

Mejora de la búsqueda de cursos de acuerdo al perfil de usuario en Moodle

Cristina Bender, Matías Asás, Claudia Deco, Ana Casali

bender@fceia.unr.edu.ar, asasmatis@gmail.com, deco@fceia.unr.edu.ar, acasali@fceia.unr.edu.ar

Facultad de Ciencias Exactas, Ingeniería y Agrimensura,

Universidad Nacional de Rosario,

Av. Pellegrini 250, Rosario, República Argentina

Resumen

En la actualidad el aprendizaje a distancia está cada vez más difundido, por lo cual se requieren nuevas técnicas de almacenamiento, búsqueda y recuperación de documentos que respondan eficientemente a esta realidad. Es necesario para ello, explotar distintos datos de contexto, como las características y las preferencias del usuario. Existen diferentes plataformas de aprendizaje que intentan facilitar la interacción entre el alumno y el docente, a fin de hacer confortable este aprendizaje a distancia. La propuesta de este trabajo es mejorar la búsqueda de cursos en estas plataformas aprovechando datos de contexto para personalizar los resultados, presentando al alumno únicamente los cursos que sean considerados de su interés y ordenados de acuerdo a sus preferencias. Con ese fin, se intenta aprender automáticamente de los casos históricos disponibles, para lo cual se utiliza el algoritmo de k-vecinos más cercanos con pesos. Las experimentaciones realizadas en la plataforma Moodle, muestran que se obtienen valores de 95% para la Precisión de la búsqueda, de 90% para el Recall y de casi el 95% en la evaluación de la Calidad del ordenamiento.

Palabras claves: Búsqueda personalizada, Ordenamiento de resultados, Plataforma Moodle, Aprendizaje automatizado.

Introducción

En tiempos donde el aprendizaje a distancia está cada vez más difundido, en donde las diferencias espaciales y temporales entre

educando y educador tienden a desaparecer, se hacen necesarias nuevas formas de almacenamiento, búsqueda, recuperación y presentación de documentos. Los métodos de enseñanza y aprendizaje se han ido adaptando a estos cambios. Por ejemplo, existen plataformas de aprendizaje en donde alumnos de las más diversas características acceden a la misma información y tienen la posibilidad de acceder al conocimiento sin la necesidad de la presencia física directa. Estas plataformas de aprendizaje son accedidas por gran cantidad alumnos, y contienen gran cantidad de información que es ofrecida a los mismos. Entre estas plataformas se encuentra Moodle, que brinda una variada cantidad de herramientas para asistir a la enseñanza y la gestión de cursos. Pero Moodle, al igual que la mayoría de las plataformas actuales, no aprovecha en su totalidad la información implícita que contiene como, por ejemplo, las características de los alumnos que la utilizan. La información, cursos en el caso de Moodle, es presentada para todos los alumnos de la misma manera, aunque las preferencias de un alumno pueden ser diferentes a las de otro de características totalmente opuestas. La propuesta de este trabajo es una solución que ofrece al usuario los cursos buscados en una plataforma de aprendizaje en un determinado orden, de acuerdo con sus necesidades y cualidades particulares, aprovechando el conocimiento global e histórico que se dispone en la plataforma. Para lograr esto, es necesario contar con una manera de almacenar, aprender de, y recuperar la información disponible. Para cumplir este objetivo se propone la utilización de técnicas de aprendizaje automatizado y la inclusión o adaptación de ciertas funcionalidades del

sistema de administración de cursos. Se eligió la plataforma Moodle por ser de código abierto y de estructura modular.

En la próxima Sección se presenta el problema. Luego se plantea y describe la solución propuesta y su implementación en la plataforma Moodle. Se analizan resultados de experimentaciones realizadas y finalmente se presentan las conclusiones.

Descripción del problema

Un LMS (Learning Management Systems) es un sistema basado en web utilizado para crear, publicar, administrar y almacenar cursos [1]. Los contenidos usualmente se almacenan como objetos descritos e identificables de forma única. Dentro de las plataformas abiertas se encuentra Moodle¹. Esta plataforma se eligió para este trabajo dado que es de código abierto; su estructura modular permite la introducción de modificaciones de manera transparente y sencilla; es segura, altamente disponible, escalable; está bien documentada; el uso de XML para importar/exportar información la hace interoperable; y es ampliamente utilizada. La estructura modular de Moodle es bastante clara y se puede separar en tres módulos fundamentales: *Usuarios* que está encargado del control de los usuarios de la plataforma (inicio y fin de sesión, control de acceso de acuerdo a diferentes niveles como alumno o profesor, y otras operaciones de administración); *Cursos* que contiene las funciones asociadas a los diferentes cursos que pueden ser tratados por la plataforma; y *Actividades* que contiene las principales funcionalidades que ofrece esta plataforma. Los datos están almacenados en una base de datos relacional.

La gran aceptación que tienen actualmente las diferentes plataformas de aprendizaje hace que gran cantidad de alumnos las utilicen.

Algunos de estos alumnos comparten características y preferencias similares entre ellos, pero otros alumnos poseen perfiles muy distintos. Los diferentes cursos ofrecidos tendrán un nivel de aceptación mayor o menor, según cuánto se acerquen las características del curso al perfil del usuario que lo busque. En este contexto, se hace imprescindible aprovechar información de usuarios y cursos, implícita dentro de la plataforma, para personalizar lo más posible la respuesta a la búsqueda del alumno, y esto de manera totalmente automática. Pero esto es algo que ni Moodle ni ninguno de los LMS más reconocidos ofrecen.

Solución propuesta

Se propone una arquitectura que le proporciona al usuario de una plataforma de aprendizaje los cursos que estén de acuerdo con sus necesidades y cualidades particulares, ordenados por mayor afinidad al mismo aprovechando el conocimiento global e histórico que se dispone en la plataforma. Esta arquitectura debe recibir y almacenar datos referentes al perfil de los alumnos y datos de las características de los cursos. Con estos datos más la información que se obtenga de los resultados de búsquedas anteriores, también almacenados en la plataforma, debe poder predecir y generar nuevos resultados para las nuevas búsquedas, y que estos resultados sean sensibles a las características particulares del alumno que la realiza.

Para esto, se agregan ciertas funcionalidades a la plataforma de aprendizaje. La primera, *Perfil de usuario*, proporciona al administrador de usuarios la posibilidad de introducir a cada usuario los valores de las características que definen su perfil personal. *Características de cursos* permite al administrador de cursos, al momento de dar de alta un nuevo curso, poder introducir las características que represente a cada curso. Un nuevo módulo, *Evaluación de Cursos*, posibilita al alumno evaluar cada curso en el que participa, estableciendo su grado de

¹ <http://moodle.org>

preferencia por un curso en particular. El *Módulo de Aprendizaje* recibe la información generada por la utilización del sistema, y tiene la suficiente inteligencia para obtener resultados y conclusiones sobre ésta. El módulo de *Ordenamiento y Filtrado* presenta al alumno cursos que coinciden con su interés de búsqueda, ordenándolos de acuerdo a sus preferencias, presentando en las primeras posiciones los cursos que le pudiesen resultar de mayor interés. A continuación se describen detalladamente estas funcionalidades y su implementación en Moodle.

Modelado del Perfil de Usuario

El objetivo es modelar el perfil del alumno para que el sistema tenga la posibilidad de relacionar al alumno con los cursos de su preferencia. Esto hace indispensable poner especial atención al modo en que se representan estas características, ya que esto condiciona a todo el modelo que se propone. Por esto, hay que evaluar la conveniencia de representar al perfil del alumno como un patrón único dependiente del usuario, que contenga características que incluyan sus datos personales (por ejemplo, clave de usuario o número de documento) para que sea unívocamente identificable; o evaluar si es más conveniente crear un perfil independiente del usuario en cuestión que pueda ser compartido con otro usuario que posea las mismas características. Se observa que tener un perfil propio de cada usuario no aporta información para la clasificación de un curso, ni ayuda a deducir si este curso es de relevancia o no para un dado usuario. Esto también genera una gran cantidad de datos, lo cual produciría importantes disminuciones a nivel de eficiencia de cualquier algoritmo que sea necesario utilizar. Por otro lado, tener un perfil general, además de disminuir la cantidad de datos en el sistema, brinda la posibilidad de generalizar. En otras palabras, al existir la posibilidad de que dos usuarios distintos tengan el mismo perfil, se puede trabajar con la siguiente suposición: “Si al alumno A le agrada/desagrada el curso C, y el

alumno B tiene el mismo perfil que A, entonces al alumno B también le agrada/desagrada el curso C”. Esta suposición es muy importante ya que es la que permite generalizar las preferencias de los usuarios sobre los cursos, y la que brinda la posibilidad de pensar algún algoritmo inteligente que tenga la capacidad de suponer de manera correcta respecto a las preferencias de los usuarios sobre los cursos.

Para lograr esto, se incluyen en el perfil del alumno las siguientes características:

- *Nivel académico.* Representado por valores numéricos entre 0 y 100, donde 0 representa al alumno inicial sin conocimientos previos y 100 lo opuesto.
- *Estilo de aprendizaje.* Es la forma en que una persona resuelve una situación relacionada a las tareas de aprendizaje. Ejemplos de estilos de aprendizaje son holístico y analítico. Esta característica también se representa con valores numéricos entre 0 y 100, donde los valores menores representan a alumnos de estilo holístico, y a medida que aumenta el valor, se encuentran los alumnos de características más analíticas.
- *Idioma/s de preferencia.* Un alumno de un país de habla hispana, puede preferir cursos en inglés. Esto se modela con un rango de números, con valores posibles entre el 0 y el 100, donde el menor valor representa preferencia estricta por el español, el 100 preferencia estricta por el inglés, pero también existe la posibilidad de elegir un valor intermedio, por ejemplo 50, lo que significa que para el alumno no hay diferencia si el curso se encuentra en inglés o en español.

Estas características son tres atributos (P_1 , P_2 y P_3 respectivamente), en la tabla de datos de los alumnos, accesibles con el nuevo módulo Perfil de Usuario.

Modelado del curso

Existen diversas características que pueden ser tomadas en cuenta en el momento de realizar la representación. ([2], [3]) tratan este tema y proponen diferentes atributos representativos. Considerando que se desea analizar la cercanía de las características de un curso a las del usuario que está utilizando el sistema, y teniendo en cuenta las características que se definieron al modelar el perfil del usuario, se proponen las siguientes para los cursos:

- *Nivel de Complejidad.* Se representa con valores numéricos entre 0 y 100. A mayor valor, mayor el nivel de complejidad en términos de aprendizaje que tiene el curso.
- *Idioma.* Es un valor que representa el idioma del curso. Los valores posibles son 0, 50 y 100, para Español, Portugués e Inglés respectivamente. Estos idiomas sirven de ejemplo y podrían agregarse alternativas según la necesidad.
- *Extensión.* Un curso puede ser muy largo o muy corto, contener muchos datos y recursos o no contener ninguno. Esta característica dimensiona esta situación y se representa con valores numéricos entre 0 y 100.
- *Práctico.* Representa si el curso caracterizado tiene una tendencia práctica o tiene características más bien teóricas. Los valores están en el rango 1-100.

Definidas estas características para un curso, se puede observar que:

- Un curso tiene un único conjunto de características que lo caracterizan.
- Un conjunto de características puede tener asociado más de curso.

La conjunción de estas dos afirmaciones, hace claro que el aprendizaje de este sistema debe estar orientado a las características y no a los cursos. Trabajar y generalizar con respecto a

conjuntos de características y no de cursos particulares facilita y mejora los resultados del aprendizaje del sistema. Por ejemplo, si a un alumno determinado le agrada un curso en particular, se puede concluir que otro curso con idénticas características también será de su agrado. Esta conclusión no podría tomarse si se trabajase sobre cursos en particular.

Dado que la plataforma Moodle, a diferencia de lo que sucede con la información del alumno, no posee ninguna manera de incluir características de los cursos, se modifica la plataforma para dar de alta y realizar cambios de atributos de los cursos correspondientes a sus características. En la Figura 1 se muestra la pantalla de la extensión realizada, donde se mantiene la estética y el formato del resto de las funcionalidades de Moodle.

Figura 1: Alta de Características de Cursos

Para almacenar estas cuatro características de los cursos se extiende el modelo de datos de Moodle agregando una tabla que las contiene.

Módulo de Evaluación de Cursos

El algoritmo de aprendizaje deberá asociar los alumnos con determinados atributos a los cursos de ciertas características. Es necesario saber que a un alumno le ha agradado el curso en el cual ha participado y, si fue de su agrado, en qué medida. Para esto no es suficiente trabajar con métodos estadísticos en términos del log del sistema, como por ejemplo saber a qué cursos ha asistido o han asistido alumnos de características similares, porque el o los alumnos pueden haber participado en un curso pero no les fue de su agrado, o no les resultó de su interés por

completo. Se hace clara la necesidad de que el sistema le brinde al alumno la posibilidad de evaluar el curso en el que se encuentra participando.

Moodle en su forma estándar no brinda esta funcionalidad. Por esto, se desarrolla un módulo de extensión de la aplicación, y se agrega una tabla en donde se almacenan los resultados de las evaluaciones que realiza el alumno a los cursos en los que ha participado. Esto lo realiza el alumno introduciendo un valor de evaluación, numérico en el intervalo [1;100], donde 100 se aplica a un curso que cubre totalmente sus expectativas y 0 es el caso contrario. Si ningún alumno con el perfil del alumno que se encuentra evaluando el curso, ha evaluado anteriormente dicho curso, el valor introducido se almacena sin modificaciones. En cambio si otro alumno del mismo perfil evaluó anteriormente este mismo curso, el valor almacenado es el valor medio entre el valor anterior y el valor nuevo.

Módulo de Aprendizaje

Se encarga de entender cómo todos los datos se conjugan y se puede aprender de ellos. El método de aprendizaje que se utiliza es el de k-vecinos más cercanos. Este algoritmo trabaja a través del almacenamiento de ejemplos de entrenamiento y aprende de ellos con el fin de poder deducir los valores para los nuevos casos que se le presenten. Las ventajas de utilizarlo son: el entrenamiento es muy rápido, es capaz de aprender funciones objetivo muy complejas, y no tiene pérdida de información. Las desventajas son: es lento al momento de consultas, y es fácilmente engañable por atributos irrelevantes. Como se observa, las ventajas que presenta este método son muy importantes y significativas para el buen desempeño del sistema. Por otro lado las desventajas no representan dificultades significativas y son tratables para que no repercutan en la eficiencia y buen desempeño del mismo. Estas dos conclusiones son determinantes al momento de seleccionar el método de k-vecinos más cercanos como

método de aprendizaje.

La propuesta es almacenar los casos en vectores de la forma:

$$\langle P_1, P_2, P_3, C_1, C_2, C_3, C_4, V \rangle$$

donde los componentes P_i con $i:1,2,3$ del vector corresponden al perfil del alumno y los componentes C_j con $j:1,2,3,4$ se corresponden a las características de los cursos. El componente V puede tomar valores 0, 1 o NULL. Si $V=0$ significa que todos los alumnos de perfil $\{P_i\}$ han evaluado como no de su agrado a los cursos de características $\{C_j\}$. Si $V=1$ todos los alumnos de perfil $\{P_i\}$ han tenido buena aceptación y les han resultado de interés los cursos de características $\{C_j\}$. Si $V=NULL$ significa que aún no ha ocurrido la relación “alumnos de perfil $\{P_i\}$ - cursos de características $\{C_j\}$ ”.

Estos vectores combinación se crean al momento del alta de un nuevo alumno o de un nuevo curso y no al momento de la búsqueda, debido al costo de eficiencia que conlleva, combinando las características del nuevo alumno o curso con todos los juegos de características de los cursos-alumnos preexistentes, y colocando el componente V en NULL. El algoritmo de aprendizaje trabaja con los datos convenientemente preparados y ordenados, para que su tratamiento y búsquedas sean lo más eficientes posibles. El algoritmo de k-vecinos más cercanos recupera entonces el conjunto de los vectores de la forma $(\{P_i\}, \{C_j\}, V)$ y divide este conjunto en dos partes. El primer subconjunto se encuentra formado por los vectores que tienen la componente V con valor 0 ó 1, es decir la combinación alumno-curso que ha sido evaluada en alguna oportunidad. El segundo subconjunto se encuentra formado por los vectores que tienen la componente V con valor NULL. Es decir, todas las combinaciones alumno-curso que hayan sido dadas de alta en el sistema y que nunca hayan recibido evaluación alguna. El primer subconjunto es utilizado como conjunto de

entrenamiento y el segundo como conjunto a ser clasificado. El resultado de la ejecución del algoritmo es un tercer conjunto, equivalente al segundo, pero con valores 0 o 1 en la última componente, de acuerdo a lo que haya aprendido el algoritmo en función de los datos de entrenamiento.

Módulo de Ordenamiento y Filtrado

El alumno realiza una búsqueda de cursos introduciendo palabras clave. En función del perfil de este usuario y considerando los vectores almacenados, se filtran los cursos de los cuales el sistema tenga información fehaciente de que un alumno de características parecidas al alumno que actualmente se encuentra realizando la búsqueda ya haya evaluado este curso como no de su agrado. Esto es, se buscan los vectores donde $V=0$. Luego de haber hecho este filtrado, el sistema cuenta con un listado de vectores, donde la preferencia está ya confirmada ($V=1$) o se encuentra sin evaluar ($V=NULL$). Con respecto a los que se encuentran sin evaluar, son analizados con el algoritmo de k-vecinos más cercanos, que predice según el conocimiento que éste tiene de las preferencias del alumno, si el curso será de su agrado o no. Por último, de los cursos que se tienen confirmados como del agrado del alumno más los obtenidos por el algoritmo se obtiene el listado final de cursos a presentar al alumno. Luego se los ordena, de forma que los de mayor relevancia para el usuario estén en los primeros lugares.

Para el ordenamiento de los resultados obtenidos hay que tener en cuenta que los cursos listados provienen de dos fuentes diferentes: por un lado los cursos que se encuentran evaluados por el usuario o algún usuario con el mismo perfil, y por el otro lado los que el sistema a través de su conocimiento del perfil del alumno interpreta que son del agrado del alumno. Es claro que el primer grupo debe tener prioridad por sobre el segundo, ya que éstos han sido explícitamente evaluados. Dentro de este primer grupo, a su

vez también hay un orden relativo entre los cursos, ya que no es lo mismo un curso que fue consentido con un 100% de aprobación que otro que lo fue con un 60% (se considera del agrado del alumno si su evaluación es superior o igual al 50%). Este valor porcentual de aprobación es el valor introducido por el/los alumno/s vía el módulo *Evaluación de cursos* descrito, y es el que se utiliza para ordenar el resultado de acuerdo a su puntuación en la evaluación. En la Figura 2 se diagrama este proceso de filtrado y ordenamiento.

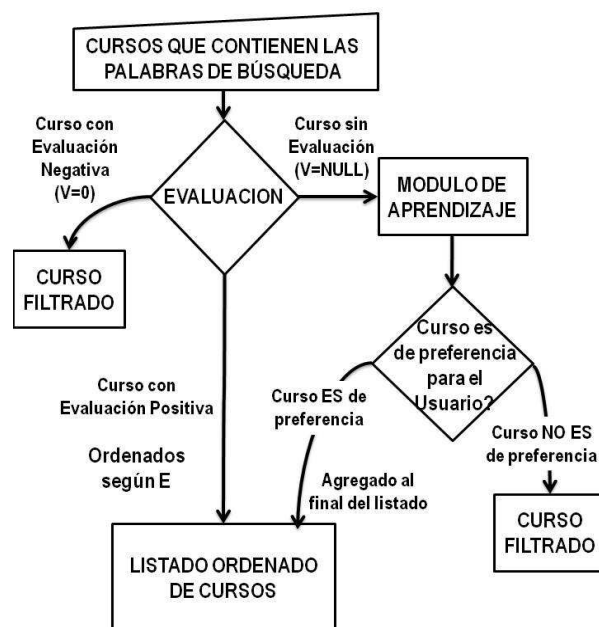


Figura 2: Ordenamiento y Filtrado

Parámetros del algoritmo de aprendizaje

La ejecución del algoritmo k-vecinos más cercanos se encuentra directamente afectada por varios parámetros: la cantidad de casos que se utilizan para entrenamiento, para validación y para test; la cantidad de clases y de atributos; la cantidad óptima de vecinos a considerar y pesos para cada uno de los atributos. El parámetro cantidad de clases vale 2 dado que son dos las posibilidades de clasificación (0 y 1). La cantidad de atributos corresponde a la suma de los atributos del alumno (3 atributos) y de los cursos (4 atributos) que hacen un total de 7. Para

entrenar y validar al algoritmo y para obtener los otros parámetros se ha generado un conjunto de prueba artificial con 300 vectores.

En primer lugar, se calcula la relación entre la cantidad de casos para entrenamiento y la cantidad de casos para validación. Para esto, se separa el conjunto de datos en dos subconjuntos: uno para entrenamiento y el otro para validación. El error mínimo se obtiene con la utilización del 85% de los datos para entrenamiento y el 15% restante para validación. Esta relación 85/15 se utiliza para obtener el resto de los parámetros. Para determinar el número óptimo de vecinos se utiliza el enfoque de validación cruzada K-Fold [4]. Siguiendo esta técnica con $K=7$, y realizando dos repeticiones, con la cantidad de vecinos entre 0 y 100, se obtiene que la cantidad óptima de vecinos que minimiza el error es 3.

Uno de los principales inconvenientes que presenta el algoritmo de k-vecinos es que la distancia entre instancias se calcula en base a todos los atributos de la instancia. Si se tiene una instancia de muchos atributos, la distancia entre vecinos puede ser dominada por un gran número de atributos irrelevantes o poco significativos. Un enfoque interesante para sortear este inconveniente, es darle diferentes pesos a cada atributo cuando se calcula la distancia entre dos instancias [5]. Esto es equivalente a estirar los ejes en el espacio euclídeo, acortando los ejes que correspondan a atributos menos relevantes y alargando los que corresponden a atributos significativos. La cantidad por la cual cada eje debe ser acortado o alargado también puede ser determinada utilizando la validación cruzada K-Fold. Es decir, seleccionar un subconjunto aleatorio de los datos de entrenamiento disponibles, y luego determinar el conjunto de atributos que lleven al mínimo el error de clasificación de los restantes ejemplos. Repitiendo este proceso múltiples veces los valores estimados para estos factores de multiplicación pueden ser determinados más adecuadamente. Con todas las combinaciones

de valores 0, $\frac{1}{2}$ y 1 para los pesos de los atributos, utilizando K-Fold y el conjunto de prueba de 300 casos con 7 atributos, se obtiene que el error mínimo se produce con los siguientes valores: $z_1=0.5$; $z_2=0.5$; $z_3=1.0$; $z_4=0.5$; $z_5=1.0$; $z_6=0.0$; $z_7=1.0$. Utilizando estos pesos el error es igual a 0.89 y es mínimo. Entonces, los ejes correspondientes a los atributos 3, 5 y 7 se mantienen, mientras que los restantes se acortan a la mitad. Esto puede interpretarse como que la distancia entre dos puntos con respecto a los atributos 3, 5 y 7, es el doble de significativa que la distancia del resto de los atributos. Teniendo en cuenta que dos de estos tres valores corresponden al idioma nativo del alumno (z_3) y al idioma en que se encuentra el curso (z_5), no sorprende que sean más significativos que otros atributos tal como la complejidad del curso (z_4). Es decir, es el doble más importante para clasificar si un curso le gusta a un alumno que éste sea en un idioma en particular que si es fácil o difícil o largo. Por otra parte también se debe remarcar el caso del atributo Extensión del Curso (z_6), cuyo peso ideal ha tomado el valor 0. Esto indica que este atributo directamente no es relevante para la clasificación. Este proceso de alargue o encogimiento de ejes, logra disminuir el impacto negativo que tiene los atributos poco o directamente no relevantes.

Evaluación de la Propuesta

El objetivo de la evaluación es analizar los resultados obtenidos con las funcionalidades propuestas en este trabajo, y compararlos con los resultados obtenidos con el uso de la plataforma Moodle sin estas funcionalidades.

La implementación de las extensiones se ha realizado en Moodle manteniendo la robustez de la plataforma, con una disminución, pero dentro de valores aceptables, de la performance del sistema. Con el objetivo de analizar la eficiencia y performance de la implementación, se genera un conjunto de datos artificial grande de 15000 registros. Este conjunto de datos es el equivalente a tener

5000 alumnos y 3000 cursos, todos ellos con características y perfiles estrictamente diferentes entre sí. Ésta es una situación exagerada, ya que lo usual es que muchos de los alumnos compartan perfiles similares, al igual que muchos cursos compartan características parecidas, pero es un conjunto válido para realizar las pruebas y analizar los tiempos, en un posible peor caso. Se realizan consultas en modo normal o estándar, es decir sin las mejoras implementadas, y en modo avanzado (consultas con el uso de la arquitectura propuesta), en un total de 15 repeticiones. La cantidad de datos en la base de datos de la plataforma se incrementa progresivamente. Los resultados obtenidos se presentan en la Tabla 1.

Cantidad de Registros	Valor promedio de consulta avanzada	Valor promedio de consulta estándar
300	0,14	0,10
1000	0,27	0,09
5000	1,05	0,10
15000	1,30	0,11

Tabla 1: Tiempos de ejecución en segundos

Como se observa, a medida que la cantidad de registros crece, también lo hace el tiempo de respuesta de la consulta avanzada. Esto no ocurre en la consulta realizada con la instalación estándar de Moodle donde los tiempos se mantienen constantes e independientes de la cantidad de datos. La dependencia del tiempo de respuesta con respecto a la cantidad de registros es consecuencia del uso del algoritmo de k-vecinos. Si bien el tiempo de respuesta con 15000 registros, se puede decir un peor caso, no es bueno, se encuentra en un nivel tolerable. Por otro lado, el tiempo de respuesta con un conjunto de datos en condiciones normales, es bastante bueno.

Evaluación de la calidad de la recuperación

Un programa es funcionalmente correcto si se comporta de manera satisfactoria de acuerdo a los requerimientos funcionales de los servicios que debería proveer. Se evalúa, en

primer lugar, la calidad de la recuperación de los cursos; y en segundo lugar, el orden que se le da a los mismos.

Se pretende que al realizar una búsqueda de cursos en la plataforma, ésta devuelva un resultado personalizado, es decir, devuelva únicamente los cursos que puedan ser de relevancia para el perfil del alumno que realiza la consulta, y no devuelva otros cursos. Esto puede evaluarse utilizando dos indicadores: Precisión, que es la cantidad de cursos relevantes del conjunto de cursos devuelto; y Recall, que es la proporción de cursos relevantes que son recuperados.

Para realizar la experimentación y poder tomar estos indicadores, se crean 6 alumnos con perfiles disímiles entre sí, y con cada uno de ellos se realizan 10 búsquedas diferentes. Cada iteración incluye los siguientes pasos: realizar la búsqueda estándar (sin parámetro "Búsqueda avanzada"), tomar la cantidad total de cursos recuperados, realizar la búsqueda personalizada (con parámetro "Búsqueda avanzada"), tomar la cantidad total de cursos recuperados, calcular Precisión y Recall. Siguiendo este procedimiento para cada uno de los alumnos, con 10 repeticiones, el promedio de los resultados obtenidos se muestra en las Tablas 2 y 3, donde la columna Búsqueda Estándar contiene los valores correspondientes a la búsqueda normal de Moodle y la columna Búsqueda Mejorada tiene los resultados utilizando la búsqueda que considera las preferencias del alumno.

Alumno	Precisión	
	Búsqueda estándar	Búsqueda mejorada
Alumno Test 1	0,3952381	1
Alumno Test 2	0,31309524	0,91666667
Alumno Test 3	0,42857143	1
Alumno Test 4	0,4047619	1
Alumno Test 5	0,725	0,85119048
Alumno Test 6	0,58571429	0,95833333
Valor Promedio	0,47539683	0,95436508

Tabla 2: Valores de Precisión

En la Tabla 2 se puede observar que el nivel de Precisión se incrementa drásticamente, duplicando el valor de precisión normal de la

plataforma. Así, el nivel de Precisión es muy superior si se utiliza la búsqueda avanzada o personalizada. Es importante destacar que el promedio total entre todos los alumnos con respecto a este indicador, es mayor al 95%. Es decir, más de 95 de cada 100 cursos devueltos en la búsqueda, son relevantes para el alumno de acuerdo a su perfil y preferencias. Este resultado es altamente satisfactorio y cumple con uno de los requerimientos principales de la propuesta, que consiste en que la búsqueda avanzada o mejorada devuelva únicamente los cursos relevantes para el usuario.

Alumno	Recall	
	Búsqueda estándar	Búsqueda mejorada
Alumno Test 1	1	1
Alumno Test 2	1	0,791666667
Alumno Test 3	1	1
Alumno Test 4	1	0,8125
Alumno Test 5	1	0,851190476
Alumno Test 6	1	0,958333333
Valor Promedio	1	0,902281746

Tabla 3: Valores de Recall

En la Tabla 3 se presentan los valores obtenidos para el indicador Recall. Aquí se observa en la Búsqueda avanzada valores menores a los observados en la Búsqueda estándar. Esto se debe a que al tratar de ajustar los resultados al perfil del usuario para que tengan la mayor precisión posible, algunos cursos pueden no ser devueltos, aún siendo relevantes. Pero los resultados son también positivos. Si bien el nivel de Recall es menor, sigue siendo muy aceptable (90,22%): aproximadamente de cada 100 cursos relevantes, en promedio 90 de ellos son devueltos en la búsqueda. Tomando como medida ambos indicadores, se puede concluir que la evaluación de ambos resulta muy satisfactoria y cumple con los objetivos deseados. Dado que el nivel de Precisión resultó muy superior, y que el nivel de Recall se mantuvo prácticamente igual descendiendo menos del 10%, se puede afirmar que la búsqueda devuelve los cursos que responden a la búsqueda del alumno, pero que también son de su preferencia.

Los cursos deben ser presentados en orden

descendente de acuerdo a las preferencias del alumno. Los cursos más significativos deben ser presentados en las primeras posiciones y los menos significativos en las últimas. Para evaluar el ordenado, se utilizan los mismos alumnos de la experimentación anterior, realizando búsquedas en modo Moodle estándar y en modo Moodle extendido, con 10 repeticiones por alumno. En el listado de cursos obtenidos con el modo Moodle estándar (Búsqueda estándar) se suprimen los cursos que no son relevantes y se ordena esta lista manualmente. Luego, se analizan los resultados de los ordenamientos obtenidos, comparando el ordenamiento manual con el obtenido con las extensiones realizadas (Búsqueda mejorada). Para esta comparación se propone el indicador *Calidad de ordenamiento* definido como el cociente entre la Cantidad de cursos correctamente ordenados sobre la Cantidad total de cursos presentados. Los resultados obtenidos se muestran en la Tabla 4. Se observa que aproximadamente el 95% de los cursos listados como resultado de la Búsqueda mejorada, se encuentran en el orden correcto, con los cursos más relevantes al principio y los menos significantes al final del listado.

Alumno	Ordenamiento	
	Búsqueda estándar	Búsqueda mejorada
Alumno Test 1	25,00%	95,00%
Alumno Test 2	73,00%	100,00%
Alumno Test 3	16,00%	87,00%
Alumno Test 4	44,00%	92,00%
Alumno Test 5	33,00%	95,00%
Alumno Test 6	42,00%	100,00%
Valor Promedio	38,83%	94,83%

Tabla 4: Calidad de Ordenamiento

Como se observa, la Búsqueda mejorada, en comparación con la Búsqueda estándar, presenta un ordenamiento muy superior. Esto es lógico teniendo en cuenta que el ordenamiento estándar utilizado es el alfabético y no se considera preferencias y características del alumno al momento del ordenamiento. Además, es importante observar que el ordenamiento de los cursos propuesto en este trabajo está muy relacionado, además del perfil del alumno y de las características de los cursos, a las

evaluaciones que el alumno hace de los mismos. Mientras más se utilice la plataforma, más evaluaciones son realizadas, y consecuentemente mejor y más exactos serán los resultados en cuanto a presentación y ordenamiento de los cursos devueltos.

Conclusiones

El tema de la personalización de los resultados de una búsqueda de información es de gran importancia en la actualidad, como consecuencia de que muchos usuarios de diferentes características acceden a las mismas herramientas. En este trabajo se propuso mejorar la búsqueda de cursos en la plataforma Moodle, estableciendo nuevas funcionalidades que permitan aprender de las preferencias de los usuarios de forma de poder presentarle a éstos sólo los cursos que les resulten relevantes de acuerdo a su perfil y ordenados por relevancia.

Para ello, fue necesario almacenar características de cursos, perfiles de usuarios y la preferencia de cada usuario por determinados cursos. Además se propone a los fines de mejorar la personalización, el uso de técnicas de aprendizaje automatizado, en particular el método de k-vecinos más cercanos. Para esto se decide trabajar tanto con descripciones de características de cursos como de perfiles de usuario que son compartidos por distintos cursos o alumnos respectivamente. En el trabajo se experimenta para obtener los parámetros óptimos para la ejecución del algoritmo de k-vecinos, obteniendo que 7 atributos son adecuados y los pesos respectivos para cada uno de ellos.

También se realizan pruebas empíricas con un conjunto de datos generados artificialmente para poder evaluar los resultados. En esta experimentación se obtiene que los resultados son altamente satisfactorios, lográndose el objetivo planteado y cumpliendo con los requerimientos necesarios. Es importante notar, que este tipo de técnicas de aprendizaje automatizado, trabaja mejor mientras más

datos tenga y más utilizada sea. Es por ello que la arquitectura propuesta, debería presentar resultados aún mejores, en condiciones de uso normal con datos reales. Se propone como trabajo futuro, utilizar la implementación de la arquitectura presentada en un ambiente con datos reales, a fin de evaluar, analizar y comparar la validez de esta hipótesis.

Agradecimientos

Este trabajo fue realizado dentro del marco del Proyecto Sistema de apoyo al docente en la búsqueda y preparación de material didáctico para la enseñanza en las escuelas santafesinas. Proyecto 219308 de la Secretaría de Estado de Ciencia, Tecnología e Innovación, Provincia de Santa Fe.

Bibliografía

- [1] Cole, J. & Foster, H. Using Moodle, Second edition, O'Reilly Media, Inc., 2008.
- [2] Bender, C., Motz, R., Deco, C., Saer, J. Recuperación personalizada de e-cursos. Proc. IX Congreso Iberoamericano de Informática Educativa, RIBIE 2008. Caracas, Venezuela, marzo 2008.
- [3] Motz, R., Guzmán, J., Deco, C. and Bender, C. Applying ontologies to educational resources retrieval driven by cultural aspects. JCS&T. Journal of Computer Science & Technology. Vol 5, N° 4, pp 279-284, 2005.
- [4] Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. Proc. 14th International Joint Conference on Artificial Intelligence. Vol 2, pp. 1137-1143, Morgan Kaufmann, 1995.
- [5] Mitchell, T. Machine Learning, McGraw-Hill, 1997.