

CALAM: Código Abierto de Learning Analytics para Moodle

Alavez, H., Bautista, T., De Mendizábal, M., González, J., Miranda, G., Moreno, G.
Coordinación de Universidad Abierta y Educación a Distancia y Facultad de Estudios Superiores Iztacala, Universidad Nacional Autónoma de México

Resumen

En el presente artículo se presentan dos herramientas para Moodle, una dirigida a los administradores, para cuantificar el tiempo que utilizan los profesores la plataforma y la otra es para que, a través de un tablero, los profesores realicen un monitoreo de todas las entregas de las actividades de los estudiantes. El proyecto CALAM (Código Abierto de Learning Analytics en MOODLE) busca formar una comunidad de profesores, administradores y programadores que, alrededor de las necesidades de mejora del proceso de enseñanza aprendizaje con *Moodle*, generen herramientas tecnológicas de uso cotidiano que les permitan enfocar sus esfuerzos en lo educativo y no en lo tecnológico.

Palabras clave

Análítica del aprendizaje, Tableros de control, *Moodle*, Evaluación, Retención, Intervención, Software libre

Antecedentes

La contribución de México al desarrollo y mejora del código de la plataforma Moodle ha sido marginal, 93 cambios de un total de 105,593, es decir, inferior al 0.09% (Contribuciones al código *Moodle*, 2017), aún cuando nuestro país es el cuarto lugar en el uso de la plataforma, sólo detrás de los Estados Unidos, España y Brasil (Estadísticas de Moodle, 2017). Quizá la contribución más significativa es la del Dr. Germán Valero que se dedicó a mantener la traducción de dicha plataforma al español de México (Traducen en la UNAM la plataforma educativa *Moodle 2*, 2012). Esta traducción no se considera parte del código central de Moodle y está clasificado como una *extensión*. Aún bajo esta perspectiva, no quiere decir que en nuestro país se utilice *Moodle* sin adaptarlo a las necesidades propias de la universidad y su práctica docente. Es común que las instituciones educativas y organizaciones usuarias de este software sea adaptado a las necesidades propias de la institución o del proyecto. En algunos casos, se agregan funciones especiales que aportan soluciones a problemas particulares (Villatoro, C., et. al., 2013). Sin embargo, ninguna de estas aportaciones se ve reflejada en el código fuente de *Moodle* y, cuando cambia la versión, se debe optar por volver a hacer esas modificaciones o rezagarse en la versión. En el caso particular del proyecto B@UNAM las valiosas aportaciones fueron, por ejemplo, una tabla de avance de actividades que permitía visualizar de forma inmediata el grado de cumplimiento de los estudiantes durante su curso.

El esfuerzo aquí presentado está encaminado a desarrollar una serie de extensiones para *Moodle* que, a través de las metodologías de trabajo del *software* libre, ofrezcan soluciones a pequeños problemas comunes y entusiasme a sus usuarios para colaborar con sus propias aportaciones. La forma de hacer estas contribuciones es a través de extensiones o plug-ins que sean fáciles de instalar por cualquier profesor o administrador de sistemas, sin necesidad de contar con mucha experiencia tecnológica ni saber programación. La ampliación de la funcionalidad de *Moodle* a través de extensiones es una técnica probada y eficaz para incluir mejoras a la funcionalidad de dicha plataforma, sin poner en riesgo la indispensable actualización tecnológica constante. Todas las aportaciones estarán en un repositorio de código libre muy popular llamado GitHub y que permite tomar el código con facilidad y enviar aportaciones cuando se desee.

En el Sistema Universidad Abierta y Educación a Distancia (SUAYED), de la Universidad Nacional Autónoma de México (UNAM), *Moodle* es el Sistema de Administración de Aprendizaje (mejor conocido por sus siglas en inglés LMS), que más se utiliza en las diferentes entidades académicas para ofrecer sus planes y programas de estudio. Incluso cabe señalar, que en una misma entidad académica y carrera, existen diferentes versiones de Moodle distribuidas en los diferentes planes de estudio. Además, hay que considerar que dada su particularidad de tener una arquitectura de código abierto, cada administrador programa adapta su versión de *Moodle* para atender las necesidades propias de la entidad académica.

De acuerdo con el *Primer censo de herramientas de gestión de cursos y repositorios digitales de aprendizaje* (González, 2016), en la Universidad Nacional Autónoma de México se utilizan diferentes “versiones de este software (...) La más utilizada en la actualidad es la 1.9 (33%), mientras que la más reciente corresponde a 2.7 o superior (12%)”.

Dada esta heterogeneidad de *Moodle*, resulta difícil pensar que los maestros que asesoran a través de esta plataforma, estén completamente capacitados en las múltiples versiones y configuraciones que tiene este Learning Management System (LMS). Además hay que considerar que un mismo maestro participa en diferentes entidades académicas, planes y programas de estudio.

Debido a esta heterogeneidad de versiones dentro de nuestra institución se tomó la decisión de desarrollar estas herramientas para la última versión disponible de Moodle, la 3.2 y no detenernos en adaptar las aplicaciones para cada una de las versiones existentes de la plataforma. Sin embargo es muy posible que este código funcione sin mayor problema cuando se utilicen versiones de Moodle mayores a la 2.6.

CALAM tiene el propósito de hacer una comunidad de maestros, administradores y programadores con un fin en particular, potenciar los datos que almacenan las bitácoras de *Moodle* y ofrecer para en una primera fase, cuadros de control (*dashboard*), que hagan más fáciles las tareas de monitoreo y seguimiento. Para lograrlo, se definen las siguientes responsabilidades de cada uno de los involucrados.

- Maestros: Son los encargados de definir las necesidades o las áreas de oportunidad a desarrollar dentro de la plataforma *Moodle*. Además, son los que mejor conocen las interacciones de los alumnos dentro del grupo, sus grados de participación y avance en actividades de aprendizaje, así como la gestión de los contenidos educativos.
- Administradores: Son los usuarios encargados de agregar o eliminar usuarios de la plataforma, matricular usuarios en los cursos, instalar y/o configurar nuevos módulos,
- Programadores: son los encargados de materializar en código las ideas de los maestros, administradores e investigadores que propongan ideas interesantes para ser implementadas.

Metodología de desarrollo de software libre

La idea central del desarrollo de software libre plasmada en el ensayo *The Cathedral and the Bazaar* (Raymond, 1997) en donde el autor hace una analogía entre la forma tradicional de desarrollar software, que compara con la construcción de una catedral, que tiene planeación centralizada, fuerte organización y un proceso con inicio y final y, por otra parte, el desarrollo de código abierto es similar a lo que ocurre en un bazar, en donde muchos intereses diversos convergen aleatoriamente y, en ocasiones, forman un proyecto común que tiene resultados específicos, pero difíciles de predecir.

De acuerdo con Raymond, en un proyecto de la complejidad del núcleo de Linux, funcionó muy bien la idea de liberar código con mucha frecuencia, aunque no estuviera del todo probado, para involucrar a más personas en el proyecto. Evidentemente esta opción implica que se debe poner mucha atención a la retroalimentación por parte de los usuarios para que las nuevas versiones de código incorporen sus sugerencias y se trabaje en mejorar el producto.

En el caso del CALAM se tomó la decisión de colocar el código fuente de las herramientas de analítica en un repositorio público GitHub, de tal forma que, cualquier persona que quiera utilizar dichas herramientas, las pueda descargar de forma inmediata mediante un sencillo procedimiento, pero además, si quiere contribuir a mejorar o ampliar el código, puede hacer creando un clon y sugiriendo los cambios directamente en el repositorio público. Si dichos cambios son aceptados, se incorporarán como parte del proyecto.

Cuando se coloca cualquier clase de código fuente en un repositorio público, es aconsejable utilizar una licencia de software libre. En el caso específico de CALAM, se consideró que la licencia más adecuada es la que utiliza *Moodle*, que es GPL versión 3 (Licencia de Moodle, 2017).

Se eligió utilizar la herramienta de control de versiones git por dos motivos en particular, tanto el núcleo de Linux, como el Moodle, utilizan este sistema de control de versiones. Un "sistema de control de versiones registra los cambios en un archivo o un conjunto de archivos de tal forma que se puedan recuperar versiones específicas posteriormente". (Chacon, 2014) En el caso particular de git, este control de versiones puede llevarse de forma distribuida, colocando un repositorio central a partir del cual los desarrolladores descargan el código fuente y, cuando puedan ofrecer una mejora, la incorporen a dicho repositorio, en donde una autoridad central decida si el cambio sugerido es pertinente, o no. Con CALAM se decidió utilizar el repositorio público GitHub que permite incorporar proyectos de cualquier tamaño bajo la condición de que sean software libre.

Mediante esta estrategia de compartir código funcional sin ninguna restricción para el usuario, se pretende entusiasmar a una creciente comunidad de usuarios que necesitan herramientas analíticas como las que se ofrecen en este esfuerzo inicial y captar posibles desarrolladores que

decidan sumarse a este proyecto. Evidentemente esto no será inmediato y requerirá de un esfuerzo importante de convocatoria y difusión para formar una comunidad inicial de usuarios que brinden comentarios sobre los productos ofrecidos y, eventualmente de desarrolladores de código que puedan introducir mejoras a las herramientas de analítica del aprendizaje que se vayan incorporando gradualmente.

Resultados

Las dos herramientas desarrolladas emplean técnicas de extracción de datos típicas de la analítica del aprendizaje y por ello el nombre que se le dio al proyecto “Código Abierto de Learning Analytics para Moodle, CALAM”.

Estas dos herramientas se desarrollaron bajo la modalidad de una extensión (o plug-in) de *Moodle*, de tal forma que el profesor o administrador que las instale, las encontrará en un bloque de *Moodle*, como se puede observar en la figura 1.



Figura 1. Forma de presentar las herramientas en la plataforma *Moodle*

La primera herramienta, “Tiempos en plataforma” sirve para medir el tiempo que un profesor emplea en la plataforma tecnológica. En los sistemas administrativos convencionales universitarios se supone que un profesor debe estar cierto tiempo frente a grupo. En el modelo a distancia esta premisa es ineficaz, puesto que el profesor nunca está con su grupo de forma presencial. Es por ello que una forma de verificar que el profesor atienda a sus alumnos, es medir su actividad laboral dentro de la plataforma. En algunas instituciones, se le exige al profesor que dedique su tiempo frente a grupo a contestar preguntas en un chat o, en casos extremos, que establezca una clase virtual completa utilizando algún sistema de videoconferencia. La propuesta aquí presentada propone una solución intermedia: verificar que el profesor ingresa a la plataforma e invierte tiempo en ella en actividades típicas como contestar preguntas en un foro, calificar actividades, establecer sesiones de chat con sus estudiantes y, en general, todo lo que requiera emplear tiempo en

el LMS. El resultado de los cálculos de esta herramienta es que se genera una gráfica interactiva como la siguiente:



Figura 2. Tiempo de uso de un usuario en la plataforma

Esta gráfica es interactiva y permite conocer el valor de cada punto al colocar el apuntador sobre ella. La línea en color amarillo representa el promedio de los usuarios consultados. En este caso el profesor fue muy dedicado durante ese periodo de tiempo.

Esta herramienta emplea a una de las tablas más importantes de Moodle que es la `mdl_logstore_standard_log`, en donde se recolectan todas las acciones que hace un usuario que ingresa en la plataforma. A partir de ahí se utilizan tres datos fundamentales, el periodo de tiempo que se quiere analizar, la “ventana” de tiempo, que es el intervalo entre dos acciones del usuario y que se ha establecido empíricamente con el valor de 1,800 segundos, y el rol o los roles que se quieren procesar. A partir del rol o los roles, se obtiene la lista de los usuarios que cumplen con el criterio, se verifica para cada usuario que existan acciones en la bitácora y se recorre todo el periodo de análisis, para contabilizar el tiempo que el usuario empleó en la plataforma. El resultado de dicho cálculo se guarda en una tabla temporal a partir de la cual se genera la gráfica interactiva. Además esta información puede ser exportada a una hoja de cálculo mediante la generación de un archivo CSV Comma Separated Values, o Valores Separados por Comas.

La segunda herramienta “Actividades entregadas” es aún más sencilla de entender porque permite visualizar, de un solo golpe, todas las actividades entregadas de sus estudiantes, de tal forma que el profesor sepa de inmediato cuáles son los estudiantes que no han entregado nada o pocas actividades, también le permite saber quiénes son los que van bien

o no en el curso y, sobre todo si se hace durante varios momentos del curso, le permite intervenir oportunamente para apoyar al estudiante en problemas.

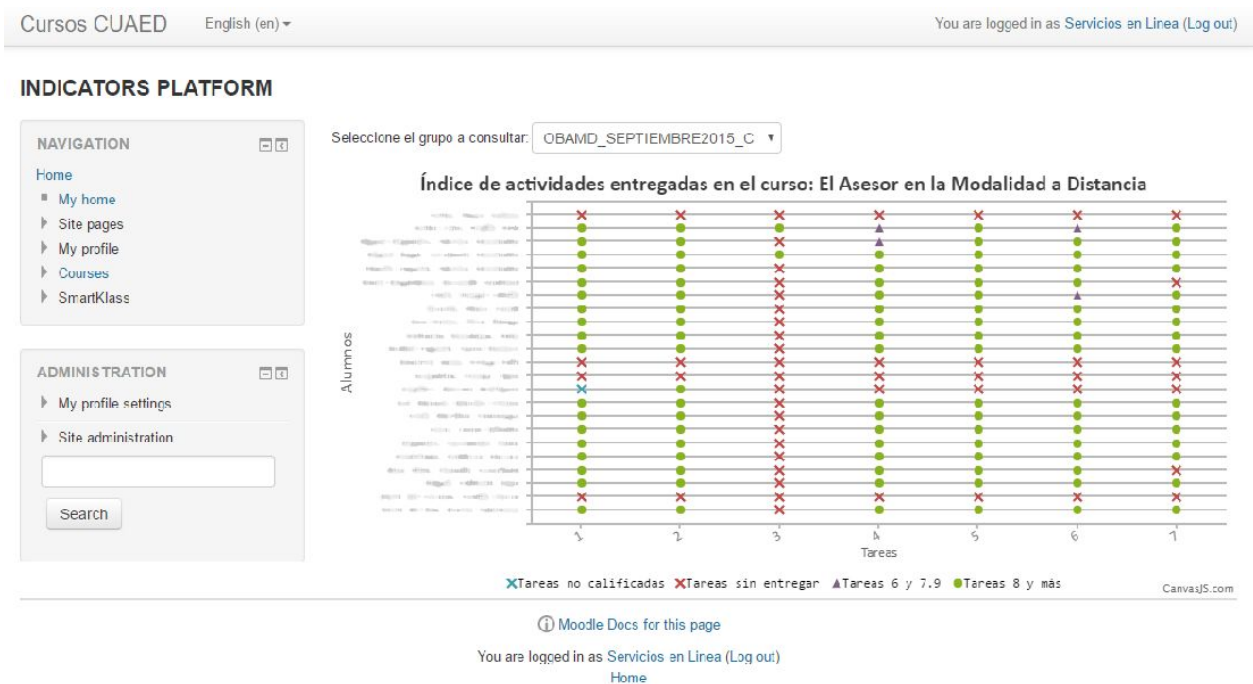


Figura 3. Tabla de actividades de un grupo de estudiantes

Desarrollos futuros

Se tienen identificadas varias áreas de oportunidad para crear más código de analítica del aprendizaje como, por ejemplo, visualizar la interacción entre los estudiantes y los profesores, que ya se tiene muy avanzada pero aún no se ha liberado el código. Otra área de oportunidad es desarrollar una herramienta que, a partir de los cuestionarios de opción múltiple, se generen estadísticas sobre las preguntas más frecuentemente mal contestadas y también las más frecuentemente bien contestadas para poder brindarle al profesor un análisis sobre cuáles son las preguntas que se deben modificar, ya sea por imposibles o por triviales. Finalmente de acuerdo con las propuestas del artículo de YeonJeong Park, se puede explorar la posibilidad de incorporar las siguientes herramientas: regularidad en el ingreso a la plataforma, visitas a la plataforma, tiempo empleado en la plataforma, visitas al repositorio, frecuencia total de ingreso. (YeonJeong, 2015).

Conclusiones

Esta es únicamente la etapa inicial de un proyecto de largo plazo de desarrollo de herramientas de analítica para el aprendizaje. Una de las premisas del desarrollo de software libre es que es buena idea liberar el código lo más pronto posible para formar una comunidad de usuarios a su alrededor. A partir de ahí poco a poco se incorporarán cada vez más voluntarios entusiasmados para cumplir un objetivo común: mejorar la calidad del aprendizaje a distancia.

Bibliografía

Contribuciones al código *Moodle*. <https://moodle.org/dev/contributions.php?version&page=4>
Consultado el 26 de abril de 2017.

Estadísticas de Moodle. <https://moodle.net/stats/> Consultado el 26 de abril de 2017.

Licencia de Moodle <https://docs.moodle.org/dev/License> Consultado el 30 de abril de 2017.

Traducen en la UNAM la plataforma educativa Moodle 2.
http://www.dgcs.unam.mx/boletin/bdboletin/2012_542.html 3 de septiembre de 2012.
Consultado el 26 de abril de 2017.

CHACON, Scott, STRAUBB, Ben, *Pro Git*, 2nd ed. Apress. 2014. Disponible en línea en
<https://git-scm.com/book/en/v2>

GONZÁLEZ, MariCarmen et al. *La UNAM digital: Primer censo de herramientas de gestión de cursos y repositorios digitales de aprendizaje*. México: UNAM, 2016.

RAYMOND, E.S. (1999). *The Cathedral & the Bazaar*. O'Reilly Recuperado de
<http://www.catb.org/~esr/writings/cathedral-bazaar/>

VILLATORO, Carmen, Aznavwrian, Lourdes, Vadillo, Guadalupe. *B@UNAM a través de sus números*. Revista mexicana de bachillerato a distancia. Número 10. Año 5. Agosto de 2013.
También se puede encontrar en versión digital en
<http://bdistancia.ecoesad.org.mx/?articulo=bunam-a-traves-de-sus-numeros>

YEONJEONG, Park. IL-HYUN, Jo. Development of the Learning Analytics Dashboard to Support Students' Learning Performance. *Journal of Universal Computer Science*, vol. 21, no.1. pp. 110-133

