom-es>.
24. Dublin Core. [En línea] [Citado el: 05 de 05 de 2014.] <http://dublincore.org/>.
25. SCORM. [En línea] [Citado el: 05 de 05 de 2014.] <http://scorm.com/>.
26. Advanced Distributed Learning. [En línea] [Citado el: 03 de 07 de 2014.] <http://www.adlnet.org/>.
27. **López, C. y García, F. J.** *Repositorios de Objetos de Aprendizaje: bibliotecas para compartir y reutilizar recursos en los entornos e-learning.* México DF, México : Biblioteca Universitaria de la Universidad Autónoma de Mexico, 2006.
28. **Ministry of Education of Canada.** *White Paper for a Learning Object Repository.* 2001.
29. **Daniel, B.K. y Mohan, P.** *A model for evaluating learning objects.* Sakatchewan, Canadá : Advanced Learning Technologies, 2004. Proceedings. IEEE International Conference., 2004.
30. GLOBE. [En línea] [Citado el: 05 de 05 de 2014.] <http://globe-info.org/>.
31. MERLOT. [En línea] [Citado el: 05 de 05 de 2014.] <http://www.merlot.org/merlot/index.htm>.
32. ARIADNE. [En línea] [Citado el: 05 de 05 de 2014.] <http://www.ariadne-eu.org/>.

33. CAREO. [En línea] [Citado el: 05 de 06 de 2014.] <http://www.careo.org/about>.
34. Moodle Documentation. [En línea] [Citado el: 05 de 05 de 2014.] [http://docs.moodle.org/26/en/Main\\_page](http://docs.moodle.org/26/en/Main_page).
35. **Ivorra, R.** *Tutorial: Creación de un módulo de actividad. Moodle (1.9.3)*. Alicante, España : Reporte técnico, 2009.
36. **García, M. J.** *Análisis del desarrollo de extensiones para Moodle - Desarrollo de un módulo para la gestión de laboratorios docentes*. Alcalá, España : Tesis de grado de Universidad de Alcalá, 2010.
37. **Motz, R., y otros.** *La Extracción de Objetos de Aprendizaje con Metadatos de Diseño Pedagógico*. Montevideo, Uruguay : IEEE-RITA, 2010. ISSN 1932-8540.
38. **Pire, T., y otros.** *Extracción Automática de Metadatos de Objetos de Aprendizaje: un estudio comparativo*. Madrid, España : VI Congreso de Tecnología en Educación y Educación en Tecnología, 2011.
39. **Motz, R., y otros.** *LookIng4LO - Sistema informático para la extracción automática de Objetos de Aprendizaje*. Montevideo, Uruguay : IEEE-RITA, 2009. ISSN 1932-8540.
40. **Alfano, M., Lenzitti, B. y Visalli, N.** *System for Automatic eXtraction of E-learning object Features*. Coimbra, Portugal : 3rd E-Learning Conference, 2006.
41. **Wai, T. y Lam, K.** *Automatic Extraction of Learning Object Metadata (LOM) from HTML Web Pages*. Amsterdam, Holanda : Proceedings of the 2005 conference on Towards Sustainable and Scalable Educational Innovations Informed by the Learning Sciences: Sharing Good Practices of Research, Experimentation and Innovation, 2005. ISBN:1-58603-573-8.
42. **Burke, R.** *Hybrid Recommender Systems: Survey and Experiments*. California, Estados Unidos : Journal User Modeling and User-Adapted Interaction, 2002.
43. **Estellés, E. y González, F.** *Towards an integrated crowdsourcing definition*. Valencia, España : Journal of Information Science, 2012.
44. **Claypool, M., y otros.** *Combining content-based and collaborative filters in an online newspaper*. Massachusetts, Estados Unidos : ACM SIGIR Workshop on Recommender Systems, 1999.
45. **Boratto, L.** *Group Recommendation With Automatic Detection and Classification of Groups*. Cagliari, Italia : Tesi di Dottorato di Università degli Studi di Cagliari, 2011.

46. **Amer-Yahia, S., y otros.** *Group Recommendation: Semantics and Efficiency*. Texas, Estados Unidos : Journal Proceedings of the VLDB Endowment, 2009.
47. **Baron, R., Baldwin, T. y Martinez, D.** *Web Scraping Made Simple with SiteScraper*. Victoria, Australia : Reporte técnico, 2010.
48. **Chiu, D.** *A Script Lenguaje for Generating Internet-bots*. Munich, Alemania : Database and Expert Systems Applications. 12th International Workshop, 2001. ISBN: 0-7695-1230-5.
49. Notas de Moodle. *Moodle*. [En línea] [Citado el: 06 de 05 de 2014.] [http://docs.moodle.org/all/es/Notas\\_de\\_Moodle\\_2.0](http://docs.moodle.org/all/es/Notas_de_Moodle_2.0).
50. Web Harvest. [En línea] [Citado el: 06 de 05 de 2014.] <http://web-harvest.sourceforge.net/>.
51. Java. [En línea] [Citado el: 06 de 05 de 2014.] [http://www.java.com/es/download/whatis\\_java.jsp](http://www.java.com/es/download/whatis_java.jsp).
52. XSLT. W3C. [En línea] [Citado el: 06 de 05 de 2014.] <http://www.w3.org/TR/xslt>.
53. XQuery. W3C. [En línea] [Citado el: 06 de 05 de 2014.] <http://www.w3.org/standards/xml/query>.
54. XPath. W3C. [En línea] [Citado el: 06 de 05 de 2014.] <http://www.w3.org/TR/2011/REC-xpath-full-text-10-20110317/>.
55. Java HTML Parser. *JSoup*. [En línea] [Citado el: 06 de 05 de 2014.] <http://jsoup.org/>.
56. Document Object Model. W3C. [En línea] [Citado el: 06 de 05 de 2014.] <http://www.w3.org/DOM/>.
57. CSS. W3C. [En línea] [Citado el: 06 de 05 de 2014.] <http://www.w3.org/Style/CSS/>.
58. JQuery. [En línea] [Citado el: 06 de 05 de 2014.] <https://jquery.org/>.
59. HTTP State Management Mechanism. *IETF*. [En línea] <http://tools.ietf.org/html/rfc2109>.
60. **Sicilia, M., y otros.** *Exploring user-based recommender results in large learning object repositories: the case of MERLOT*. Barcelona, España : Procedia Computer Science - Elsevier B.V, 2010.
61. **Zapata, A., y otros.** *A framework for recommendation in learning object repositories: An example of application in civil engineering*. Oxford, Reino Unido : Journal Advances in Engineering Software, 2013. ISSN: 0965-9978.

62. **Alag, S.** *Collective Intelligence in Action*. 2009. ISBN: 1933988312.60.
63. **Papagelis, M. y Plexousakis, D.** *Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents*. Nueva York, Estados Unidos : Journal Engineering Applications of Artificial Intelligence, 2005.
64. The Apache Software Foundation. [En línea] [Citado el: 06 de 05 de 2014.] <http://www.apache.org/>.
65. Apache Hadoop. [En línea] [Citado el: 07 de 05 de 2014.] <http://hadoop.apache.org/>.
66. **Dean, J. y Ghemawat, S.** *MapReduce: a flexible data processing tool*. Nueva York, Estados Unidos : Magazine: Communications of the ACM - Amir Pnueli: Ahead of His Time, 2010.
67. **Sebastian, S. y Owen, S.** *Collaborative Filtering with Apache Mahout*. Dublin, Irlanda : RecSysChallenge, 2013.
68. iText, Programmable PDF Software. [En línea] [Citado el: 06 de 05 de 2014.] <http://itextpdf.com/>.
69. **Garlan, D. y Shaw, M.** *An Introduction to Software Architecture*. Nueva Jersey, Estados Unidos : Advances in Software Engineering and Knowledge Engineering, World Scientific, 1993.
70. **Gamma, E., y otros.** *Design Patterns - Elements of Reusable Object-Oriented Software*. 1994. ISBN-13: 978-0201633610.
71. Java Servlet Technology. ORACLE. [En línea] [Citado el: 06 de 05 de 2014.] <http://www.oracle.com/technetwork/java/index-jsp-135475.html>.
72. Java Server Pages Technology. ORACLE. [En línea] [Citado el: 06 de 05 de 2014.] <http://www.oracle.com/technetwork/java/javaee/jsp/index.html>.
73. **Sandhu, R., y otros.** *Role-Based Access Control Models*. Virginia, Estados Unidos : IEEE Computer, 1996.
74. Apache Shiro. [En línea] [Citado el: 06 de 05 de 2014.] <http://shiro.apache.org/>.
75. **Lemire, D. y Maclachlan, A.** *Slope One Predictors for Online Rating-Based Collaborative Filtering*. New Port Beach, Estados Unidos : SIAM International Conference on Data Mining, 2005. ISBN: 978-0-89871-593-4.
76. MySQL. [En línea] [Citado el: 06 de 05 de 2014.] <http://www.mysql.com/>.

77. Java Persistence API. *ORACLE*. [En línea] [Citado el: 06 de 05 de 2014.] <http://www.oracle.com/technetwork/java/javasee/tech/persistence-jsp-140049.html>.
78. Eclipse link. [En línea] [Citado el: 06 de 05 de 2014.] <https://www.eclipse.org/eclipselink/>.
79. **Woychowsky, E.** *Ajax: Creating Web Pages with Asynchronous JavaScript and XML*. Nueva Jersey, Estados Unidos : Prentice Hall, 2006. ISBN-10: 0-13-227267-9.
80. VirtualBox. [En línea] [Citado el: 06 de 05 de 2014.] <https://www.virtualbox.org/>.
81. Debian. [En línea] [Citado el: 06 de 05 de 2014.] <http://www.debian.org/>.
82. Apache Tomcat. [En línea] [Citado el: 06 de 05 de 2014.] <http://tomcat.apache.org/>.
83. Apache JMeter. [En línea] [Citado el: 11 de 05 de 2014.] <http://jmeter.apache.org/>.
84. **Herlocker, J., Terveen, L. y Riedl, J.** *Evaluating Collaborative Filtering Recommender Systems*. Nueva York, Estados Unidos : Journal ACM Transactions on Information Systems, 2004.
85. **Owen, S., y otros.** *Mahout in Action*. Nueva York, Estados Unidos : Manning Publications Co, 2012. ISBN 9781935182689.
86. Flickr. [En línea] [Citado el: 08 de 05 de 2014.] <https://www.flickr.com/>.
87. YouTube. [En línea] [Citado el: 08 de 05 de 2014.] <http://www.youtube.com/>.
88. Wikimedia. [En línea] [Citado el: 08 de 05 de 2014.] <http://www.wikimedia.org/>.
89. **Wang, J., de Vries, A. y Reinders, M. T. J.** *Unifying User-based and Item-based Collaborative Filtering Approaches by Similarity Fusion*. Nueva York, Estados Unidos : Proceeding SIGIR '06 Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, 2006. ISBN: 1-59593-369-7.

# Anexo



## 1. Descripción de Actividades y Recursos que se pueden incluir en Moodle.

En Moodle, las distintas herramientas o módulos que se pueden incluir en un curso se dividen en dos grandes categorías: Actividades y Recursos. A continuación se detallan los distintos tipos de módulos disponibles.

*Base de datos.* El módulo de actividad de base de datos permite a los participantes crear, mantener y buscar información en un repositorio de registros. La estructura de las entradas la define el docente según una lista de campos. Los tipos de campo incluyen casilla de verificación, botones de radio, menú desplegable, área de texto, URL, imagen y archivo cargado. La presentación visual de la información al listar, ver o editar las entradas de la base de datos se controla mediante plantillas de base de datos. Las actividades de base de datos pueden ser compartidas entre los cursos como opción preestablecida de manera que un profesor también puede importar y exportar las entradas de base de datos. Si el filtro de base de datos "auto enlace" está activo, cualquier entrada de la base de datos podrá ser enlazada automáticamente desde donde las palabras o frases aparecen dentro en el curso. Un docente puede permitir comentarios en las entradas. Las entradas también pueden ser calificadas por profesores u otros estudiantes (evaluación por pares). Las calificaciones se agregarán para formar una calificación final que se registrará en el libro de calificaciones.

Las actividades de base de datos tienen muchos usos, como por ejemplo:

- Una colección de enlaces de colaboración web, libros, reseñas de libros, referencias de revistas, etc.
- Para la visualización de fotos, carteles, sitios web o poemas de los estudiantes, ya así poder ser comentados por otros estudiantes.

*Chat.* La actividad chat permite a los participantes tener una discusión en formato texto de manera sincrónica en tiempo real. El chat puede ser una actividad puntual o puede repetirse a la misma hora cada día o cada semana. Las sesiones de chat se guardan y pueden hacerse públicas para que todos las vean o limitadas a los usuarios con permiso para ver los registros de sesiones del chat. Los chats son especialmente útiles cuando un grupo no tiene posibilidad de reunirse físicamente para poder conversar cara-a-cara, como por ejemplo:

- Reuniones programadas de estudiantes inscritos a cursos en línea, para permitirles compartir experiencias con otros compañeros del mismo curso pero de diferentes ciudades o países.
- Un estudiante que temporalmente no puede asistir en persona, podría chatear con su profesor para ponerse al día del trabajo escolar.
- Estudiantes que empiezan a trabajar se juntan para discutir sus experiencias entre ellos y con el docente.
- Una sesión de preguntas y respuestas con un orador invitado de una localidad diferente (a distancia).
- Sesiones para ayudar a los estudiantes a prepararse para exámenes, donde el docente,

o los estudiantes, hagan preguntas de ejemplo.

*Consulta.* El módulo Consulta permite al profesor hacer una pregunta especificando las posibles respuestas posibles. Los resultados de la elección pueden ser publicados después que los estudiantes hayan respondido, después de cierta fecha, o no publicarse. Los resultados pueden ser publicados, con los nombres de los estudiantes o de forma anónima.

Una Consulta puede utilizarse:

- Para realizar una encuesta rápida que estimule a los alumnos a reflexionar sobre un tema.
- Para comprobar rápidamente que los estudiantes han entendido algo concreto.
- Para facilitar la toma de decisiones, por ejemplo permitiendo a los estudiantes votar algún aspecto relacionado con el curso.

*Cuestionario.* La actividad Cuestionario permite al profesor diseñar y plantear cuestionarios con preguntas de tipo múltiple opción, verdadero/falso, coincidencia, respuesta corta y respuesta numérica. El profesor puede permitir que el cuestionario se intente resolver varias veces, con las preguntas ordenadas o seleccionadas aleatoriamente del banco de preguntas. Se puede establecer un tiempo límite. Cada intento se califica automáticamente, con la excepción de las preguntas de tipo "ensayo", y el resultado se guarda en el libro de calificaciones. El profesor puede determinar si se muestran y cuándo se muestran al usuario los resultados, los comentarios de retroalimentación y las respuestas correctas.

Los cuestionarios pueden usarse para hacer:

- Exámenes del curso
- Mini Test para tareas de lectura o al final de un tema
- Exámenes de práctica con preguntas de exámenes anteriores
- Para ofrecer información inmediata sobre el rendimiento
- Para auto-evaluación

*Encuestas predefinidas.* El módulo de actividad Encuestas predefinidas proporciona una serie de instrumentos que se han mostrado útiles para evaluar y estimular el aprendizaje en entornos en línea. Un profesor puede usarlos para recopilar información entre sus alumnos que le ayude a conocer mejor su clase así como su propia forma de enseñar.

*Foro.* El módulo de actividad Foro permite a los participantes establecer debates asíncronos que pueden tener lugar durante un período prolongado de tiempo. Hay varios tipos de foro para elegir, como un foro estándar donde cualquier persona puede iniciar una nueva discusión en cualquier momento, un foro en el que cada alumno puede publicar una sola discusión, o un foro de preguntas y respuestas donde los estudiantes primero deben hacer una aportación antes de poder ver los mensajes de otros estudiantes. El profesor puede permitir que se adjunten archivos a los mensajes del foro. Las imágenes adjuntas se muestran en el mensaje del foro. Los participantes pueden suscribirse a un foro para recibir notificaciones de nuevos mensajes en el foro. Un profesor puede establecer el modo de suscripción opcional, forzado o automático, o prohibir totalmente la suscripción. Si es necesario, los estudiantes pueden ser

bloqueados a partir de un determinado número de aportaciones en un período de tiempo determinado, lo que puede impedir que determinadas personas dominen las discusiones.

Los aportes al foro pueden ser evaluados por los profesores o por los estudiantes (evaluación por pares). Las calificaciones pueden consolidarse para formar una calificación final que se registra en el libro de calificaciones.

Los Foros tienen muchos usos, como por ejemplo:

- Un espacio social para que los estudiantes se conozcan
- Publicar avisos en los cursos (usando un foro de noticias con suscripción forzada)
- Debatir el contenido del curso o los materiales de lectura
- Continuar en línea un debate planteada previamente en una sesión presencial (cara a cara)
- Debates solo para profesores (mediante un foro oculto)
- Centro de ayuda donde los tutores y los estudiantes pueden dar consejos
- Área privada uno-a-uno para comunicación alumno-profes (con un foro con distintos grupos y con un estudiante por grupo)
- Ampliación de actividades, como una puesta en común donde los estudiantes reflexionen y propongan soluciones

*Glosario.* El módulo de actividad glosario permite a los participantes crear y mantener una lista de definiciones, de forma similar a un diccionario, o para recoger y organizar recursos o información. El profesor puede permitir que se adjunten archivos a las entradas del glosario. Las imágenes adjuntas se mostrarán en la entrada. Las entradas se pueden buscar y se puede navegar por ellas en orden alfabético o por categoría, fecha o autor. Las entradas pueden aprobarse por defecto o requerir la aprobación de un profesor antes de que sean visibles para los demás alumnos. Si se ha habilitado el filtro de vinculación automática del glosario, las entradas se enlazan automáticamente cuando las palabras o frases aparecen en el curso. El docente puede permitir comentarios en las entradas. Las entradas también se pueden calificar por profesores o por los demás estudiantes (evaluación por pares). Las calificaciones pueden agregarse para formar una calificación final que se registra en el libro de calificaciones.

Los glosarios tienen muchos usos, como:

- Un registro cooperativo de términos clave.
- Un espacio para darse a conocer, donde los estudiantes nuevos añadan su nombre y sus datos personales.
- Un recurso con "consejos prácticos" con las mejores prácticas en un tema concreto.
- Un área para compartir vídeos, imágenes o archivos de sonido.
- Un recurso con "asuntos que recordar".

*Herramienta externa.* El módulo de actividad de herramienta externa les permite a los estudiantes interactuar con recursos educativos y actividades alojadas en otros sitios de internet. Por ejemplo, una herramienta externa podría proporcionar acceso a un nuevo tipo de actividad o de materiales educativos de una editorial. Para crear una actividad de herramienta externa se requiere un proveedor de herramienta que soporte LTI (Learning Tools Interoperability, en español, Interoperabilidad de Herramientas de Aprendizaje). Un maestro

puede crear una actividad de herramienta externa o hacer uso de una herramienta configurada por el administrador del sitio.

Las herramientas externas difieren de los recursos URL en varias formas:

- Las herramientas externas están conscientes del contexto, por ejemplo: tienen acceso a información acerca del usuario que invocó la herramienta, como por ejemplo la institución, curso y nombre.
- Las herramientas externas soportan leer, actualizar y borrar calificaciones asociadas con la instancia de la actividad.
- Las configuraciones de la herramienta externa crean una relación de confianza entre su sitio Moodle y el proveedor de la herramienta, permitiendo la comunicación segura entre ambos.

*Lección.* La actividad lección permite a un profesor presentar contenidos y/ o actividades prácticas de forma interesante y flexible. Un profesor puede utilizar la lección para crear un conjunto lineal de páginas de contenido o actividades educativas que ofrezcan al alumno varios itinerarios u opciones. En cualquier caso, los profesores pueden optar por incrementar la participación del alumno y asegurar la comprensión mediante la inclusión de diferentes tipos de pregunta, tales como la elección múltiple, respuesta corta y correspondencia. Dependiendo de la respuesta elegida por el estudiante y de cómo el profesor desarrolla la lección, los estudiantes pueden pasar a la página siguiente, volver a una página anterior o dirigirse a un itinerario totalmente diferente. Una lección puede ser calificada y la calificación registrada en el libro de calificaciones.

Las lecciones pueden ser utilizadas:

- Para el aprendizaje auto dirigido de un nuevo tema.
- Para ejercicios basados en escenarios o simulaciones y de toma de decisiones.
- Para realizar ejercicios de repaso diferenciadas, con distintos conjuntos de preguntas de repaso, dependiendo de las respuestas dadas a las preguntas anteriores.

*Paquete SCORM.* Un paquete SCORM es un conjunto de archivos que se empaquetan conforme a una norma estándar para los objetos de aprendizaje. El módulo de actividad SCORM permite cargar y añadir a los cursos paquetes SCORM o AICC como archivos Zip. El contenido se muestra normalmente en varias páginas, con navegación entre las páginas. Hay varias opciones para la visualización de los contenidos, con ventanas pop-up, en tablas de contenidos, con botones de navegación, etc. Las actividades SCORM generalmente incluyen preguntas calificables, que se registra en el libro de calificaciones.

Las actividades SCORM se pueden usar:

- Para la presentación de contenidos multimedia y animaciones.
- Como herramienta de evaluación.

*Taller.* El módulo de actividad taller permite la recopilación, revisión y evaluación por pares del trabajo de los estudiantes. Los estudiantes pueden enviar cualquier contenido digital (archivos), tales como documentos de procesador de texto o de hojas de cálculo y también pueden escribir el texto directamente en un campo empleando un editor de texto (dentro de Moodle). Los envíos son evaluados empleando un formato de evaluación de criterios múltiples definido por el

profesor. El proceso de revisión por pares y el formato para comprender cómo funciona la evaluación se pueden practicar por anticipado con envíos de ejemplo proporcionados por el maestro, junto con una evaluación de referencia. A los estudiantes se les dará la oportunidad de evaluar uno o más de los envíos de sus pares estudiantes. Los que envían y los que evalúan pueden permanecer anónimos si se requiere así. Los estudiantes tendrán dos calificaciones para la actividad de taller: una calificación por enviarlo y otra por la evaluación de sus pares. Ambas calificaciones se guardan en el libro de calificaciones.

*Tarea.* El módulo de Tareas permite a un profesor evaluar el aprendizaje de los alumnos mediante la creación de una tarea a realizar que luego revisará, valorará y calificará. Los alumnos pueden presentar cualquier contenido digital (archivos), como documentos de texto, hojas de cálculo, imágenes, audio y vídeos entre otros. Alternativamente, o como complemento, la tarea puede requerir que los estudiantes escriban texto directamente en un campo utilizando el editor de texto. Una tarea también puede ser utilizada para recordar a los estudiantes tareas del "mundo real" que necesitan realizar y que no requieren la entrega de ningún tipo de contenido digital. Al revisar las tareas, los profesores pueden dejar comentarios de retroalimentación y subir archivos, tales como anotaciones a los envíos de los estudiantes, documentos con observaciones o comentarios en audio. Las tareas pueden ser clasificadas según una escala numérica o según una escala personalizada, o bien, mediante un método de calificación avanzada, como una rúbrica. Las calificaciones finales se registran en el libro de calificaciones.

*Wiki.* El módulo de actividad wiki les permite a los participantes añadir y editar una colección de páginas web. Un wiki puede ser colaborativo, donde todos pueden editarlo, o puede ser individual, donde cada persona tiene su propio wiki que solamente ella podrá editar.

Se conserva un histórico de las versiones previas de cada página del wiki, permitiendo consultar los cambios hechos por cada participante.

Los wikis tienen muchos usos, como por ejemplo:

- Para generar unos apuntes de clase colaborativamente entre todos.
- Para los profesores de una escuela que planean una estrategia o reunión de trabajo en equipo.
- Para estudiantes que trabajarán en equipo en un libro en línea, creando contenidos de un tema elegido por sus tutores.
- Para la narración colaborativa o creación de poesía grupal, donde cada participante escribe una línea o un verso.
- Como un diario personal para apuntes para examen o resúmenes (wiki personal).

*Carpeta.* El recurso Carpeta permite al profesor mostrar un grupo de archivos relacionados dentro de una única carpeta. Se puede subir un archivo comprimido (Zip) que se descomprimirá (unzip) posteriormente para mostrar su contenido, o bien, se puede crear una carpeta vacía y subir los archivos dentro de ella. Una carpeta se puede usar para:

- Agrupar una serie de documentos sobre un tema, por ejemplo, un conjunto de exámenes de otros años en formato PDF, o una colección de archivos para crear un proyecto concreto por parte de los estudiantes.

- Crear un espacio de subida de archivos compartido entre los profesores del curso (se debería ocultar la carpeta a los alumnos para que lo vean solo los docentes).

*Etiqueta.* El módulo etiqueta permite insertar texto y elementos multimedia en las páginas del curso entre los enlaces a otros recursos y actividades. Las etiquetas son muy versátiles y pueden ayudar a mejorar la apariencia de un curso si se usan cuidadosamente. Las etiquetas pueden ser utilizadas:

- Para dividir una larga lista de actividades con un subtítulo o una imagen.
- Para visualizar un archivo de sonido o vídeo incrustado directamente en la página del curso.
- Para añadir una breve descripción de una sección del curso.

*Libro.* El módulo libro permite al profesor crear un recursos multi-página en un formato similar al de un libro, con capítulos y subcapítulos. Los libros pueden contener archivos de medios además del texto, y son útiles para mostrar textos largos divididos en secciones.

Se puede utilizar un libro para:

- Mostrar material de lectura para un módulo concreto de estudio.
- Como un manual departamental.
- Como un compendio de información para el estudiante.

*Página.* El recurso Página permite a los profesores crear una página web mediante el editor de textos. Una Página puede mostrar texto, imágenes, sonido, vídeo, enlaces web y código incrustado (como por ejemplo los mapas de Google) entre otros. Entre las ventajas de utilizar el recurso Página en lugar del recurso de Archivo está que el recurso es más accesible (por ejemplo, para usuarios de dispositivos móviles) y de más fácil actualización.

Una página puede ser utilizada para:

- Dar a conocer los términos y condiciones de un curso o el programa de la asignatura.
- Para incrustar varios vídeos o archivos de sonido, junto con un texto explicativo.

*Paquete de contenido IMS.* Un paquete de contenidos IMS permite mostrar dentro del curso paquetes de contenidos creados conforme a la especificación IMS Content Packaging.

*URL.* El recurso URL permite que el profesor pueda proporcionar un enlace de Internet como un recurso del curso. Todo aquello que esté disponible en línea, como documentos o imágenes, puede ser vinculado, la URL no tiene por qué ser la página principal de un sitio web. La dirección URL de una página web en particular puede ser copiada y pegada por el docente, o bien, éste puede utilizar el selector de archivo y seleccionar una URL desde un repositorio, como Flickr [86], YouTube [87] o Wikimedia [88] (dependiendo de qué repositorios están habilitados para el sitio). Hay una serie de opciones de visualización de la URL, como incrustada o abierta en una nueva ventana, y opciones avanzadas, como agregar información a la URL, como el nombre de un estudiante.

## 2. Proceso de Filtrado Colaborativo

El objetivo de un algoritmo de filtrado colaborativo es sugerir nuevos objetos o predecir la utilidad de cierto objeto para un usuario en particular basado en sus intereses y opiniones de otros usuarios con intereses similares. En un caso típico se tiene una lista de usuarios y una lista de ítems [7]. A su vez, cada usuario posee una lista de elementos sobre los cuales ha expresado su opinión. Estas opiniones (o puntajes) pueden ser dadas *explícitamente* por el usuario (p.e. el usuario puntúa un artículo que compró en una tienda online) o *implícitamente* mediante un análisis de los registros de compras del usuario. Es importante notar que es posible que la lista de ítems de un usuario pueda ser vacía. Al usuario que se le quiere realizar la recomendación se le llama *usuario activo* o *usuario objetivo*. Para hablar de filtrado colaborativo es necesario definir dos conceptos fundamentales; *predicción* y *recomendación*. La **predicción** es un valor numérico que expresa la puntuación otorgada por el *usuario objetivo* al ítem, predicha por el sistema de recomendación. Se asume que el ítem a analizar no ha sido puntuado. El valor de la predicción está en la misma escala que el valor de los puntajes. La **recomendación** es la lista de elementos que le gustarán más al usuario objetivo.

Los algoritmos de FC representan una matriz de Usuarios  $\times$  Ítems, donde en cada celda se tienen los puntajes correspondientes. Por ejemplo, en la celda  $(i, j)$  se tiene el valor numérico del puntaje otorgado por el usuario  $i$  sobre el ítem  $j$ , o el valor 0 en el caso de que el usuario aún no haya puntuado el elemento. Como se mencionó anteriormente, los algoritmos de FC se pueden clasificar en dos categorías; las basadas en el usuario (*basadas en memoria*) y las basadas en los ítems (*basadas en el modelo*).

### Algoritmos basados en los ítems

En los algoritmos basados en los ítems, se examinan los elementos puntuados por el usuario y se determinan qué tan similares son al ítem objetivo. Luego se recuperan los  $n$  ítems más similares, mientras que al mismo tiempo se calcula el grado de similitud de cada uno. Finalmente se realiza la predicción mediante sumas ponderadas de los puntajes de los elementos similares. A continuación se profundizará en el cálculo de *similitud* y *predicción*.

#### Similitud

Existen varias técnicas de computar la similitud entre elementos, entre ellas; *Similitud basada en el coseno*, *Correlación de Pearson*, y *Similitud basada en el coseno ajustado*.

#### Similitud basada en el coseno

En este caso, los dos ítems son tratados como dos vectores de cardinalidad  $m$ , siendo  $m$  la cantidad de Usuarios en el sistema. La similitud entre ellos es expresada mediante el cálculo del coseno del ángulo entre los vectores.

$$sim(i, j) = \text{coseno}(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\|\vec{i}\|_2 \times \|\vec{j}\|_2}$$

Donde  $\cdot$  es el producto vectorial, y  $\|\cdot\|$  es la norma de un vector.

#### Correlación de Pearson

En este caso, la similaridad entre dos ítems  $i$  y  $j$  es expresada mediante el cálculo de la correlación de Pearson. Para definir dicha correlación, primero debemos acotar el conjunto de usuarios del sistema, analizando únicamente al subconjunto  $U$  de usuarios que han puntuado a los ítems  $i$  y  $j$ .

$$sim(i, j) = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_i) (R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_i)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_j)^2}}$$

Donde  $R_{u,i}$  es el puntaje del usuario  $u$  sobre el ítem  $i$ , y  $\bar{R}_i$  es el puntaje promedio de ítem  $i$ .

#### Similaridad basada en el coseno ajustado

Una diferencia fundamental entre el cálculo de similaridades en el Filtrado Colaborativo basado en el usuario y el basado en los ítems es que en el primero la similaridad se calcula en función de las filas de la matriz de Usuarios x Ítems, mientras que en el otro se calcula en función de las columnas. Calcular la similaridad utilizando el coseno (explicado en el punto anterior) tiene una gran desventaja; la diferencias de puntajes entre distintos usuarios no es tenida en cuenta. La similaridad basada en el coseno ajustado mejora este problema eliminando el promedio de puntajes del usuario correspondiente de cada par de ítems puntuados.

$$sim(i, j) = \frac{\sum_{u \in U} (R_{u,i} - \bar{R}_u) (R_{u,j} - \bar{R}_u)}{\sqrt{\sum_{u \in U} (R_{u,i} - \bar{R}_u)^2} \sqrt{\sum_{u \in U} (R_{u,j} - \bar{R}_u)^2}}$$

Donde  $\bar{R}_u$  es el promedio de los puntajes otorgados por el usuario  $u$ .

#### Predicción

Luego de determinado el conjunto de ítems más similares, el siguiente paso (y más importante) es analizar los puntajes otorgados por el usuario objetivo y aplicar alguna técnica para obtener la *predicción*. A continuación se detallan dos técnicas: Suma ponderada y Regresión.

### Suma Ponderada

Para determinar la predicción de un ítem  $i$  para un usuario  $u$  se calcula la suma de puntajes dados por el usuario a los ítems similares a  $i$ . Luego, cada puntaje es ponderado por el grado de similitud entre el par de ítems en análisis.

$$P_{u,i} = \frac{\sum_{j \in S_i} (s_{i,j} \times R_{u,j})}{\sum_{j \in S_i} (|s_{i,j}|)}$$

Donde  $S_i$  es el conjunto de ítems similares a  $i$ ,  $R_{u,j}$  es el puntaje del usuario  $u$  sobre el ítem  $j$ , y  $s_{i,j}$  es el grado de similitud entre los ítems  $i$  y  $j$ .

### Regresión

A diferencia de las sumas ponderadas, en el modelo de regresión no se utilizan los puntajes de los objetos similares directamente, sino que utiliza una aproximación basada en el modelo de regresión. Es decir, se aplica la misma fórmula, pero en vez de utilizar los puntajes de los ítems similares, se utiliza un valor aproximado que se calcula de la siguiente forma:

$$R'_{u,j} = \alpha R'_{u,i} + \beta + \epsilon$$

Donde  $\alpha$  y  $\beta$  son parámetros, y  $\epsilon$  es el error del modelo.

## Algoritmos basados en los usuarios

Estos algoritmos predicen al usuario objetivo el interés sobre un ítem en base a la información de puntuaciones de usuarios con perfiles similares al usuario al objetivo [89]. Cuanto más sean las puntuaciones de usuarios similares, más contribuyen a la predicción del ítem objetivo. El conjunto de los usuarios similares puede ser seleccionado estableciendo un umbral, o tomando los  $N$  usuarios cuyos perfiles son los más parecidos (*top-N*). En el caso de *top-N*, el conjunto de los  $N$  usuarios similares  $S_k(u_k)$  al usuario  $k$ , puede verse como:

$$S_k(u_k) = \{u_a | \text{rank}_{s_a}(u_k, u_a) \leq N, x_{a,m} \neq \emptyset\}$$

Donde  $|S_u(u_k)| = N$ .

$S_u(u_k, u_a)$  es la similitud entre los usuarios  $k$  y  $a$ . La similitud del coseno y la correlación de Pearson son las medidas de similitud más populares en filtrado colaborativo. La similaridad puede ser también adquirida a través de datos de entrenamiento. Utilizando la similitud basada en el coseno del ángulo para la comparación de dos perfiles de usuarios, la puntuación predicha para el ítem objetivo  $m$  por el usuario objetivo  $k$  se puede computar como:

$$\hat{x}_{k,m} = \bar{u}_k + \sum_{u_a \in S_u(u_k)} S_u(u_k, u_a) (x_{a,m} - \bar{u}_a) / \sum_{u_a \in S_u(u_k)} S_u(u_k, u_a)$$

Donde  $u_k$  y  $u_a$  denotan la puntuación promedio realizada por los usuarios  $k$  y  $a$  respectivamente.

Los métodos varían según como sean tratadas las puntuaciones desconocidas, es decir,  $x_{a,m} = \emptyset$ . Las puntuaciones faltantes pueden ser reemplazadas por 0, lo cual disminuiría la predicción, o pueden ser reemplazadas por una interpolación del promedio de las puntuaciones del usuario.

### 3. Recomendaciones grupales

#### Relevancia grupal

Son dos las estrategias más utilizadas para el cálculo de la relevancia grupal. Una estrategia es calculando la relevancia promedio, y la otra es tomando la relevancia grupal igual a la menor de las relevancias individuales de los miembros del grupo. La relevancia de un ítem  $i$  de un grupo  $G$ , denotado como  $rel(G, i)$  puede verse entonces como:

$$rel(G, i) = \frac{1}{|G|} \sum_{u \in G} relevancia(u, i)$$

Que consiste en tomar el promedio de las relevancias individuales. O se puede tomar como:

$$rel(G, i) = \text{Min}_{u \in G} (relevancia(u, i))$$

Que toma la relevancia grupal como la relevancia mínima.

La segunda estrategia considera los casos extremos donde por ejemplo en un contexto donde se recomiendan películas, se tiene un grupo en el cual hay un usuario que no le agradan las películas de terror, en este caso el grupo calificará este tipo de películas con un puntaje bajo, a pesar de que existan usuarios que tengan preferencia por este tipo de películas.

#### Discrepancia grupal

La discrepancia grupal refleja el grado de discrepancia que hay entre los miembros de un grupo para cierto ítem  $i$ . Se consideran dos métodos principales para determinar esta discrepancia. Un método se conoce como discrepancia par a par promedio y el otro como varianza de discrepancia. La discrepancia para de un grupo  $G$  para un ítem  $i$  se denota como  $dis(G, i)$  y puede calcularse como:

$$dis(G, i) = \frac{2}{|G|(|G| - 1)} \sum (|relevancia(u, i) - relevancia(v, i)|)$$

Donde  $u \neq v$  y  $u, v \in G$ . O también puede calcularse como:

$$dis(G, i) = \frac{1}{|G|} \sum_{u \in G} (relevancia(u, i) - \lambda)^2$$

Donde  $\lambda$  es la media de las relevancias individuales para el ítem  $i$ .

Intuitivamente, cuanto más cercanos sean los puntajes del usuario  $u$  y  $v$  para el ítem  $i$ , menos será la discrepancia para el ítem  $i$ .

## **Función consenso**

Finalmente, combinando la relevancia grupal con la discrepancia grupal, se obtiene la función consenso o *consensus function*:

$$F(G, i) = w_1 \times rel(G, i) + w_2 \times (1 - dis(G, i)), w_1 + w_2 = 1$$

Las variables  $w_1$  y  $w_2$  se utilizan para ponderar la relevancia y discrepancia.

Luego, para obtener el listado de recomendaciones para el grupo  $G$ , se debe identificar una lista de ítems que cumplan las siguientes condiciones:

1.  $|I_G| = k$
2.  $\forall i \in I_G, u \in G, u$  no ha puntuado  $i$  antes.
3.  $I_G$  está ordenada de forma decreciente según los puntajes calculados por la función consenso, y se cumple que  $\nexists j \in I / F(G, j) > F(G, i), j \notin I_G, i \in I_G$

## 4. Herramienta de extracción de datos: Web Harvest

Web Harvest es una herramienta libre de extracción de datos desde la web, desarrollada en Java. Permite obtener las páginas web deseadas y extraer la información requerida de ellas. La herramienta ejecuta scripts escritos en un lenguaje de programación basado en la utilización de etiquetas o *tags*.

Web Harvest permite definir procedimientos de extracción a través de archivos de configuración basados en XML. Cada archivo de configuración describe una secuencia de procesos que ejecutan una tarea común con el fin de lograr la meta final. Estos procesos se ejecutan en cadena, por lo tanto la salida de la ejecución de un proceso es la entrada del siguiente. Esto puede explicarse mejor con el siguiente fragmento de configuración:

```
<xpath expression="//a[@shape='rect']/@href">
  <html-to-xml>
    <http url="http://www.misitio.com/">
  </html-to-xml>
</xpath>
```

Cuando Web Harvest ejecuta esta parte del archivo de configuración, ocurre lo siguiente:

1. El proceso *http* descarga el contenido de la URL especificada.
2. El proceso *html-to-xml* limpia ese HTML produciendo contenido XHTML.
3. El proceso *xpath* busca links (tag HTML `<a>` con atributo *shape* con valor igual a "rect") específicos en el XHTML obtenido en el paso previo dando como resultado una secuencia de URLs (el atributo *href* de los links obtenidos).

Además Web Harvest soporta un conjunto de procesos que sirven para la manipulación de variables, saltos condicionales, bucles, funciones, operaciones con archivos, procesamiento de HTML y XML y manejo de excepciones.

## 5. Herramienta de recomendación: Apache Mahout

Apache Mahout es una herramienta utilizada en el área de la recomendación, desarrollada en un proyecto perteneciente a la Fundación Apache. Su objetivo es la construcción de una librería de aprendizaje automático escalable.

Los conjuntos de datos o *dataset* que contienen todas las interacciones por parte de los usuarios (accesos o puntajes sobre los ítems), se representan mediante una clase denominada *DataModel*. Ésta provee diversos métodos de acceso, por ejemplo, permite obtener la cantidad de usuarios (*getNumUsers*) y las interacciones de un ítem en particular (*getPreferencesForItem*). Mahout permite almacenar éstas interacciones tanto en memoria, como en disco o bases de datos relacionales. En particular, utilizar bases de datos es muy útil ya que es posible configurar la herramienta de forma que tome los datos de preferencias almacenados directamente desde una tabla y compute las recomendaciones fácilmente.

A continuación se detalla a modo de ejemplo como configurar la conexión a la base de datos utilizando clases de Apache Mahout específicas para MySQL.

```
MysqlDataSource dataSource = new MysqlDataSource();
dataSource.setURL("jdbc:mysql://localhost:3306/repositorio");
dataSource.setUser("db_user");
dataSource.setPassword("db_pass");
String preferenceTable = "usuario_loan";
String userIDColumn = "usuarioId";
String itemIDColumn = "loanId";
String preferenceColumn = "puntaje";
String timestampColumn = "fechaPuntaje";
MySQLJDBCDataModel mysqlModel = new MySQLJDBCDataModel(dataSource, preferenceTable,
userIDColumn, itemIDColumn, preferenceColumn, timestampColumn);
```

Primero, se ingresan los datos específicos a la conexión, como URL, usuario y contraseña. Luego, se especifican los detalles de la estructura de la tabla donde se almacenan las interacciones entre usuarios e ítems. Finalmente, se crea una instancia del elemento *mysqlModel*, el cual es utilizado como parámetro de las operaciones de recomendación provistas por Mahout.

En escenarios reales con grandes volúmenes de datos, es posible querer excluir elementos y no considerar toda la información existente para evitar problemas de performance. Para estos casos, Mahout provee la clase configurable denominada *CandidateItemsStrategy* la cual se encarga de recuperar los ítems que pueden ser recomendados para un usuario particular.

En un sistema de recomendaciones basado en usuarios se puede utilizar la clase *UserNeighborhood* para obtener el conjunto de usuarios que actúan como jurado para encontrar los ítems a recomendar. Mientras que para un sistema de recomendaciones basado en ítems, Mahout provee las clases *ItemBasedRecommender* e *ItemSimilarity*, que permiten

computar la recomendación de acuerdo la similaridad entre elementos. El framework también incluye diversas medidas como Correlación de Pearson, Similaridad basada en el coseno, etc.

A continuación se presenta un ejemplo del uso de estas clases con Apache Mahout, donde se obtienen los tres elementos “más similares”.

```
ItemBasedRecommender recommender = new GenericItemBasedRecommender(mysqlModel,  
    new PearsonCorrelationSimilarity(mysqlModel));  
List<RecommendedItem> similarItems = recommender.mostSimilarItems(itemID,3);
```

Mahout también ofrece herramientas para la evaluación de la calidad de las predicciones en una recomendación utilizando un subconjunto de datos aleatorio. Mediante la clase *RecommenderEvaluator* provista por Mahout, es posible calcular diferentes métricas tales como Mean Average Error y Root Mean Squared Error. Y utilizando la clase *RecommenderIRStatsEvaluator*, es posible obtener estadísticas tales como Precision, Recall y otras medidas. Por más detalles, ver sección 6.1.

## 6. Seguridad con Apache Shiro

A continuación se presentan ejemplos de las configuraciones básicas de Apache Shiro.

1. `jdbcRealm.authenticationQuery = select u.password from usuario u where u.mail = ?`
2. `jdbcRealm.userRolesQuery = select r.nombre from rol r, usuario_rol ru, usuario u where u.mail = ? and u.id = ru.usuarios_id and ru.roles_id = r.id`
3. `jdbcRealm.permissionsQuery = select p.codigo from rol r, rol_permiso pr, permiso p where r.nombre = ? and r.id = pr.roles_id and pr.permisos_id = p.id`

En la primera línea se indica la estructura de la tabla donde se almacena la contraseña de acceso al sistema (autenticación). En la línea 2 se especifica la forma de obtener los roles de un determinado usuario. Y en la línea 3, se indica la forma de obtener todos los permisos asociados a un rol. A partir de estas tres sentencias Apache Shiro es capaz de determinar el conjunto de permisos que posee un determinado usuario (autorización).

Es fundamental especificar el motor de base de datos (MySQL en este caso) y la información para establecer la conexión:

```
dataSource = com.mysql.jdbc.jdbc2.optional.MysqlDataSource
dataSource.url = jdbc:mysql://localhost:3306/repositorio
dataSource.serverName = localhost
dataSource.user = BD_USER
dataSource.password = BD_PASS
dataSource.databaseName = repositorio
dataSource.portNumber = 3306
```

Shiro también resuelve la autenticación mediante un formulario web. Para ello, es necesario indicar el nombre de los campos dentro del formulario y la URL del mismo.

1. `authc.loginUrl = /jsp/auth/login.jsp`
2. `authc.usernameParam = userName`
3. `authc.passwordParam = password`
4. `authc.successUrl = /jsp/auth/login.jsp`

En la primera línea se indica la ruta a la página de autenticación. Todo intento de acceso a un sitio que requiera de un usuario autorizado, será redirigido a esta página. Las líneas 2 y 3 especifican el nombre de los parámetros donde se debe ingresar el nombre de usuario y contraseña. La última línea, indica la ruta de la página a que debe redirigir el sistema una vez que el usuario se ha autenticado correctamente.

Por último, el framework permite seleccionar qué directorios dentro de la aplicación web son públicos (accesibles sin autenticación) y cuales no (se debe estar autenticado en el sistema para poder acceder). Esto se especifica mediante las siguientes sentencias:

```
/css/**=anon  
/js/**=anon  
/jsp/auth/**=authc  
/jsp/*s*=anon
```

Donde se indica que los archivos bajo el directorio *css* y *js* son públicos (usuario anónimo), mientras que los que se encuentran bajo el directorio */jsp/auth/* requieren autenticación. Además, Apache Shiro dispone de una librería específica para JSP, la cual mediante el uso de etiquetas (*tags*) se puede discriminar qué componentes de la vista serán desplegados al usuario de acuerdo a los permisos que éste tenga asignados. Estas etiquetas se identifican con el prefijo *shiro*. A continuación se detallan algunos ejemplos.

`<shiro:guest/>` - Despliega el contenido sólo si el usuario no es conocido por el sistema.

`<shiro:user/>` - Despliega el contenido sólo si el usuario se encuentra autenticado en el sistema.

`<shiro:principal/>` - Despliega el nombre del usuario con el que éste se autentica en el sistema.

`<shiro:hasPermission name="doSomething"/>` - Despliega el contenido sólo si el usuario posee el permiso identificado por la clave *doSomething*. Para chequear que el usuario efectivamente posee dicho permiso es que el framework utiliza las sentencias *userRolesQuery* y *permissionsQuery* explicadas anteriormente.

`<shiro:lacksPermission name="doSomething"/>` - Despliega el contenido sólo si el usuario NO posee el permiso identificado por la clave *doSomething*.

`<shiro:hasRole name="admin"/>` - Despliega el contenido sólo si el usuario posee el rol identificado por la clave *admin*.

`<shiro:lacksRole name="admin"/>` - Despliega el contenido sólo si el usuario NO posee el rol identificado por la clave *admin*.

`<shiro:hasAnyRoles name="rol1, rol2, rol3"/>` - Despliega el contenido sólo si el usuario posee alguno de los roles identificados por la clave *rol1, rol2, rol3*.

## 7. Interfaz de usuario

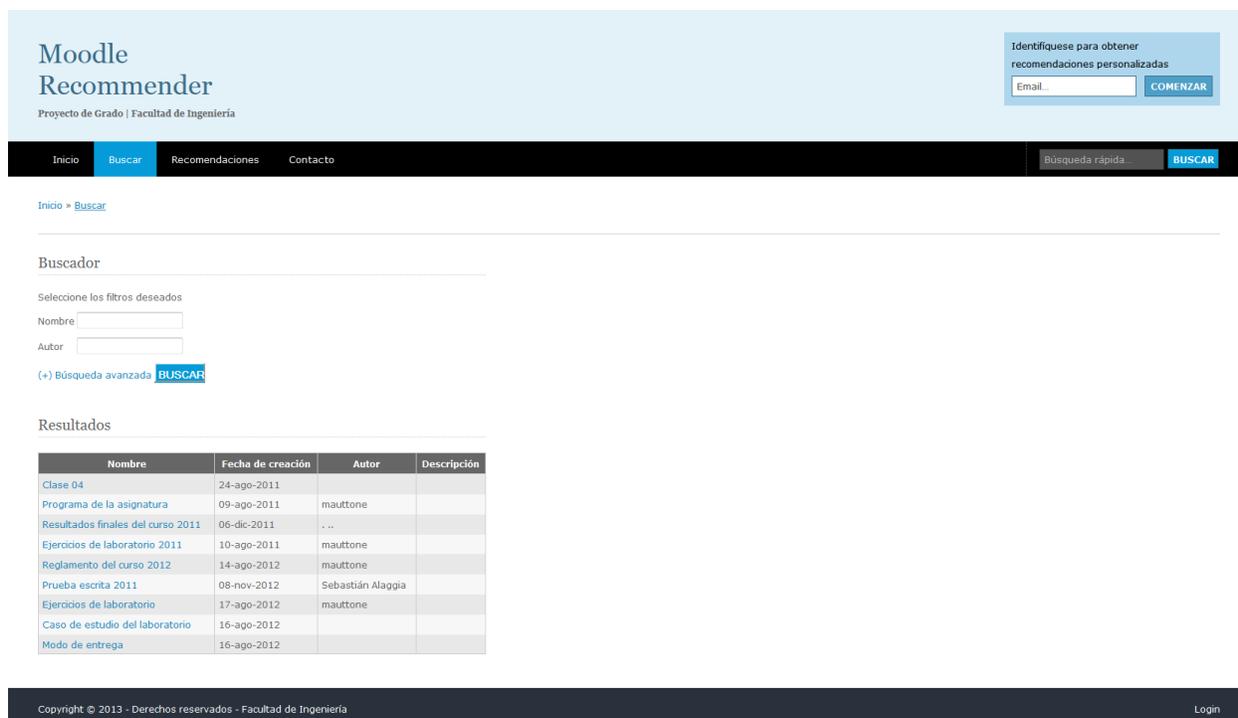
A continuación se presentan algunos ejemplos de las vistas generadas en la aplicación.



The screenshot shows the home page of the Moodle Recommender application. At the top left, the logo reads "Moodle Recommender" with the subtitle "Proyecto de Grado | Facultad de Ingeniería". A navigation bar below the logo contains links for "Inicio", "Buscar", "Recomendaciones", "Administración", and "Contacto". On the right side of the navigation bar, there is a search box labeled "Búsqueda rápida" and a "BUSCAR" button. The main content area features a large blue banner with the text "Sistema de recomendación de objetos de aprendizaje" and "Una guía para el aprendizaje", accompanied by a "Comenzar >" button. To the right of the banner is a 3D white figure holding a magnifying glass. Below the banner, the "Sobre este sitio" section provides introductory text about the project. The "Destacados" section on the right lists two PDF documents: "Clase 04" and "Clase 10". The footer contains copyright information for 2013 and a "Logout" link.

Figura 19 - Pantalla de inicio

La Figura 19 muestra la pantalla de inicio del sitio web creado. Esta pantalla contiene una descripción del sitio y en el lateral derecho, se presentan los recursos más destacados.



Moodle Recommender  
Proyecto de Grado | Facultad de Ingeniería

Identifíquese para obtener recomendaciones personalizadas  
Email... **COMENZAR**

Inicio **Buscar** Recomendaciones Contacto

Búsqueda rápida... **BUSCAR**

Inicio » [Buscar](#)

Buscador

Seleccione los filtros deseados

Nombre

Autor

(+) Búsqueda avanzada **BÚSCAR**

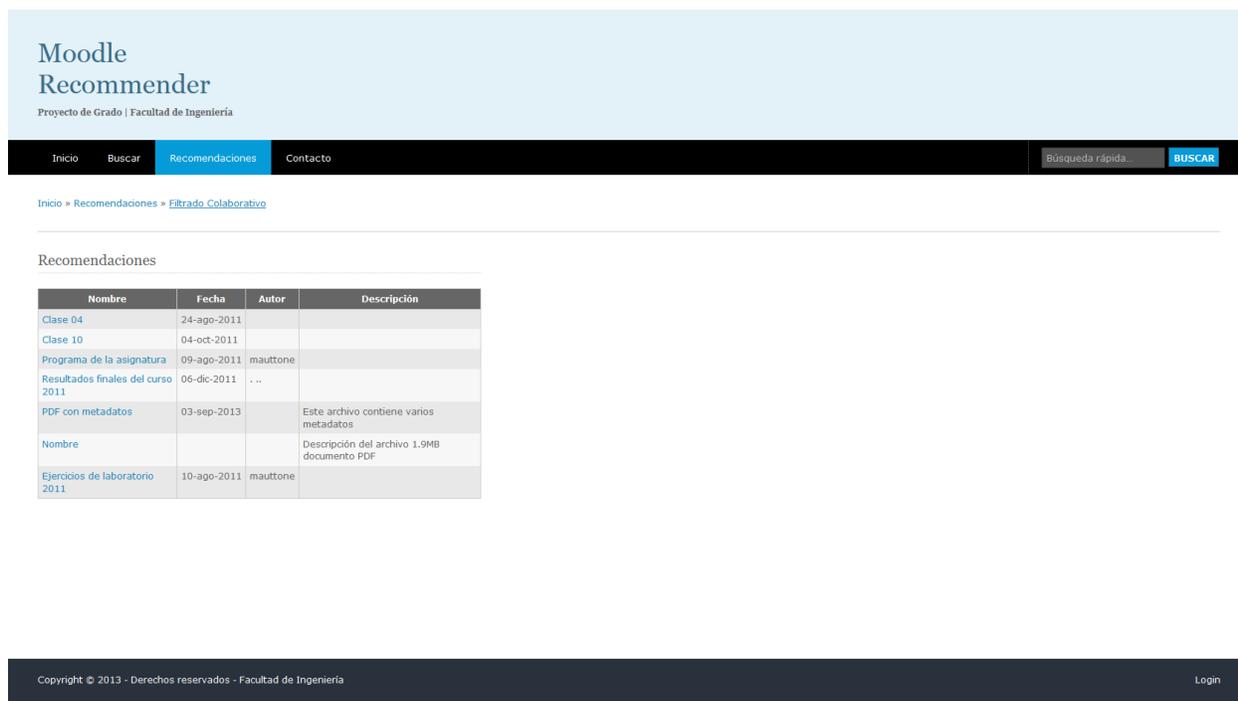
Resultados

Nombre	Fecha de creación	Autor	Descripción
Clase 04	24-ago-2011		
Programa de la asignatura	09-ago-2011	mauttone	
Resultados finales del curso 2011	06-dic-2011	...	
Ejercicios de laboratorio 2011	10-ago-2011	mauttone	
Reglamento del curso 2012	14-ago-2012	mauttone	
Prueba escrita 2011	08-nov-2012	Sebastián Alaggia	
Ejercicios de laboratorio	17-ago-2012	mauttone	
Caso de estudio del laboratorio	16-ago-2012		
Modo de entrega	16-ago-2012		

Copyright © 2013 - Derechos reservados - Facultad de Ingeniería Login

**Figura 20 - Pantalla de búsqueda de OA**

La Figura 20 muestra la pantalla de búsqueda de OA. Esta pantalla contiene filtros básicos como Nombre y Autor, y también permite una búsqueda avanzada a través de metadatos como fecha de subida, fecha modificado, etc. Para acceder a ella se debe acceder al menú Buscar.



Moodle Recommender  
Proyecto de Grado | Facultad de Ingeniería

Inicio **Buscar** **Recomendaciones** Contacto

Búsqueda rápida... **BUSCAR**

Inicio » Recomendaciones » [Filtrado Colaborativo](#)

Recomendaciones

Nombre	Fecha	Autor	Descripción
Clase 04	24-ago-2011		
Clase 10	04-oct-2011		
Programa de la asignatura	09-ago-2011	mauttone	
Resultados finales del curso 2011	06-dic-2011	...	
PDF con metadatos	03-sep-2013		Este archivo contiene varios metadatos
Nombre			Descripción del archivo 1.9MB documento PDF
Ejercicios de laboratorio 2011	10-ago-2011	mauttone	

Copyright © 2013 - Derechos reservados - Facultad de Ingeniería Login

**Figura 21 - Pantalla de recomendaciones de filtrado colaborativo**

En la Figura 21 se muestra la pantalla para obtener recomendaciones basadas en filtrado colaborativo. Para acceder a ella se debe navegar por el menú Recomendaciones y seleccionar uno de los algoritmos correspondientes de filtrado colaborativo.

Moodle Recommender  
Proyecto de Grado | Facultad de Ingeniería

Identifíquese para obtener recomendaciones personalizadas  
Email... COMENZAR

Inicio Buscar **Recomendaciones** Contacto

Búsqueda rápida... **BUSCAR**

Inicio > Recomendaciones > **Grupales**

Buscador

Seleccione los filtros deseados

Nombre:  **BUSCAR**

Grupos

Nombre	
Curso de Pruebas	<a href="#">Detalle</a>
Fundamentos de la Web Semántica	<a href="#">Detalle</a>
Prueba	<a href="#">Detalle</a>
Simulación a Eventos Discretos	<a href="#">Detalle</a>

Copyright © 2013 - Derechos reservados - Facultad de Ingeniería Login

**Figura 22 - Pantalla de recomendaciones grupales**

En la Figura 22 se muestra la pantalla para obtener las recomendaciones grupales. Seleccionando el grupo, se pueden observar dichas recomendaciones. Para acceder a esta pantalla se debe navegar por el menú Recomendaciones, seleccionando la opción: Recomendaciones grupales.

**Moodle  
Recommender**  
Proyecto de Grado | Facultad de Ingeniería

Identifíquese para obtener recomendaciones personalizadas

Email:  [COMENZAR](#)

---

[Inicio](#)
[Buscar](#)
[Recomendaciones](#)
[Contacto](#)

[BUSCAR](#)

Qué tan útil le resultó este documento?

☆☆☆☆☆

*Simulación a Eventos Discretos*

### Ejercicios de laboratorio

- Las entregas consisten en la realización de un informe de la resolución de cada ejercicio, siguiendo las pautas indicadas por los docentes. Para ejercicios de implementación, además se deben entregar los programas fuente y ejecutables.
- La primera entrega consta de los ejercicios 1 al 11. La segunda entrega consta de los ejercicios 12 al 16.
- La mayoría de los ejercicios consisten en la resolución de ejercicios del libro del curso, *Simulation modelling with Pascal*, de R. Davies y R. O'Keefe.

**Modelos de Simulación a Eventos Discretos**

*Ejercicio 1:*  
Resolver el ejercicio 1.5.3 partes c y d del libro del curso.

*Ejercicio 2:*  
Resolver el ejercicio 1.5.4 del libro del curso, para el caso de estudio del banco.

**Metodología de modelado**

*Ejercicio 3:*  
Resolver el ejercicio 2.13.1 del libro del curso.

*Ejercicio 4:*  
Resolver el ejercicio 2.13.4 del libro del curso.

**Implementación**

*Ejercicio 5:*  
Compilar y ejecutar el código provisto relativo al caso del hospital simple. Mostrar un breve ejemplo de la salida del programa, probando que la simulación funciona correctamente.

*Ejercicio 6:*  
Resolver el ejercicio 3.9.4 del libro del curso.

**Modelado de datos de entrada**

*Ejercicio 7:*  
Resolver el ejercicio 4.11.6 del libro del curso.

*Ejercicio 8:*  
Resolver el ejercicio 4.11.8 del libro del curso.

Información del documento

- Título: Ejercicios de laboratorio 2011
- Fecha: 10-ago-2011
- Autor: mauttone
- Descripción:

Haga clic en [ejercicios2011.pdf](#) para ver el archivo.

PDF con metadatos

Este archivo contiene varios metadatos

[Abrir documento »](#)

Resultados finales del curso 2011

[Abrir documento »](#)

Clase o6

[Abrir documento »](#)

Copyright © 2013 - Derechos reservados - Facultad de Ingeniería

Login

Figura 23 - Pantalla de detalle de OA

En la Figura 23 se muestra el detalle de un OA. En esta pantalla se muestra el contenido del OA junto con sus metadatos en la parte superior derecha de la pantalla. En la parte inferior se muestran OA relacionados con el OA principal. Esta pantalla permite ingresar un puntaje al OA que se está visualizando. El puntaje se asigna seleccionando la cantidad de estrellas (del 1 al 5) en la parte superior de la pantalla. A esta página se accede a través de la pantalla de búsqueda de OA, o a través de las recomendaciones integradas con la plataforma Moodle.

[Inicio](#) » [Contacto](#)

Escribe un comentario

Nombre (requerido)

Mail (requerido)

Enviar

Resetear formulario

### Figura 24 - Pantalla de contacto

La Figura 24 muestra la pantalla para ingresar comentarios hacia los administradores del sistema.

## 8. Manual de instalación

Para poner en funcionamiento el servicio de recomendaciones se deben seguir los siguientes pasos descritos a continuación.

### 1. Importar la máquina virtual

Desde el VirtualBox, seleccionar Importar servicio virtualizado y seleccionar el archivo *moodleRecommender\_server.ova*. De esta forma se crea la máquina virtual (VM) que contiene el software de base del servidor del sistema de recomendaciones.

### 2. Acceso al servidor

Una vez creada la VM, inicia y se accede al sistema operativo mediante el ingreso de las siguientes credenciales; nombre de usuario: *root* y contraseña: *d1!Abete4888*.

### 3. Configuración del servidor de base de datos

En la VM el servidor de base de datos MySQL ya está configurado y en ejecución. El mismo tiene creado el esquema con las tablas requeridas por la aplicación. El script que genera dicho esquema se adjunta bajo el nombre *moodleRecommender\_schema.sql*. Para acceder al servidor de base de datos se deben ingresar las siguientes credenciales; nombre de usuario: *root* y contraseña: *d1!Abete4888*.

Nota: Por motivos de seguridad, se crea otro usuario (usuario: *moodle*, contraseña: *recommender*) con privilegios mínimos para la conexión que utiliza la aplicación.

### 4. Configuración de servidor de aplicaciones

En la VM el servidor de aplicaciones Apache Tomcat está configurado, con el empaquetado de la aplicación (*RecommenderWeb.war*) copiado dentro del directorio de *webapps*. Se debe iniciar el servicio para que esté disponible. Para ello se debe ejecutar la siguiente línea desde una consola: `sh apache-tomcat/bin/startup.sh`

Opcional: Para ver el log, se debe ejecutar:

```
tail -f apache-tomcat-7.0.47/logs/catalina.out
```

### 5. Configurar reenvío de puertos VirtualBox

Una vez accesible al servicio desde la VM, se debe configurar un reenvío de paquetes dentro del equipo Host para que los pedidos que le lleguen sean reenviados a la VM. Esto se puede realizar fácilmente utilizando el VirtualBox. Para ello, seleccionar con el botón derecho la VM y acceder a Configuración. Allí ingresar a Red -> Reenvío de puertos. Finalmente, se debe ingresar la regla que realice el reenvío como se muestra en la Figura 25.

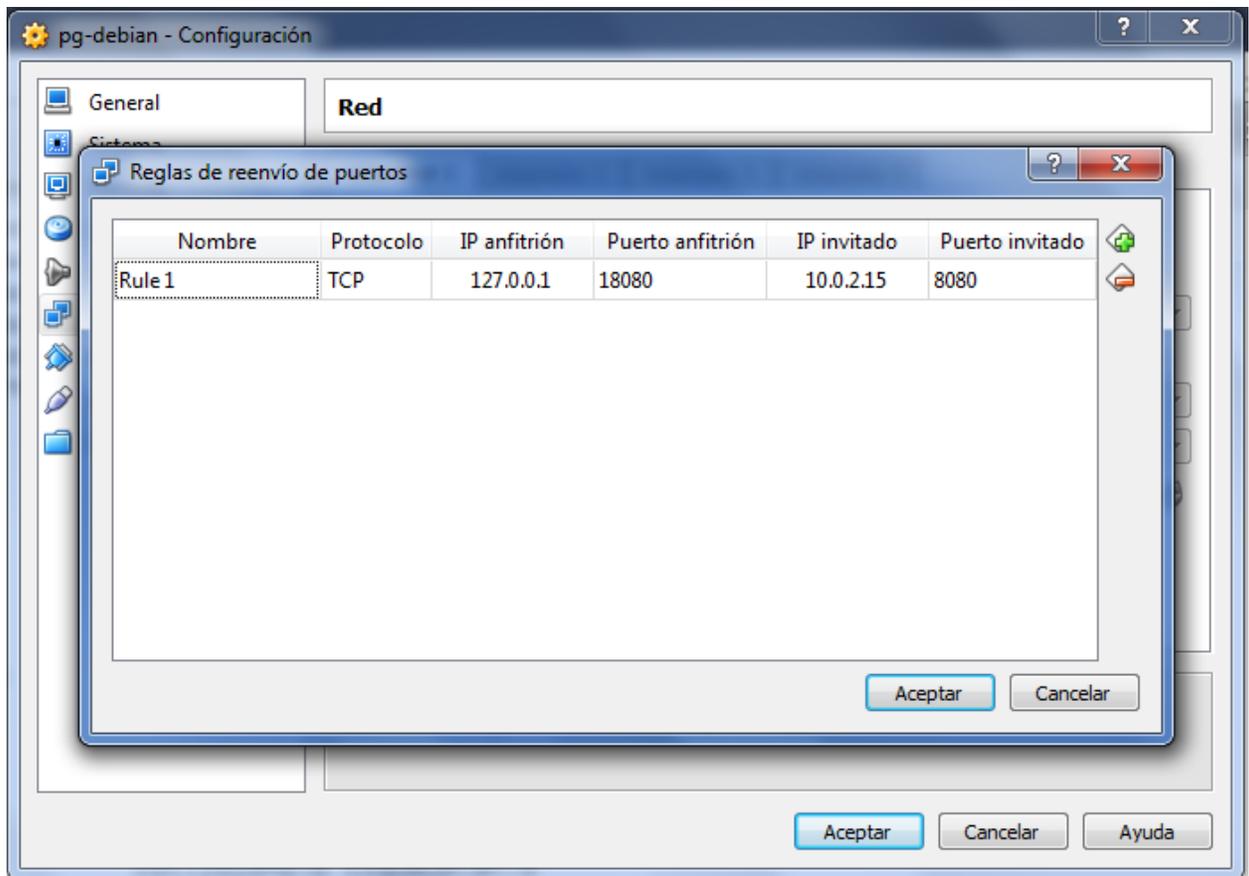


Figura 25 - Configuración reenvío de puertos en VirtualBox

#### 6. Recomendaciones integradas con Moodle

Finalmente, sólo falta ingresar el script que permite integrar las recomendaciones con la plataforma Moodle. Para ello, se debe ingresar al curso deseado dentro de Moodle y *Activar el modo de edición*, para habilitar las opciones de edición del curso. Luego, se debe seleccionar *Agregar un bloque* y finalmente, seleccionar el tipo de bloque HTML como se muestra en la Figura 26.

Luego se deben completar los datos del nuevo bloque como se muestra en la Figura 27. En el campo Contenido, se debe ingresar el texto del script que se adjunta en *moodleRecommender\_script.htm* actualizando la URL del servidor, de forma que indique la IP y puerto del equipo Host donde reside el servidor. Finalmente se debe indicar la ubicación y posición en la pantalla del curso donde se desea que se muestre el bloque.

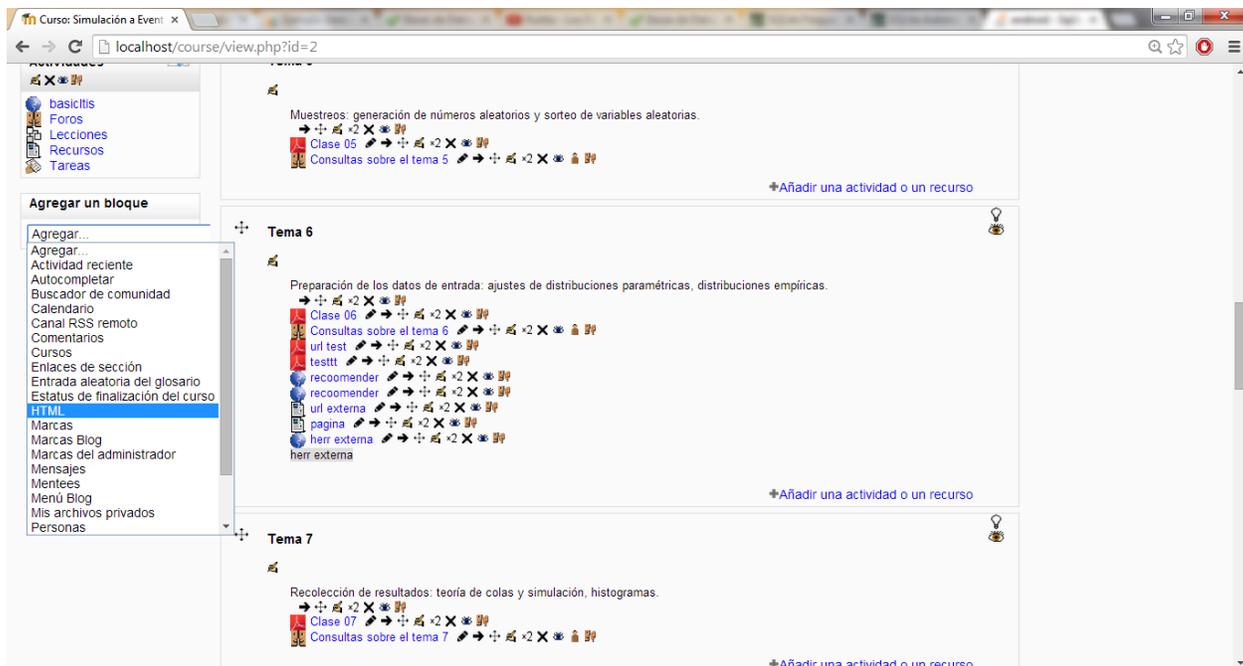


Figura 26 - Agregar bloque HTML en Moodle

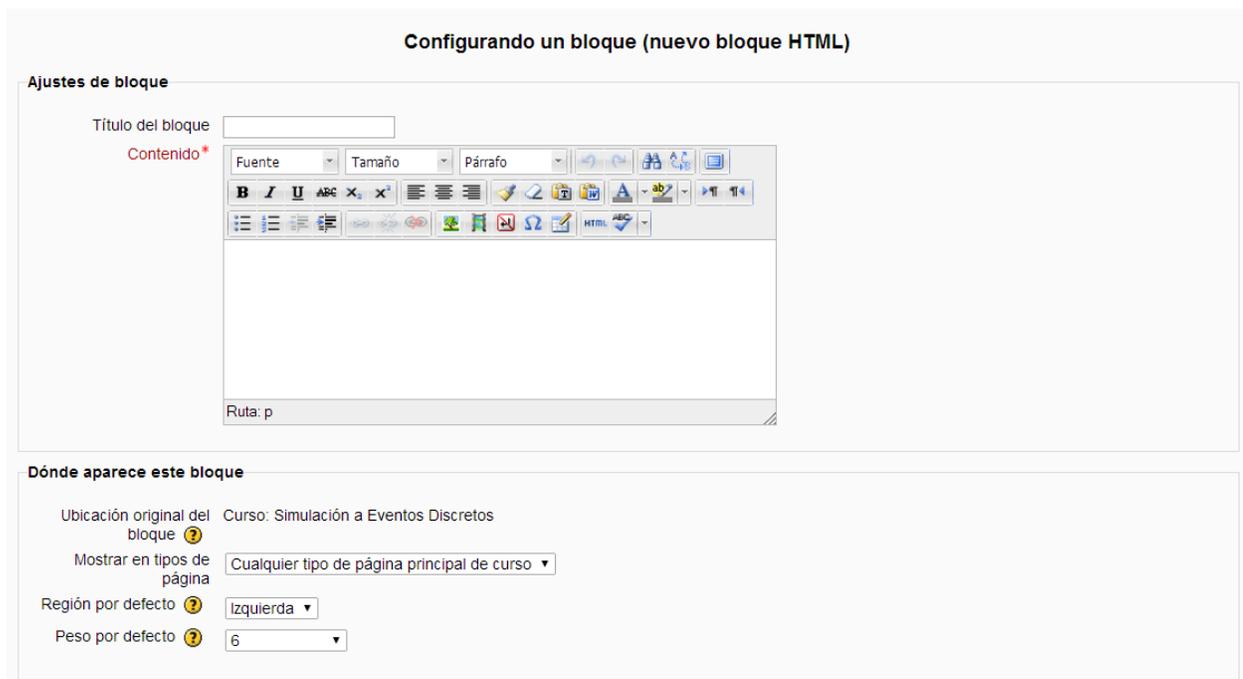


Figura 27 - Editar bloque HTML en Moodle

## 9. Detalle de OA extraídos del curso Fundamentos de la Web Semántica

Autor	Creador	Descripción	Creación	Modificado	Archivo	Título
NISO		Understanding Metadata	02/06/04	23/06/04	UnderstandingMetadata.pdf	
user		Ontology Networks	16/09/13	16/09/13	ConceptosBasicosMetadatos.pdf	
rmoz		Metadatos-eGob	15/04/08	15/04/08	MetadatosGov.pdf	
		Transparencias Tercer Encuentro			TercerEncuentro.pdf	
usuario	Microsoft Word 2013	Ejemplos de metadatos estructurales	04/10/13	04/10/13	EjemploMetadatosEstructurales.pdf	
	TeX output 2010.08.31:1536	Artículo sobre Arquitectura de la Web Semántica	31/08/10	31/08/10	final-semantic-web-architecture.pdf	
	QuarkXPress	Material sobre XML	17/05/05	17/05/05	ch2_FINAL.pdf	
	dvips(k) 5.96 Copyright 2005 Radical Eye Software	reasoning-web.dvi	15/04/09	15/04/09	foundationsRDFDB.pdf	
			25/05/99	19/05/04	p351-hammer.pdf	
	Acrobat PDFMaker 8.1 for PowerPoint		07/11/13	07/11/13	IntroduccionRDF.pdf	
Patricia Shaw	pdftopdf filter	ppts de apuntes de clase de la Prof. Patricia Shaw.	05/02/09	05/02/09	RDFS-Apuntes-Patricia-Shaw.pdf	ais13-rdfs.key
	dvips(k) 5.96 Copyright 2005 Radical Eye Software		25/05/09	25/05/09	simple-and-efficient-RDFS.pdf	
		Presentación del quinto encuentro presencial del curso 2013.			Ontologias.pdf	
	TeX output 2002.01.31:1820	Capítulo 2 Description Logics Handbook: Theory, Implementation, and Applications. Editores Franz Baader, Diego Calvanese, Daniele Nardi y Peter F. Patel-Schneider. 2003.	31/01/02		Capítulo del Handbook of Description Logic.pdf	
		Capítulo 5 (OWL Formal Semantics) de Foundations of Semantic Web Technologies. Pascal Hitzler, Markus Krötzsch y Sebastian Rudolph. 2009 Chapman & Hall/CRC			Capitulo%205.pdf	
inco	Acrobat PDFMaker 8.1 for PowerPoint	Notas de clase sobre Semantic aa la Tarski, con ejemplo de cubos y el diagrama de conceptualizacion usando ontologia	26/11/13	26/11/13	Tarski.pdf	Transformación de Esquemas

inco	Acrobat PDFMaker 8.1 for PowerPoint		27/11/13	27/11/13	SEAN-Ejemplos-inferencias-owl.pdf	Transformación de Esquemas
------	-------------------------------------	--	----------	----------	-----------------------------------	----------------------------

Tabla 5 - Recursos extraídos del curso Fundamentos de la web semántica