

Extendiendo Google Course Builder mediante Proyectos Realistas en un Curso de Master

Carlos Delgado Kloos, *Senior Member, IEEE*,
Pedro J. Muñoz-Merino, *Member, IEEE*, Mario Muñoz Organero, *Member, IEEE*

Title—Extending Google Course Builder with Real-World Projects in a Masters' Course

Abstract—The contribution of this paper is twofold. On the one hand, we present an experience of a Masters' level course based on real-world projects with interaction with real developers from industry. This reality check created a special motivation for students as well as provided a useful experience for their future work. On the other hand, we describe the technical details of the three projects carried out by the students. These were extensions to Google's MOOC platform Course Builder. They were about an improvement in the management of quizzes, an extension with learning analytics tools, and an integration with Mozilla Open Badges.

Index Terms—assessment, badges, learning analytics, learning by doing, MOOCs, real-world projects

I. INTRODUCCIÓN

SALMAN Khan, el creador de la Khan Academy que tanto impacto ha tenido en la enseñanza online, reflexiona en [1] sobre la desconexión entre las expectativas de los estudiantes, la experiencia del aula tradicional y la necesidad creciente de creadores activos en el mercado de trabajo. Comenta que los estudiantes aprenden mucho más trabajando en problemas reales que en proyectos definidos artificialmente. El reto intelectual de estar trabajando en un problema real produce un aprendizaje mejor en el momento y más útil para el trabajo futuro. También habla de la importancia de realizar prácticas en empresas como Google o Apple.

Dentro de la asignatura de *Plataformas para Comunidades en Red* del programa de *Master en Ingeniería Telemática* del Departamento de Ingeniería Telemática de la Universidad Carlos III de Madrid, hemos optado por un enfoque eminente-mente práctico en la línea que propone Salman Khan.

En las siguientes secciones explicamos en primer lugar la dinámica de la asignatura y a continuación el tema concreto tratado. Se describen después los tres proyectos realizados. La aportación de este artículo se realiza en dos niveles, por una parte en relación a la metodología y enfoque realizado para el desarrollo de la asignatura y por otra parte por los resultados obtenidos por los equipos en sus proyectos.

Los autores son profesores del Dep. Ingeniería Telemática, Universidad Carlos III de Madrid, Av. Universidad, 30, 28911 Leganés (Madrid), España. E-mail: cdk@it.uc3m.es, pedmume@it.uc3m.es, munozm@it.uc3m.es

II. LA DINÁMICA DE LA ASIGNATURA

Los contenidos concretos de la asignatura han variado en las distintas ediciones, no así los objetivos globales y la dinámica. Los objetivos globales de la asignatura incluyen desarrollar la creatividad, obtener habilidades para la implementación de aplicaciones software, aprender a colaborar en un equipo de trabajo (liderazgo, negociación, resolución de conflictos, etc.), entendimiento de documentación técnica avanzada, realización de presentaciones y propuesta de una propuesta de negocio. Así pues, la asignatura no solo tiene como objetivo aprender conocimientos tradicionales, sino también una serie de habilidades transversales que son muy útiles de cara al mercado laboral.

El hecho de tratarse de una asignatura de Máster posibilita e invita a utilizar un formato alternativo, no centrado en la instrucción tradicional, sino en conseguir que el estudiante aprenda por la práctica. Es por ello que siempre se ha enfocado la asignatura por medio de proyectos realizados en grupos por parte de los estudiantes. La temática de los proyectos ha ido variando a lo largo de los años:

- 2010: Aplicaciones colaborativas con *Google Wave*
- 2011: Aplicaciones de realidad aumentada con *Junaio*
- 2012: Aplicaciones de gamificación con *OpenFeint*
- 2013: Extensiones de *Google Course Builder* (GCB)

La plataforma GCB utilizada en 2013 es un sistema software educativo de código abierto que permite la realización de cursos masivos abiertos on-line.

Los alumnos contaban con la ayuda de los profesores de la asignatura (los autores de este artículo y en el pasado también Abelardo Pardo, ahora en la Universidad de Sídney), pero además de tutores externos que aportaban la visión de la empresa. Las clases se utilizaban en parte para explicar la tecnología necesaria para desarrollar el proyecto, pero sin el objetivo de ser exhaustivos, pues los alumnos eran capaces de estudiarse la documentación igual que lo podía hacer el profesor. Más énfasis se hacía en otras competencias menos técnicas, pero no menos importantes en especial para el trabajo futuro: cómo ser creativos y hacer *brainstorming*, cómo realizar trabajo en grupo de forma colaborativa, como hacer presentaciones impactantes, etc.

En el año 2013, contamos con la colaboración del equipo de desarrollo de *Google Course Builder*, con los que se realizaban videoconferencias cada 15 días, además de realizarse algunas interacciones por correo electrónico. Las presentaciones finales contaron con la asistencia presencial

o remota de todos los profesores y tutores externos, así como de otros invitados. Además de estas presentaciones los alumnos tuvieron que hacer un vídeo en el que vendieran sus desarrollos a personas que pudieran estar interesadas. Finalmente, de los dos primeros proyectos surgieron artículos que fueron presentados en un congreso [2][3], con lo que el trabajo realizado recibió un reconocimiento adicional y sirvió para que los estudiantes avanzasen en el dominio de la escritura de documentación técnica.

La asignatura tuvo un total de 25 sesiones de 90 minutos. La planificación de la misma fue la siguiente con este orden:

- Presentación. 1 sesión.
- Tecnologías necesarias. 6 sesiones. En estas sesiones, los profesores presentaron una introducción de las tecnologías clave para realizar proyectos en GCB: HTML, JavaScript, Python, Google App. Engine y el propio GCB. Los profesores dieron una explicación de aspectos relevantes de estas tecnologías pero sin ser exhaustivos, y los alumnos tuvieron que realizar una serie de ejercicios en clase y en casa relacionadas con ellas. Esta introducción a las tecnologías sirve para que los alumnos tengan un conocimiento inicial, sepan buscar más conocimientos relacionados con ellas, etc. pero según el tipo de proyecto a desarrollar, los alumnos debieron profundizar más en cada una de ellas.
- Sesión invitada por el equipo de GCB. 1 sesión. En ella, el propio equipo de GCB dio una introducción sobre la plataforma.
- Creatividad, colaboración y planificación. 4 sesiones. En estas sesiones los alumnos aprendieron técnicas de creatividad como el *brainstorming*, metodologías de colaboración y de planificación, así como herramientas asociadas. Los profesores consensuaron con el equipo de GCB una serie de temas entre los que los alumnos podían elegir su proyecto. Los alumnos conformaron los 3 grupos de trabajo y eligieron 3 de los temas presentados. No obstante, dentro del tema presentado hay mucha libertad para la elección concreta de las características a implementar, por lo que los alumnos con la ayuda de los profesores tuvieron que ser creativos para determinar las funcionalidades en cada uno de los 3 proyectos. Como resultado de estas sesiones, cada grupo realizó una especificación de su proyecto.
- Desarrollo. 11 sesiones. En estas sesiones los alumnos realizaron la implementación de las funcionalidades del módulo. El profesor actúa en esta fase como un mentor, acompañando a los alumnos, para orientarles y ayudarles en aspectos como la búsqueda de información, toma de decisiones de la arquitectura, programación del código, resolución de conflictos del grupo, etc. Todos los profesores guían y orientan a cada grupo de alumnos, pero cada uno de los profesores es asignado a un grupo donde su implicación es mayor. En ningún caso estas clases son teóricas sino que el profesor se sienta con el grupo para ayudar en los diferentes aspectos. Por otro lado, una persona del equipo de GCB es asignada a cada grupo y le ayuda en el desarrollo en 4 de estas sesiones, realizando videoconferencias. Durante el desarrollo,

los profesores van marcando retos y metas intermedias a conseguir durante el proyecto, que suelen coincidir con las videoconferencias con el equipo de GCB.

- Cómo hacer presentaciones. 1 sesión.
- Presentación final. 1 sesión. Los alumnos presentan su trabajo final ante los miembros del equipo de GCB, profesores, y otros alumnos, investigadores y profesores.

La experiencia realizada ha sido muy positiva como se puede concluir de los comentarios y opiniones de los alumnos, a los que motiva el trabajar en proyectos reales, con la ayuda de una empresa como Google, y aprendiendo habilidades muy importantes para su desempeño profesional. Varios alumnos han expresado su preferencia por este esquema de trabajo respecto a las clases tradicionales.

Finalmente, los alumnos tuvieron que establecer de manera anónima un orden de los 3 grupos de trabajos tras la exposición, según les había gustado más o menos. Así mismo, también tuvieron que establecer un orden de cómo habían trabajado los propios miembros de su equipo de mejor a peor. Con esta evaluación propia de los alumnos, junto con la evaluación de los profesores y los comentarios de los miembros del equipo de GCB, se calculó la nota final de cada uno de los alumnos.

III. LA TEMÁTICA: PLATAFORMAS DE MOOCS

La temática de la edición del curso 2012/13 era el estudio de las plataformas de MOOCs (*Massive Open Online Courses*, Cursos en Línea Masivos Abiertos) como plataformas para comunidades de aprendizaje en red.

Los MOOCs han surgido de forma más o menos disruptiva en los últimos dos años, sobre todo de la mano de la educación universitaria, agrupando tecnologías que ya se venían usando en la educación on-line en un nuevo marco que parte de la idea de automatizar procesos y tareas, tanto de exposición de contenidos, como de colaboración en el curso así como de evaluación al alumno para poder ofrecer los cursos de forma masiva a coste reducido. La idea de fondo es hacer llegar el contenido de los cursos tradicionalmente enseñados a puerta cerrada dentro de las instituciones universitarias a todo aquel que lo necesite a través de Internet. Podríamos decir que al igual que el comercio electrónico de finales del milenio pasado permitía romper las barreras espaciales a la hora de vender productos de forma que las empresas ya no estaban limitadas a captar compradores locales, sino que podían vender en cualquier parte del mundo (con una logística de soporte adecuada), ahora la universidad no está obligada a limitarse a competir por los alumnos locales, sino que puede atraer a cualquier alumno en cualquier parte del mundo (igualmente con una logística de soporte si se requiere por ejemplo tener garantías en el proceso de evaluación como iremos viendo). Esto hace que las universidades tengan que competir en un mercado global de más calidad, y que las universidades de prestigio puedan quitar alumnos a las universidades que tradicionalmente han vivido del mercado de alumnos local.

Tras el lanzamiento de las primeras iniciativas en Estados Unidos (Udacity [1], Coursera [5], edX [6], NovoEd [7], etc.), otras iniciativas han ido apareciendo para el despliegue

de MOOCs en otras regiones del mundo como MiríadaX [8] en España, iversity [9] en Alemania, FutureLearn [10] en Reino Unido, Open2Study [11] en Australia, FUN [12] en Francia, XuetangX [13] en China o OpenUpEd [14] en la Unión Europea. Muchas universidades están poniendo en marcha su propio entorno para el despliegue de estos cursos en línea masivos y abiertos o bien usando alguna de las plataformas ya instaladas en el mercado.

IV. LOS PROYECTOS DESARROLLADOS EN GCB

La plataforma seleccionada para la asignatura fue *Google Course Builder* por tener acceso tanto al código fuente como a los propios desarrolladores de la misma que nos proporcionaron tanto soporte tecnológico como visión de la plataforma.

GCB es una plataforma basada sobre el API de Python del Google Application Engine que ofrece herramientas tanto para el despliegue de contenidos como de actividades de evaluación sumativa y formativa. El proyecto surgió al liberalizar la tecnología que el propio Google utilizó para el lanzamiento de su primer MOOC (“Power Searching with Google”) de cara a posicionarse como un referente en Internet como tecnología abierta para la generación de MOOCs y a su vez como herramienta basada en la infraestructura de los servidores de Google. Finalmente, a 10 de septiembre de 2013, el proyecto de Google se ha fusionado con la plataforma edX de cara a poner al servicio de una de las instituciones más prestigiosas en la enseñanza de la ingeniería en el mundo, la fuerza de desarrollo del equipo de Google y unir esfuerzos para crear un referente en el mundo de los MOOCs.

Las proyectos desarrollados por los alumnos fueron:

- una herramienta para la mejora de gestión de ejercicios,
- una herramienta para ampliar las capacidades de analítica sobre los datos de uso de la plataforma y
- una herramienta para integrar GCB con el sistema de insignias de Mozilla (Mozilla OpenBadges [14]).

Estas tres nuevas herramientas fueron consensuadas con el equipo de Google como potenciales mejoras que se alineaban con el *roadmap* de desarrollo por parte de Google. Se describen a continuación los proyectos desarrollados.

V. EXTENSIÓN DE GCB PARA EJERCICIOS EN FORMATO GIFT

En esta sección describimos el proyecto realizado por uno de los equipos de alumnos que consistió en la selección de un determinado formato para la descripción de ejercicios y la implementación de una herramienta de autoría para ese formato en GCB.

GCB permite desde la primera versión la definición de ejercicios que se pueden corregir de manera automática. La definición de estos ejercicios sin embargo no es cómoda. El profesor (o el personal de apoyo) los tiene que programar en JavaScript. Esto puede no ser un problema si la persona conoce mínimamente este lenguaje, pero no es asumible en caso contrario. A pesar de que el formato no es muy complicado y sólo se necesita conocer un subconjunto, es

conveniente simplificar al máximo la tarea de autoría. Además, típicamente habrá todo un ciclo de vida asociado a la definición y despliegue de ejercicios. Después de la definición de ejercicios, viene la necesidad de reutilizarlos, de mantenerlos, de importarlos y exportarlos a otras instancias de la misma plataforma o de distintas plataformas. Hay una gran cantidad de formatos que han sido utilizados por entornos de aprendizaje entre los que no se encuentra JavaScript tal como se utiliza en GCB. .

El formato elegido debería permitir la escritura de ejercicios directamente sin ningún editor especial. También debería ser suficientemente expresivo como para representar la mayoría de los tipos de ejercicios previstos en GCB y ser aceptado por alguna otra plataforma de aprendizaje de uso en instituciones educativas.

Un ejemplo de ejercicio expresado en la notación esperada por GCB es el siguiente:

```
{questionType: 'multiple choice',
questionHTML: '¿Cuánto es 1+1?',
choices:
[
['1', false, 'Falso, la respuesta es 2'],
['2', true, '¡Correcto!'],
['3', false, 'Falso, repasa la lección'],
]
}
```

En la literatura se pueden encontrar una multitud de formatos para ejercicios [16]. Algunos de ellos son IMS QTI [17], Moodle XML [18], Hot Potatoes [19], OpenMark [20], DocBook [21], y el formato de Blackboard [22]. Muchos de ellos son formatos basados en XML y se utilizan típicamente por medio de editores específicos. Algunos son formatos propietarios no textuales, que también precisan de editores.

Existe una clase de formatos que tiene una serie de propiedades interesantes. Nos referimos a formatos como Markdown [23] que se ha utilizado en wikis. El texto se puede introducir en las wikis sin que los usuarios tengan que tener conocimientos de HTML y sin que tengan que renunciar a unas características mínimas que proporciona el formato. Markdown es un lenguaje de marcado ligero que parece texto, pero que contiene algunas sencillas marcas de estilo que pueden ser traducidas fácilmente a HTML. Por ejemplo, el siguiente texto escrito en Markdown

```
Heading
=====
* an item in a bulleted list
* another item
```

puede ser traducido a HTML de la siguiente forma

```
<h1>Heading</h1>
<ul>
<li>an item in a <b>bulleted</b> list</li>
<li>another item</li>
</ul>
```

Siguiendo la filosofía de Markdown, GIFT es un formato de marcado ligero para la descripción de ejercicios [24], [25], [26]. La pregunta se escribe como texto normal y la o las respuestas (correctas o incorrectas) entre llaves. A continuación se presenta un ejemplo en formato GIFT:

```
1 + 1 es igual a {=2 =two}
1 + 1 es igual a {=2 ~3}
```

En la primera línea hay una pregunta con dos posibles respuestas correctas (pregunta tipo relleno de blanco) y en la segunda una pregunta con dos posibles respuestas, en donde una es correcta y una incorrecta (pregunta tipo opción múltiple). La figura 1 muestra su presentación. Con esta sencilla notación se puede representar un conjunto amplio de tipos de ejercicios, como verdadero/falso, de respuesta numérica, opción múltiple (con una o varias respuestas correctas), emparejado, relleno de blanco, respuesta corta, redacción, etc.

El ejemplo indicado arriba en notación GCB se representaría en GIFT de la siguiente forma (# separa el texto de realimentación para el estudiante):

```
¿Cuánto es 1+1?
{
  ~1 #Falso, la respuesta es 2
  =2 #¡Correcto!
  ~3 #Falso, repasa la lección
}
```

Las razones para la selección de este formato fueron varias:

- Es fácil de entender y escribir, por lo que no es necesario utilizar un editor especializado.
- Es un lenguaje sencillo pero potente que contempla la mayoría de los tipos de ejercicios que se necesitan.
- Fue creado por la comunidad de Moodle y aceptado por esta plataforma, por lo que es compatible con esta plataforma.
- Existen herramientas de conversión desde Word o Excel con macros definidas a GIFT. Esto facilita la adopción.
- Encaja muy bien con Markdown con lo que se puede dar formato a las preguntas.

Una vez seleccionado GIFT como formato, los alumnos tuvieron que diseñar e implementar la extensión para GCB. Se diseñaron varios módulos:

- Un editor que permitiese escribir ejercicios en formato GIFT a través del *dashboard* proporcionado por GCB (GIFT en la figura)
- Un editor gráfico que permitiese la edición guiada de ejercicios en GIFT con opciones (GUI en la figura)
- Un módulo que tradujese el formato GIFT al formato JavaScript de GCB
- Módulos de importación y exportación en formato GIFT

Desde el punto de vista técnico se implementaron los siguientes módulos (véase Fig. 3.):

- Un parser del formato GIFT al formato interno de JavaScript de GCB (*geditorParser.py*)
- Un visualizador de ejercicios GIFT (*geditorRenderex.py*). También permite la edición de ejercicios.
- Un comprobador de las respuestas dadas por los alumnos (*geditorChecker.py*). Esta comprobación se realiza en el lado del servidor. Esto proporciona un nivel de seguridad extra a lo implementado en GCB, pues un usuario con conocimientos técnicos podría ver

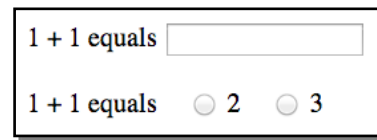


Fig. 1. Presentación de los dos ejemplos en formato GIFT

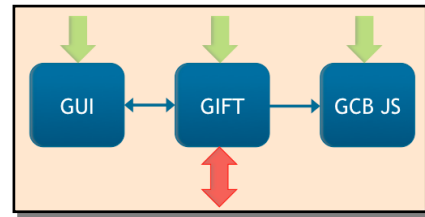


Fig. 2. Arquitectura con módulos principales

las respuestas correctas, ya que la comprobación se hace en el lado del cliente.

- Un módulo principal (*geditorMain.py*) que llama a los anteriores y enlaza con el código original de GCB.

A continuación se muestran algunas figuras de relevancia de la herramienta. En la Fig. 4 se muestra la opción adicional de edición de ejercicios en formato GIFT.

En las figuras 5 y 6 se muestra el editor gráfico para dos tipos de ejercicios (de tipo numérico y de emparejamiento) con sus respectivas opciones. En la figura 7 se muestra la visualización para un estudiante para el ejercicio creado en la figura 6.

El desarrollo realizado permite algunas funcionalidades adicionales, que no estaban contempladas en la versión estándar de GCB:

- La caja del ejercicio para completar con una palabra se puede poner en medio de una frase, no solo al final.
- En el caso de respuestas abiertas, puede haber más de una respuesta correcta.
- En el caso de ejercicios numéricos se puede especificar rangos de valores correctos
- Se pueden indicar textos de ayuda específicos para cada respuesta y no solamente de forma genérica por ejercicio.
- Se pueden definir pesos para los ejercicios a la hora del cálculo de la nota final
- Las preguntas se pueden expresar utilizando HTML, Markdown y otros formatos

La extensión proporcionada no sólo resuelve el problema de la usabilidad a la hora de definir ejercicios. También mejora la interoperabilidad y proporciona una funcionalidad superior.

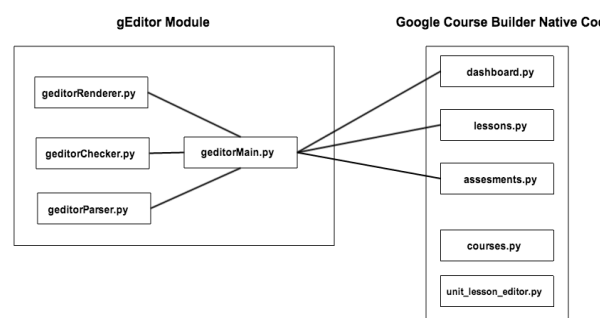


Fig. 3. Módulos principales desarrollados

VI. EXTENSIÓN DE GCB PARA LEARNING ANALYTICS

El área de *learning analytics* es habitualmente concebida en un sentido amplio como aquella encargada de la recolección, almacenamiento y análisis de datos con el fin de tener un mayor conocimiento del aprendizaje y poder mejorarlo. En entornos MOOC de aprendizaje como es GCB, las técnicas de *learning analytics* tienen gran importancia porque al tener que tratar con gran número de usuarios, son necesarios procedimientos que den automáticamente información útil a los profesores y a los alumnos ya que al ser tan elevado el número es difícil realizar un seguimiento personalizado.

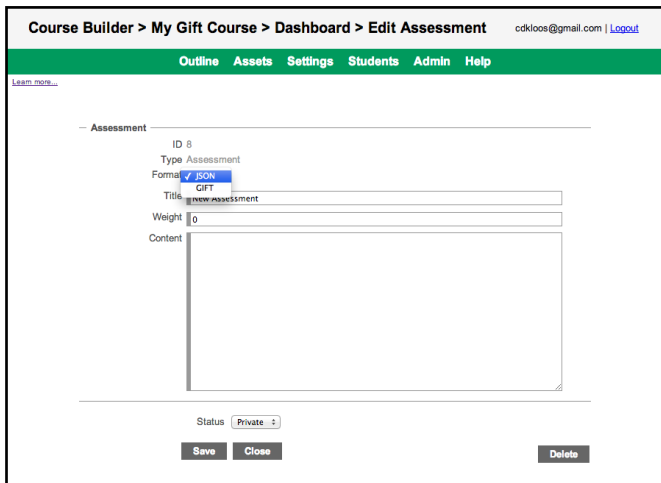


Fig. 4. Editor textual GIFT con dos posibles formatos: JavaScript y GIFT

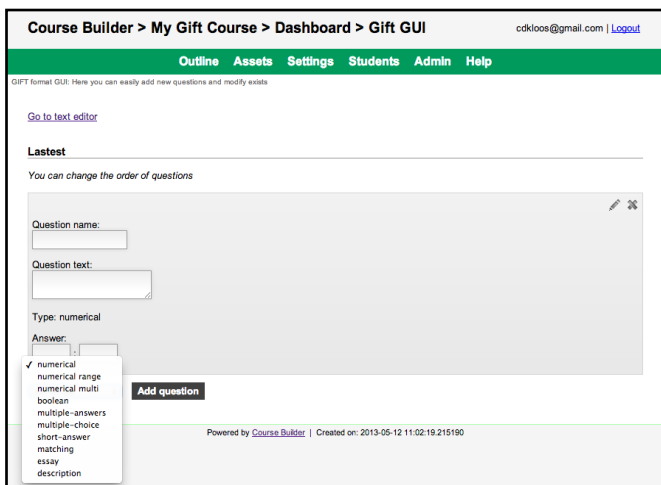


Fig. 5. Editor gráfico GIFT con ejercicio de tipo numérico

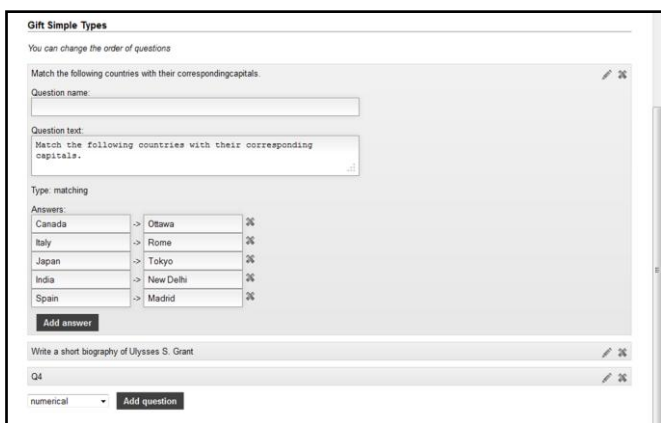


Fig. 6. Editor gráfico GIFT con ejercicio de tipo emparejamiento

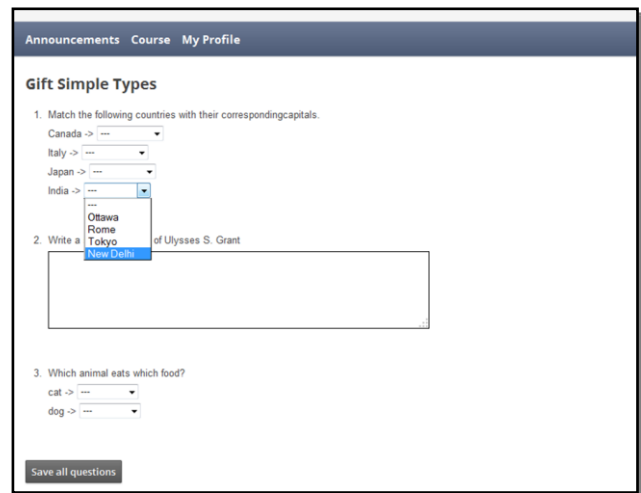


Fig. 7 Visualización para el estudiante de ejercicio de fig. 6

El proceso de *learning analytics* se puede concebir como compuesto de un conjunto de fases adaptadas de [27].

- En primer lugar está la captura y almacenamiento de datos que suelen venir dados en forma de eventos. Por ejemplo un evento puede ser “El usuario X editó el recurso Y a las S horas en la plataforma P”. Existen especificaciones y metodologías para realizar ese almacenamiento de eventos como CAM (*Contextualized Attention Metadata*) [28] que puede tener mapeo a XML pero también a otros formatos como RDF para funcionar en la Web semántica [29].
- Seguidamente está el procesado y combinación de datos para inferir información inteligente a partir de datos de bajo nivel. Usualmente los datos de bajo nivel son difícilmente interpretables por los actores del aprendizaje y se necesitan métodos para realizar ese procesado y combinación para obtener métricas de más alto nivel. En esta línea se encuentra por ejemplo el trabajo en [30] donde se proporcionaron parámetros como la efectividad, eficiencia o hábitos en la realización de ejercicios para la plataforma Khan Academy.
- Finalmente, en la última fase está la visualización y recomendación. Por un lado, se trata de presentar la información de una manera que los usuarios puedan comprender de forma sencilla, y por otra de ir más allá e interpretar el contexto educativo concreto para poder realizar recomendaciones precisas. Estas recomendaciones serán realizadas a los alumnos principalmente, aunque también a los profesores (por ejemplo qué materiales podría ser recomendable su revisión o cambio).

Cada una de las citadas fases tiene sus retos en general, que se pueden particularizar para la plataforma GCB. En concreto, a continuación se exponen aquellos retos que se abordaron en el proyecto, agrupados en cada una de las fases comentadas.

En la recolección de datos, es necesario tener todos los eventos necesarios para poder inferir los parámetros seleccionados. En el caso de GCB muchos de los datos de los eventos de usuarios (por ejemplo de los *assessments*) están almacenados en el *DataStore* del *AppEngine*. Pero hay eventos de herramientas externas (como de *YouTube*

para visualizar los vídeos) que son utilizadas en GCB para los que no está esa información en la base de datos.

Entre los retos del procesado de información para obtener información inteligente a partir de datos de bajo nivel se encuentra qué información seleccionar, qué datos tomar y cómo combinarlos para poder inferir la información que se está buscando. Cuando se desarrolló el proyecto, GCB daba sólo información estadística general para los *assessments* tal como el número de alumnos matriculados, los que completaron cada *assessment* o la media de un *assessment* concreto. Así mismo, no se daba información relativa a los vídeos como el número de accesos, cuáles se empezaban pero no se visualizaban o qué trozos de vídeos eran ampliamente visualizados y cuáles no.

Finalmente, en el área de visualizaciones es necesario dar gráficos precisos sobre lo que está sucediendo. Y en el ámbito de recomendaciones es necesario dar notificaciones para que se pueda mejorar y corregir el aprendizaje. Durante la ejecución del proyecto, GCB no tenía visualizaciones ni recomendaciones relacionadas ni con vídeos ni con los *assessments*.

Para dar respuesta a algunos de estos retos en GCB en las diferentes fases de *learning analytics*, el equipo de alumnos que desarrollaron el proyecto tuvieron que añadir las siguientes funcionalidades:

- Recolección de eventos de *YouTube*. Los eventos que se capturaron fueron los de inicio, finalización, pausar, rebobinar, avance. El recolector de eventos reside en el cliente en el navegador y utiliza la tecnología JavaScript. También se usa el API de *YouTube*. Este API no tiene los eventos de rebobinar y avance, sino que tiene los de inicio, finalización, pausar y play. Por ello, en el trabajo realizado se calcula esos dos nuevos eventos viendo los otros. Cuando sucede uno de los eventos, se lanza una petición al servidor mediante la tecnología AJAX. En el servidor se procede a guardar la información de cada uno de los eventos, utilizando el DataStore, en un nuevo tipo de entidad creada denominada *YoutubeEvent()* que almacena por cada evento la fecha, el usuario, el vídeo accedido, el tipo de evento, o el tiempo en que sucedió dicho evento. Gran parte de este desarrollo es reutilizable en otras plataformas que utilicen *YouTube* para ubicar sus vídeos. La parte de cliente sería totalmente reutilizable, y la parte del servidor se podría adaptar para por ejemplo guardar los datos en otra base de datos o utilizar otra interfaz de visualización.
- Visualizaciones relacionadas con los eventos de *YouTube*. Entre las visualizaciones disponibles se encuentra una representación de todos los vídeos, donde se indica el número total de alumnos que empezaron cada vídeo y el número que finalizaron dicho vídeo (figura 8). De esta manera, se puede conocer si cada uno de los vídeos ha sido suficientemente accedido, y aquellos vídeos donde por ejemplo muchos alumnos lo empezaron pero no lo terminaron. Por otro lado, se desarrolló un gráfico por cada vídeo, donde se representa el tiempo de duración del vídeo en el eje X respecto a los diferentes identificadores de alumnos en el eje Y, de forma que cada línea representa un alumno y se ven sus eventos

de pausar, rebobinar y avanzar. Cada uno de estos se representa con un color diferente. Además, otros gráficos muestran para cada vídeo, dividido en intervalos de tiempo, el número de eventos de cada tipo en cada uno de dichos intervalos. Estos gráficos permiten por ejemplo ver a los profesores qué partes del vídeo han sido vistas por pocos alumnos (puede ser indicadores de que esas partes no son interesantes y necesitan ser revisadas) o han sido vistas mucho y de manera repetida por los alumnos (puede ser indicador que se necesita expandir esas partes de explicación porque los alumnos necesitan verlas muchas veces para retenerlas).

- Visualizaciones relacionadas con los tests de autoevaluación. Entre las visualizaciones desarrolladas está por ejemplo un gráfico que pone cada *assessment* del curso en una línea. Dicha línea está dividida por tantas porciones como número de ejercicios tenga dicho *assessment*. Cada una de esas porciones tendrá un color que representará cuan bien lo han hecho los alumnos en dicho ejercicios, siendo por ejemplo rojo si menos de un 25% han contestado adecuadamente dicho ejercicio. De esta manera, un profesor puede conocer cuáles son los ejercicios más problemáticos del curso. Además, también se muestra la media de intentos que necesitaron los alumnos para hacer bien el *assessment*.
- Recomendaciones. Entre las recomendaciones se encuentran el repasar los conceptos cubiertos por cuestiones donde los usuarios no lo hicieron bien en los *assessments*. Los profesores son así alertados y pueden obtener más información a través de las visualizaciones.

Por otro lado, se necesitaba validar estos desarrollos con usuarios que accedieran a GCB para realizar los cursos. Sin embargo, no era factible realizar cursos reales en GCB antes de la finalización de la asignatura de Máster para que accedieran los alumnos a probar las herramientas y servicios implementados. Por ello, los alumnos que desarrollaron el proyecto crearon un generador de alumnos simulados. Este generador de alumnos simulados, simulaba multitud de alumnos accediendo a la aplicación y realizando muchas interacciones con la plataforma GCB, en concreto con respecto a los dos aspectos que se querían testear en este caso: interacciones con vídeos y con *assessments*. El programa realizado relacionado con este generador de alumnos está escrito en *python* y genera peticiones de bajo nivel HTTP. Las peticiones llegan al servidor en formato JSON. Los datos generados siguen una distribución uniforme. En un futuro se pueden pensar en otras distribuciones para la generación de alumnos ficticios, de forma que por ejemplo pudieran estar acordes a ciertos modelos pedagógicos.

Es remarcable que el generador ficticio de alumnos es un aspecto que es ortogonal a *learning analytics*, pero que adquiere una especial relevancia en ésta área, ya que se requieren gran cantidad de datos y diferentes posibilidades para el testeo de la aplicación, así como para poder mostrar ejemplos. Este generador se puede aplicar a otros desarrollos.

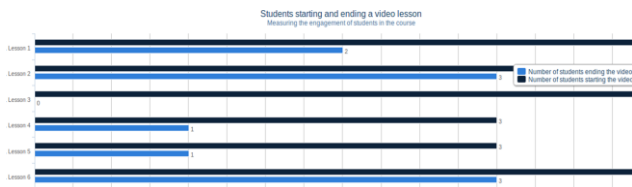


Fig. 8. Visualización de número de alumnos que empezaron y terminaron cada vídeo

VII. EXTENSIÓN DE GCB PARA INTEGRACIÓN CON MOZILLA BADGES

El aprendizaje a lo largo de la vida va dotando al alumno de una serie de competencias que le son proporcionadas desde diferentes instituciones, bien a través de cursos presenciales como de cursos a distancia, bien según un plan de enseñanza reglada como de manera informal. Estas competencias deben poderse certificar o justificar con garantías, de forma que los alumnos puedan convencer a empleadores u otras terceras partes sobre las mismas. En esta línea surgió el proyecto Mozilla OpenBadges [14] que permite la creación de un repositorio centralizado y asociado a cada alumno en el que las instituciones formativas puedan emitir justificantes de adquisición de competencias y asociarlas a un alumno, de forma que este las pueda mostrar a otras personas y éstas puedan validar su autenticidad. La integración de GCB con la plataforma de Mozilla OpenBadges permite mejorar las capacidades con las que dotar a una plataforma de MOOCs para que los alumnos que superen de forma exitosa la evaluación de un curso on-line masivo y abierto puedan obtener el reconocimiento del mismo a través de una plataforma que centralice los logros de dichos alumnos.

La integración de GCB con OpenBadges presentaba algunos retos de interés que hacían del proyecto una tarea compleja, al menos para conseguir una integración completa. La estructura de datos del alumno dentro de GCB debía extenderse para ir añadiendo referencias a los badges emitidos dentro de los cursos. A la vez, se tenían que generar URLs permanentes dentro de la plataforma para capturar las garantías asociadas a cada badge emitido. La generación de estas credenciales requería una programación conjunta cliente-servidor de forma que los badges emitidos en el servidor de cursos, una vez generados usaran las APIs cliente Javascript para su comunicación con los servidores de Mozilla OpenBadges. Finalmente, de cara a mejorar la experiencia de los usuarios, era interesante tener en cuenta la integración de la autenticación de usuario dentro de GCB y de Mozilla Persona. Para conseguir los retos anteriores, el equipo de alumnos que desarrollaron el proyecto tuvo que añadir las siguientes funcionalidades a GCB:

- **AnswerHandlerBadge:** esta es la clase principal que valida los criterios necesarios para la emisión de una nueva credencial al alumnos. Se permite la definición de diferentes criterios a cumplir por el alumno como la superación de un umbral mínimo en la puntuación de un curso para poder emitir un determinado badge.
- **BadgeModel:** permite definir la información y su codificación JSON para los badges. Almacena de forma persistente los datos para los badges emitidos.

- **Badges:** modelo sin estado para la generación de los ficheros JSON para cada badge.
- **Badgetest:** se utiliza para la validación del sistema de emisión de credenciales.

Dada las limitaciones de tiempo en el curso y la dificultad que suponía la integración para autenticación única del usuario mediante la integración de GCB con Mozilla Persona, este objetivo no pudo finalizarse a tiempo dentro del marco del proyecto, pero sí sirvió para comprender muchos de los aspectos de funcionamiento internos a GCB.

VIII. CONCLUSIÓN

En este artículo, se describen dos aportaciones:

- El desarrollo de tres extensiones para la plataforma de MOOCs GCB, que complementa de forma interesante las funcionalidades de esta plataforma. Los desarrolladores de Google nos comentaron que como resultado de esta experiencia habían obtenido algunas ideas para el desarrollo de algunas características de GCB.
- Una docencia basada en proyectos de la vida real y con utilidad real, en la que enfatiza el *learning by doing*. Se potenciaron competencias como la creatividad, el trabajo en grupo, la resolución de conflictos, la presentación oral del trabajo realizado, la documentación, la discusión con expertos, el trabajo con contextos técnicos cambiantes, la orientación al mercado, etc.

Todas las partes expresaron su satisfacción por la orientación de la asignatura, lo cual nos anima a seguir en esta línea.

AGRADECIMIENTOS

Agradecemos el trabajo realizado por los estudiantes de este curso. Ellos son los principales artífices del trabajo desarrolla-do. Elsa Bartolomé, Diego García Martín, Ostap Kovalisko y Rosbel Serrano integraron el equipo del primer proyecto. Roderick E.A. Fadou, Omar Khaled Aly, Miriam Marciel y Foivos Michelinakis formaron parte del segundo. Los integrantes del tercer equipo fueron Ahmed Ayman, Loay A.E. Elgindy, Nour Gharib, Youssef M. Safwat y José Luis Nieto.

Agradecemos también especialmente al equipo de Google Course Builder, sin cuya participación no podíamos haber llevado a cabo la experiencia de esta forma: Pavel Simakov, como ingeniero jefe, John Cox, John Orr y Sean Lip, como ingenieros que apoyaron cada uno de los tres proyectos. En cualquier caso, Google no tiene ninguna responsabilidad sobre el código desarrollado. También agradecemos la participación de Javier Arias de Google, Francisco Gómez Molinero de VisualTools y Jorge Ruiz Magaña de SoluSoft en ésta y anteriores ediciones de la asignatura.

Este trabajo ha sido parcialmente financiado por los proyectos EEE (TIN2011-28308-C03-01) del Plan Nacional de I+D+i y eMadrid (S2009/TIC-1650) de la Comunidad de Madrid. El primer autor agradece una ayuda de movilidad para profesores de la Fundación CajaMadrid para realizar una estancia en Harvard y MIT en el curso 2012/13.

REFERENCIAS

- (Todas las URLs han sido consultadas en Diciembre 2013)
- [1] S. Khan, "What College Could Be Like," *Comm. ACM*, vol. 56, no. 1, pp. 41-43, Jan. 2013
 - [2] C. Delgado Kloos, O. Kovalisko, E. Bartolomé, D. García, "An Extension for Google Course Builder for Authoring GIFT Assessments," en *Proc. SINTICE 2013: XV Simp. Internac. de Tecnologías de la Información y las Comunic. en la Educación*, Madrid, 2013, pp. 14-20
 - [3] M. Marciel, F. Michelinakis, R. Fanou, P.J. Muñoz-Merino, "Enhancements to Google Course Builder: Assessments Visualisation, YouTube Events Collector and Dummy Data Generator," en *Proc. SINTICE 2013: XV Simp. Internac. de Tecnologías de la Información y las Comunicaciones en la Educación*, Madrid, 2013, pp. 6-13.
 - [4] Udacity, <http://udacity.com>
 - [5] Coursera, <http://coursera.com>
 - [6] edX, <http://edx.org>
 - [7] NovoEd, <http://novoed.com>
 - [8] MiriadaX, <http://miriadax.net>
 - [9] Iversity, <http://iversity.org>
 - [10] FutureLearn, <http://futurelearn.com>
 - [11] Open2Study, <http://open2study.com>
 - [12] FUN, <http://france-universite-numerique.fr>
 - [13] XuetaangX, <http://xuetaangx.com>
 - [14] OpenUpEd, <http://openuped.eu>
 - [15] OpenBadges, <http://openbadges.org/>
 - [16] I. Gutiérrez, C. Delgado Kloos, R.M. Crespo, "Assessing Assessment Formats: The Current Picture", en *EDUCON 2010: IEEE Annual Global Engineering Education Conference*, 14-16 Abril 2010, Madrid
 - [17] IMS Global Learning Consortium: IMS QTI. <http://www.imsglobal.org/question/>
 - [18] Moodle, MoodleXML, docs.moodle.org/24/en/Moodle_XML_format
 - [19] Half-baked Software, HotPotatoes, <http://hotpot.uvic.ca>
 - [20] The Open University, OpenMark, www.open.ac.uk/openmarkexamples/
 - [21] DocBook, <http://www.docbook.org>
 - [22] Blackboard, <http://ondemand.blackboard.com/assess.htm>
 - [23] John Gruber, Markdown, <http://daringfireball.net/projects/markdown/>
 - [24] Moodle. GIFT format. [http://docs.moodle.org/24/en/GIFT_format](https://docs.moodle.org/24/en/GIFT_format)
 - [25] Microformats. GIFT format. <http://microformats.org/wiki/gift>
 - [26] GIFT reference. http://buypct.com/6_reference.pdf
 - [27] D. Clow, The learning analytics cycle: closing the loop effectively, 2nd Internat. Conf. Learning Analytics and Knowledge, pp. 134-138, 2012
 - [28] M. Wolpers, J. Najjar, K. Verbert, and E. Duval, Tracking actual usage: the attention metadata approach. *Education Technology & Society*, vol. 10, no. 3, pp.106-121, 2007
 - [29] P.J. Muñoz-Merino, M. Wolpers, K. Niemann, M. Friedrich, A. Pardo, C. Delgado Kloos, and M. Muñoz-Organero, CAM in the semantic web world, I-SEMANTICS Conference, 2010
 - [30] P.J. Muñoz-Merino, J.A. Ruipérez Valiente, C. Delgado Kloos, Inferring higher level learning information from low level data for the Khan Academy platform. *Proceeding LAK'13 Proc. Third International Conference on Learning Analytics and Knowledge*, pp. 112-116



Carlos Delgado Kloos (M'05-SM'06) es Dr. Ingeniero de Telecomunicación por la Universidad Politécnica de Madrid y Máster y Doctor en Informática (Dr. rer. nat.) por la Universidad Técnica de Múnich.

Es Catedrático de Ingeniería Telemática en la Universidad Carlos III de Madrid, en donde también es Coordinador del Grupo de Investigación GAST y del laboratorio Gradient, Director del Máster en Gestión y Producción en eLearning y Vicerrector de Infraestructuras y Medio Ambiente. Es autor de más de 200 publicaciones en revistas y congresos, ha participado en y coordinado una multitud de proyectos de investigación a nivel nacional e internacional y ha desempeñado numerosos cargos de gestión.

El Prof. Delgado Kloos es el representante nacional español en IFIP TC3 sobre Educación, el coordinador de la red eMadrid sobre e-learning en la Comunidad de Madrid y el Director de la Cátedra UNESCO "Educación Digital Escalable para Todos".



Pedro J. Muñoz-Merino (M'10) es miembro del IEEE desde 2010. En 2003 obtuvo el título de Ingeniero de Telecomunicación por la Universidad Politécnica de Valencia, y en 2009 el de Doctor en Ingeniería Telemática por la Universidad Carlos III de Madrid. Su campo de estudio fundamental es el de las tecnologías del aprendizaje.

Es profesor visitante en la Universidad Carlos III de Madrid, donde ha trabajado durante más de 10 años. Sus líneas de investigación actuales se centran fundamentalmente en los tutores inteligentes, la web semántica, analítica de aprendizaje y la gamificación. Ha realizado 2 estancias de investigación largas: una de más de 3 meses en Irlanda en la empresa Intel en 2005, y otra en Alemania en el Instituto de Tecnología Fraunhofer de más de 6 meses en 2009-2010. Asimismo, es autor de aproximadamente 50 publicaciones científicas y ha participado en más de 20 proyectos de investigación.

Desde mayo de 2012 está acreditado por ANECA como profesor Titular de Universidad. Además, ha obtenido el premio AMPER otorgado por el Colegio Oficial de Ingenieros de Telecomunicación (COIT) al Mejor Proyecto Fin de Carrera en Ingeniería Telemática, así como el premio Extraordinario de Doctorado por la Universidad Carlos III de Madrid.



Mario Muñoz-Organero (M'08) obtuvo el título de Ingeniero de Telecomunicación en la Universidad Politécnica de Cataluña (UPC) en 1996, el título de Master en Administración de empresas (MBA) en la UNED en 2002 y el de Doctor Ingeniero de Telecomunicación en la Universidad Carlos III de Madrid en 2004. Desde 2007 es Profesor Titular de Ingeniería Telemática en la Universidad Carlos III de Madrid, y desde 2012 subdirector de la Escuela Politécnica Superior, en esta misma universidad.

Ha publicado más de 100 artículos en revistas y congresos internacionales. Sus intereses incluyen las tecnologías middleware para el m-learning, las arquitecturas ubicuas para el e-learning y las arquitecturas abiertas orientadas a servicios. Desarrolla su labor investigadora dentro del grupo de Aplicaciones y Servicios Telemáticos (GAST) del que es miembro del equipo gestor desde 2007. Ha participado como investigador principal en varios proyectos nacionales e internacionales como el proyecto GEEWHEZ financiado por la UE dentro del FP7 o los proyectos ARTEMISA, OSAMI, HAUS, REME-DISS, IRENE y COMINN dentro del Plan Nacional de I+D financiados por el Gobierno de España. Ha participado como miembro del equipo investigador en más de 15 proyectos tanto nacionales como internacionales.