

Diseñando un MOOC en edX: Introducción a la Programación con Java – Parte 1

Carlos Alario-Hoyos, Carlos Delgado Kloos, Iria Estévez-Ayres, Carmen Fernández-Panadero, Jorge Blasco, Sergio Pastrana, Guillermo Suárez-Tangil, Julio Villena-Román

Departamento de Ingeniería Telemática y Departamento de Informática
Universidad Carlos III de Madrid

Avda. de la Universidad 30, 28911, Leganés (Madrid), España

calario@it.uc3m.es, cdk@it.uc3m.es, ayres@it.uc3m.es, mcfp@it.uc3m.es, guizos@gmail.com,
spastran@inf.uc3m.es, guillermo.suarez.tangil@gmail.com, jvillena@it.uc3m.es

Resumen- Los MOOCs (*Massive Open Online Courses* – Cursos Online Masivos Abiertos) han supuesto una revolución en la educación. Diseñar y poner en marcha un MOOC es una tarea compleja y además dependiente fuertemente del tipo de curso y de la plataforma donde éste se vaya a desplegar. Este artículo presenta el diseño del MOOC “*Introducción a la Programación con Java – Parte 1: Empezando a Programar en Java*” desplegado en la plataforma edX y puesto en marcha a finales de abril de 2015. Este curso de cinco semanas introduce los principios básicos de la programación utilizando como herramienta el lenguaje Java. El MOOC ha surgido de la colaboración entre los departamentos de Ingeniería Telemática e Informática de la Universidad Carlos III de Madrid. En el momento de su apertura el curso contaba con más de 60.000 alumnos de 190 países (según los datos proporcionados por edX) y en el momento de la escritura de este artículo el MOOC todavía se encontraba en el periodo de impartición. Además de presentar el diseño del curso, este artículo extrae algunas de las principales lecciones aprendidas por parte del equipo docente tras el diseño y puesta en marcha de este MOOC.

Palabras Clave- MOOC, edX, programación, Java.

I. INTRODUCCIÓN

Los MOOCs (*Massive Open Online Courses* – Cursos Online Masivos Abiertos) han supuesto un cambio muy importante en la forma en la que se aprende y se enseña hoy en día en la práctica totalidad de áreas de conocimiento [1]. Los MOOCs han supuesto una oportunidad para acceder a contenidos de alta calidad sobre los que construir nuestro aprendizaje acerca de una determinada materia, y al mismo tiempo han permitido crear un escaparate en el que los profesores pueden extender su docencia fuera del aula, a lo largo de todo el mundo [2]. Además, las tecnologías, formatos y lecciones aprendidas de los MOOCs, pueden utilizarse también para mejorar la docencia en el campus a través de los conocidos como SPOCs (*Small Private Online Courses* – Cursos Online Pequeños Privados) [3].

La Universidad Carlos III de Madrid (UC3M) ha apostado fuertemente por incorporar los MOOCs (y los SPOCs) a su oferta educativa desde 2012 [4][5], participando en dos de las principales iniciativas MOOC: edX [6] y MiríadaX [7]. Las primeras experiencias con MOOCs se llevaron a cabo a principios de 2013 en la plataforma MiríadaX ofreciendo tres cursos en español [8]. Desde febrero de 2014 la UC3M forma parte de edX, en cuya plataforma ha comenzado a lanzar sus primeros cursos en inglés y español a principios de 2015.

El apoyo de la UC3M a la creación de MOOCs (y SPOCs) entre su personal docente se ha materializado también en la creación de la Unidad de Tecnología Educativa e Innovación Docente (UTEID) [9], cuyo objetivo principal es apoyar a profesores que desean poner en marcha MOOCs y SPOCs asesorando en cuestiones tecnológicas, audiovisuales, de creación de contenidos, de reutilización de contenidos de terceros o de despliegue en diferentes plataformas, entre otras. La labor realizada por la UTEID se ha recogido y publicado en forma de buenas prácticas [10][11][12][13].

En este artículo se presenta el diseño del MOOC “*Introducción a la Programación con Java – Parte 1: Empezando a Programar en Java*” (“*Introduction to Programming with Java - Part 1: Starting to Programming in Java*”) [14] realizado a través de la colaboración entre los departamentos de Ingeniería Telemática e Informática de la UC3M y con el apoyo de la UTEID y desplegado en la plataforma edX (sección II). Además, en este artículo se recogen un conjunto de lecciones aprendidas por el equipo docente durante el proceso de diseño y puesta en marcha de este MOOC (sección III). El artículo finaliza recogiendo las conclusiones de este proceso (sección IV).

II. DISEÑO DEL MOOC

El MOOC “*Introducción a la Programación con Java – Parte 1: Empezando a Programar en Java*” tiene una duración de cinco semanas de contenidos de aprendizaje y fue lanzado el 28 de abril de 2015, por lo que en el momento de escritura de este artículo todavía se encuentra en el periodo de impartición. El MOOC tiene previsto terminar el 30 de junio dando un periodo extra sobre las cinco semanas iniciales para que los alumnos, especialmente los que se incorporan a última hora, terminen las actividades de evaluación. El curso arrancó con más de 60.000 alumnos registrados de 190 países. Este MOOC es el primero de una serie de tres módulos (MOOCs) dedicados a enseñar los principios básicos de la programación utilizando como herramienta principal el lenguaje Java. El MOOC se ha desplegado en la plataforma edX y se imparte enteramente en inglés, aunque los vídeos en los que se presentan los contenidos teóricos están subtítulos en español, portugués y chino mandarín, además de en inglés. La carga de trabajo estimada del alumno es de 5-7 horas de trabajo a la semana.

A. Ciclo de trabajo

El equipo docente del MOOC lleva a cabo su diseño y despliegue en colaboración con la UTEID. Para ello se establece un ciclo de trabajo en el que el equipo docente se encarga de generar la estructura del curso en la plataforma *Studio Edge* [15] (donde *Studio* es la herramienta de autoría de edX y *Edge* es la instalación de pruebas de edX como paso a la migración definitiva a edx.org); graba los vídeos utilizando los equipos y salas de audiovisuales gestionadas por la UTEID; genera las actividades y laboratorios; y monta la versión final de los vídeos y ejercicios en *Studio Edge*. Por otro lado, la UTEID se encarga de producir las versiones finales de los vídeos (añadiendo transparencias, efectos y animaciones cuando procede); crea, corrige y gestiona los subtítulos asociados a cada vídeo; añade los componentes de discusión (foro de edX) relacionados con cada contenido y la pestaña correspondiente a redes sociales a utilizar en el curso (Twitter en este caso); migra de *Edge* a edX antes del comienzo del curso; y mantiene copias de seguridad del curso regularmente. La UTEID además da soporte adicional en ciertas tareas como la generación de algunos ejercicios especiales que requieren animaciones, o el despliegue de ejercicios con cierta complejidad en edX como los de evaluación entre pares.

Como sistema de compartición y gestión de contenidos se utiliza una herramienta propia de la UC3M llamada GEL [4]. GEL permite almacenar vídeos y publicarlos automáticamente en YouTube y al mismo tiempo en un servidor propio de la UC3M. Además se utiliza Google Drive para hacer un seguimiento del estado de cada uno de los materiales, compartir y editar guiones y transparencias de cada uno de los vídeos, compartir y editar ejercicios, y compartir y editar los mensajes a enviar a los alumnos.

B. Visión general y estructura

Los tres primeros pasos en el diseño y despliegue de un MOOC en edX son críticos ya que permiten promocionar el curso y captar potenciales alumnos. Estos son: la creación de una “*about page*” o *página de presentación* estándar con la información general del curso (descripción breve, objetivos, descripción del profesorado, carga semanal, fechas de impartición, idioma de impartición); la creación de un *video de presentación* atractivo sobre el curso de entre uno y dos minutos que se añade a la página de presentación; y la creación del “*syllabus*” o *ficha del curso* en la que se incluye al menos la descripción y objetivos detallados, la estructura del curso desglosada por semanas, el sistema de evaluación, el calendario del curso y se indica también cómo se llevará a cabo la comunicación del profesorado con los alumnos. En el caso de este MOOC se contrataron los servicios profesionales de una empresa externa a la universidad para el vídeo de presentación, y además se fue actualizando la versión del “*syllabus*” a medida que se iban diseñando las diferentes semanas del mismo.

El MOOC consta de cinco semanas con una estructura similar precedidas de una semana 0. La semana 0 se utiliza para incluir un par de vídeos de introducción y motivación relacionados con la temática del curso, una encuesta para conocer mejor a los alumnos y un componente de discusión para que estos puedan presentarse a sus compañeros. Cada semana (o sección en terminología edX) tiene una estructura similar y está dividida en subsecciones. Cada subsección tiene una o varias unidades y cada unidad puede tener uno o varios componentes (p.ej. un componente de vídeo, otro de texto y otro de discusión). Las subsecciones pueden clasificarse de la siguiente manera (ver Figura 1 para el detalle de las subsecciones para la primera semana):

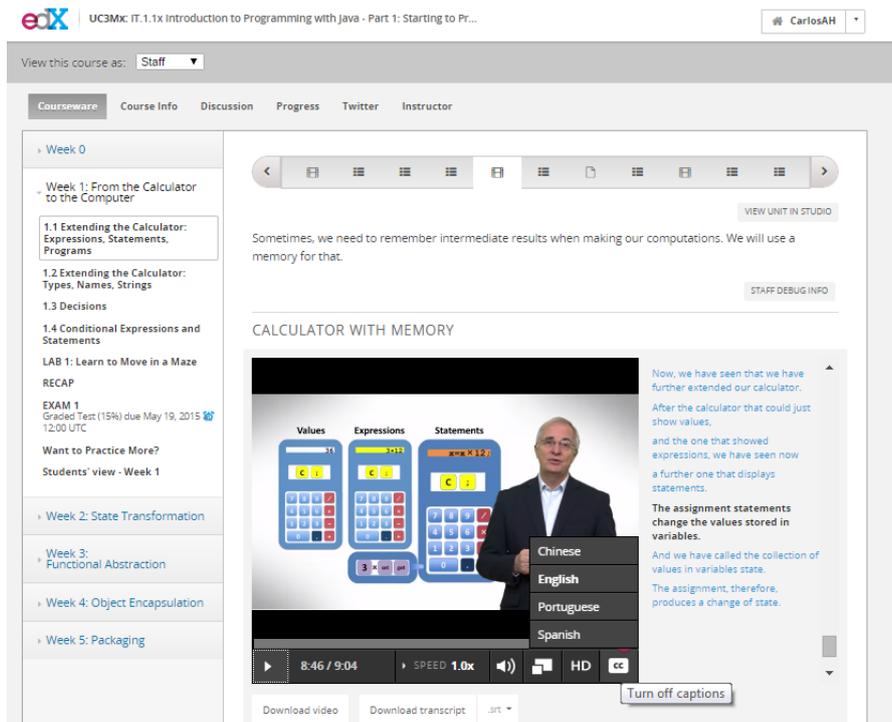


Fig. 1. Captura de pantalla del MOOC. En la barra izquierda se observa la estructura del curso (*Courseware*) y el detalle para la primera sección o semana (“*Week 1: From the Calculator to the Computer*”) con sus subsecciones. En la barra superior derecha se observan las diferentes unidades para el caso de la subsección 1.1 (“*Extending the Calculator: Expressions, Statements, Programs*”). En la parte inferior derecha se presenta uno de los vídeos (*Calculator with Memory*) con subtítulos en inglés, español, portugués y chino; tanto el vídeo como los subtítulos pueden descargarse.

- **Contenidos principales de la semana.** Estos se distribuyen en cuatro subsecciones (o cuatro cuartos) cada uno de ellos presentando y ejercitando conceptos relacionados. En la Figura 1 puede verse el caso de la primera semana con las subsecciones 1.1 a 1.4. Cada una de estas subsecciones contiene varias unidades en las que se incluyen vídeos (típicamente de 2 a 5 vídeos cortos por subsección) y múltiples ejercicios relacionados con los conceptos explicados en los vídeos. Estos ejercicios pueden desarrollarse sobre las herramientas propias de edX o sobre herramientas externas como se detalla más adelante. Además, puede haber explicaciones complementarias textuales sobre los conceptos explicados e incluso código que los alumnos tienen que descargar para trabajar sobre él.
- **Contenidos de laboratorio (Lab).** Todas las semanas incluyen varios ejercicios relacionados entre sí, agrupados bajo la subsección laboratorio, en la cual se ejercitan los contenidos aprendidos en la semana correspondiente. Las claves del diseño del laboratorio son las mismas que en las que se basan muchos juegos [16]: motivación voluntaria del alumno, fallar pronto y de una forma divertida; aplicar lo que se ha aprendido para alcanzar un objetivo. Estos tres principios se aplican en dos niveles de granularidad: en cada ejercicio individual; y en el laboratorio global, el cual sigue una estructura incremental a lo largo de las cinco semanas de tal forma que se van integrando diferentes ejercicios hasta conseguir un juego completo. La metodología aplicada en el laboratorio está basada parcialmente en la metodología PhyMEL (*Physical, Mental and Emotional Learning*) [17] según la cual al principio se hacen ejercicios cortos de entrenamiento que luego se combinan para conseguir un proyecto global y en el que se intentan evocar aspectos: físicos (p.ej. un muñeco que se mueve por un laberinto); emocionales (satisfacción por las recompensas a corto plazo y a largo plazo como conseguir superar los ejercicios individuales y el proyecto global); y mentales (comprender conceptos y algoritmos relacionados con la temática de cada semana). Por restricciones en los tipos de ejercicio propios de la plataforma, en el laboratorio se utilizan herramientas externas a edX, las cuales se detallan más adelante.
- **Recapitulación (Recap).** En esta subsección se proporciona un resumen textual con los conceptos más importantes que han aparecido a lo largo de la semana para que sirva de apoyo al trabajo del alumno. En esta subsección también se proporcionan las soluciones a los ejercicios más complejos planteados durante la semana, particularmente aquellos en los que se utilizan herramientas externas a edX.
- **Examen (Exam).** En esta subsección se plantea un examen con varios ejercicios de corrección automática sobre los temas tratados en la semana. Los alumnos tienen que realizar el examen en un periodo de tiempo delimitado (típicamente dos semanas después de la apertura de los contenidos de esa semana).
- **Ejercicios extra (Want to practice more?).** Se ofrece a los alumnos una subsección con un conjunto de ejercicios adicionales sobre esa semana para aquellos que quieran continuar trabajando y ejercitando todo lo explicado en la semana.
- **Visión de los estudiantes (Students' view).** Una vez finalizada la semana se ofrece a los alumnos en esta subsección un vídeo resumen de unos 5-8 minutos grabado por estudiantes del Colegio Americano de Madrid (16-17 años) que han trabajado los contenidos de la semana y que comentan informalmente sus impresiones en parejas o grupos de tres con un lenguaje más cercano al alumno medio que cursa este MOOC. En los vídeos, los estudiantes explican con sus propias palabras algunos de los conceptos más complejos de la semana. Además, comentan aquellas actividades o vídeos que más les han gustado y cuáles consideran que son mejorables. El objetivo principal de estos vídeos es conseguir que el alumno que participa en el MOOC vea reflejado su propio proceso de enseñanza-aprendizaje en los comentarios y reflexiones de estudiantes con un nivel semejante al suyo. Se pretende conseguir un aumento de motivación de los alumnos, además, de fomentar la reflexión del alumno acerca de los contenidos de la semana y de su propia evolución en el curso.
- **Evaluación entre pares (peer assessment).** En dos de las semanas (semanas 3 y 5) se incluye entre el examen y la subsección de ejercicios extra una subsección adicional con un ejercicio de evaluación entre pares, tal y como se explica más adelante, en el que los alumnos del MOOC deben escribir un pequeño fragmento de código en Java que es evaluado por sus propios compañeros.

C. Vídeos

Los vídeos son los elementos centrales de las explicaciones teóricas en este MOOC. En total se han creado 64 vídeos con contenido propio y desde cero, los cuales se distribuyen a lo largo de las cinco semanas del curso. Todos los vídeos tienen una duración mayor de 3 minutos y menor de 9 minutos, dependiendo de los contenidos a tratar. La mayor parte de los vídeos está en el rango de los 5-7 minutos, el cual es un criterio habitual de buenas prácticas en este tipo de cursos. Por ejemplo, la duración óptima de un vídeo para maximizar la motivación del alumno es de unos 6 minutos según edX [18], aunque ésta puede variar en función de la materia explicada.

Todos los vídeos están grabados en inglés, aunque se ofrecen con subtítulos en cuatro idiomas: inglés, español, portugués y chino mandarín. Para ello se ha realizado una colaboración con la empresa *3Play Media* a través de edX, la cual se ha encargado de proporcionar los subtítulos en inglés, portugués y chino mandarín, mientras que la UTEID se ha encargado de realizar las transcripciones al español. De esta forma se busca mejorar la experiencia del alumno de este MOOC, independientemente de su país de origen y lengua materna. Un paso importante y necesario en el ciclo de trabajo del subtítulo es realizar primero las transcripciones en inglés y que un profesor del curso se encargue de corregirlas manualmente, antes de que pueda pasarse a la fase de transcripción a los otros tres idiomas, ya que siempre se detectan algunos fallos de transcripción en la terminología propia del curso.

Todos los vídeos del curso se ofrecen en alta resolución y en baja resolución. Tanto los vídeos en alta y en baja resolución, como los subtítulos en los diferentes idiomas se ofrecen a los alumnos para su descarga. El hecho de ofrecer vídeos con baja resolución está motivado para facilitar que

aquellos alumnos que tengan un menor ancho de banda o cuyos proveedores de servicios de Internet impongan limitaciones en sus tarifas de descarga puedan también seguir cómodamente el curso. Los vídeos se alojan en YouTube y en un servidor propio de UC3M (para su descarga) y se enlazan desde edX. Además, edX se ha encargado de convertir los vídeos del MOOC para que puedan ser accesibles desde la aplicación móvil de edX, la cual está disponible para Android y iOS.

A lo largo de los diferentes vídeos aparecen varios de los profesores del curso, lo cual puede conllevar dificultades debido a los distintos estilos de cada uno de ellos. Sin embargo, se ha hecho un esfuerzo desde el diseño del curso para homogeneizar el estilo de los vídeos y mantener un mismo nivel de calidad. Por ejemplo, la mayoría de los vídeos están grabados en un estudio de grabación con un croma detrás y utilizan transparencias o animaciones de apoyo. Algunos vídeos utilizan únicamente capturas de pantalla (mostrando por ejemplo el funcionamiento de un trozo de código), por lo que no era necesario el uso del estudio de grabación, aunque siempre se ha utilizado un micrófono profesional para minimizar posibles ruidos y mejorar la calidad del audio. Uno de los problemas al grabar con croma y posteriormente eliminarlo en producción es que el color blanco del fondo puede dar una sensación de vacío al alumno que ve el vídeo. Por ello se ha incluido en todos los vídeos una “viñeta” elíptica que desvanece del color blanco al negro en los bordes del vídeo. Además, se ha incorporado en todos los vídeos un sonido característico hacia el final del mismo que da pie a un breve resumen de lo que se ha tratado en el vídeo.

El contenido de los vídeos propiamente dicho suele amenizarse con la utilización de anécdotas de la vida real (p.ej. la toma de decisiones en un programa ejemplificada a través de la decisión del profesor de salir de casa con un paraguas dependiendo de si llueve o no). Además, se utilizan objetos del mundo físico para acompañar los vídeos (en el ejemplo anterior un paraguas que el profesor lleva en el vídeo, o unas muñecas rusas que el profesor utiliza para explicar el concepto de recursión) y se cuida que el profesor tenga un tono agradable. Los vídeos van precedidos siempre de un breve texto en la plataforma, el cual sirve para resumir brevemente lo que se va a contar o plantear una pregunta de reflexión.

Además de los 64 vídeos de contenidos propios del curso se han generado: un vídeo de introducción; un vídeo de cierre y despedida de los profesores; y cinco vídeos con la visión de los estudiantes grabados con alumnos del Colegio Americano de Madrid. En la medida de lo posible se ha intentado ofrecer también estos vídeos en alta y baja calidad y con subtítulos en los distintos idiomas.

D. Actividades

El MOOC ha sido diseñado para fomentar la interacción del alumno con los materiales y enseñarle a programar progresivamente a través de múltiples actividades. De hecho en cada semana se han diseñado más de 50 actividades de diferente duración y dificultad las cuales forman parte de las subsecciones: contenidos principales, contenidos de laboratorio, examen y ejercicios adicionales. En el caso de las subsecciones de contenidos principales, las actividades se

entrelazan con los vídeos, de tal forma que ayudan a afianzar los conceptos que acaban de ser explicados. En el caso de los contenidos de laboratorio las actividades se enmarcan dentro de una historia denominada la “mansión encantada” y durante la cual el alumno tendrá que aprender a moverse por laberintos tomando decisiones y recurriendo a repeticiones de acciones a través de bucles, programar un algoritmo de “backtracking” (o vuelta atrás) para resolver un laberinto y aprender los principios de orientación a objetos creando personajes (objetos) que le ayuden en su misión en la mansión. En el caso de la subsección de ejercicios adicionales, éstos se plantean como retos para aquellos alumnos más ambiciosos o que ya tienen conocimientos previos y que por tanto necesitan menos tiempo para completar el resto de ejercicios.

Para diseñar las actividades del MOOC se han utilizado tanto herramientas propias de edX como herramientas externas a edX. Las herramientas externas estaban o bien integradas en edX (mediante código JavaScript, a través de un iFrame, o a través del estándar LTI [19] – *Learning Tools Interoperability* – el cual es soportado por edX), o bien requerían que el alumno se las descargase e instalase en su máquina. Para la evaluación sumativa de los alumnos únicamente se han utilizado herramientas propias de edX, debido a la dificultad de integrar las herramientas externas con el sistema de calificaciones de edX.

Dentro de las herramientas propias de edX destacan las de corrección automática. Entre ellas se han utilizado los siguientes tipos: problema de respuesta múltiple (*multiple choice problem*), problema de casillas de verificación (*checkbox problem*), problema de lista desplegable (*dropdown problem*), problema de entrada de texto (*text input problem*) y problema de arrastrar y soltar (*drag and drop problem*). De estos merece la pena destacar los dos últimos. Los problemas de entrada de texto deben usarse con mesura ya que son bastante conflictivos al requerir que el alumno proporcione una solución determinada y exacta tal y como se define en la configuración del ejercicio. Añadir espacios, utilizar mayúsculas y minúsculas indistintamente, o signos de puntuación puede conllevar que el sistema considere incorrecta una solución que puede ser perfectamente válida. Por otro lado, los problemas de arrastrar y soltar son problemas muy interactivos en los que el alumno tiene que arrastrar elementos desde una barra inferior a una imagen que aparece justo encima. El profesor define la parte de la imagen en la que ese elemento representa una solución válida durante la configuración de la actividad. Este tipo de actividades son costosas de diseñar, debido a la necesidad de definir manualmente el espacio de soluciones posibles, pero tienen una buena aceptación entre los alumnos debido a su interactividad. Un inconveniente importante de los problemas de arrastrar y soltar es que actualmente no funcionan bien en algunos dispositivos móviles como *tablets* y *smartphones*. La Figura 2 muestra un ejemplo (ya resuelto) de un problema de arrastrar y soltar en el que un alumno debe rellenar huecos en un código en Java utilizando los operadores que se le proporcionan en la barra inferior; algunos de los operadores deben utilizarse varias veces mientras que otros no deben utilizarse para resolver correctamente el problema.

CONDITIONALS (NON GRADED ACTIVITY) (1/1 point)

The body mass index (BMI) is a well-known number to classify people according to their body fat. It is calculated as the mass of the person (in kg) divided by the square of the height (in meters). The following code segment calculates the category of a person given his/her mass and weight. Complete the code using the operators that are shown below and taking into account the categories and BMI values shown in the table.

Note: `double` is another data type in Java like `int`, but that allows representing decimal values. We will see more about data types later in this course.

(This exercise may not be compatible with mobile devices: we recommend that you use a computer)

Category	BMI
Underweight	Less than 18.5
Normal weight	From 18.5 (inclusive) to 25 (exclusive)
Overweight	From 25 (inclusive) to 30 (exclusive)
Obese	From 30 on

```
double mass = 90;
double height = 1.90;
double bmi = mass / (height * height);
String category;

if (bmi < 18.5) {
    category = "underweight";
}
if (bmi >= 18.5 && bmi < 25) {
    category = "normal weight";
}
if (bmi >= 25 && bmi < 30) {
    category = "overweight";
}
if (bmi >= 30) {
    category = "obese";
}
```



Fig. 2. Ejemplo de problema de *drag and drop* (ya resuelto) donde el alumno debe rellenar los huecos en el código de la parte superior con los operadores proporcionados en la parte inferior.

Otra de las herramientas propias de edX utilizadas en este MOOC es la evaluación entre pares (*peer assessment – P2P*). En las dos actividades P2P incluidas en este curso se proporciona a los alumnos un enunciado de un código a resolver y se les da una “rúbrica” (o criterios) de evaluación detallada. Los alumnos suben su tarea resuelta a edX, el cual distribuye cada tarea a un número de compañeros determinado durante la configuración de la actividad (típicamente 3-5, pero solamente considerando aquellos que también han entregado la tarea). Para facilitar la evaluación entre pares es muy importante definir una “rúbrica” concreta, con las diferentes partes a evaluar claramente segmentadas, incluyendo los valores cuantitativos posibles para cada parte, y ejemplos para cada valor cuantitativo. Este tipo de actividades normalmente requieren dos plazos: un primer plazo para la entrega de la actividad y un segundo plazo para la revisión entre pares, por lo que es necesario dejar tiempo suficiente a los alumnos en ambos casos. En este MOOC las actividades de evaluación entre pares únicamente se han utilizado como parte de la evaluación sumativa del curso.

Además se han integrado varias herramientas externas desarrolladas en JavaScript por el equipo docente específicamente para el curso. Entre ellas destacan: varios tipos de calculadoras de complejidad variable (con operadores para números enteros, booleanos y ambos); un coche con control remoto; y un conversor de decimal a binario, incluyendo soporte a números negativos a través del sistema de representación en complemento a dos.

También se ha integrado la herramienta web de Google *Blockly* [20] (ver ejemplo en Figura 3), la cual permite crear juegos sencillos a través de una representación visual. Con *Blockly* el alumno ensambla bloques de pseudocódigo arrastrando y soltando desde un panel central hacia un panel en la parte derecha, y sin necesidad de conocer la sintaxis del

lenguaje de programación. *Blockly* se ha particularizado para generar una secuencia de laberintos de dificultad incremental en los laboratorios de la primera y segunda semana del MOOC adaptados a la temática de la mansión encantada.

Como herramienta de compilación (y ejecución) de código Java desde el navegador se ha integrado *Codeboard* [21] (ver ejemplo en Figura 4). Esta herramienta, integrada a través de LTI en edX, ha permitido la inclusión de numerosos ejercicios de apoyo, creados y salvados previamente por el profesorado, que el alumno puede cambiar, compilar y ejecutar de forma dinámica desde su navegador. La herramienta establece un tiempo máximo de ejecución para evitar problemas como bucles infinitos, muy comunes entre los alumnos que están aprendiendo a programar. Si el alumno comete un error o desea empezar de nuevo el ejercicio, basta con refrescar la página del navegador para encontrar otra vez el ejercicio en su estado inicial. En el caso de que el alumno quiera guardar los cambios realizados, puede abrir una cuenta en la plataforma; esta opción se les menciona a los alumnos, pero no es en absoluto obligatoria. La herramienta se ha usado como parte de los laboratorios a partir de la semana 2. Es importante comentar que esta herramienta externa no se usa para validar ni entregar código (aunque sería posible); sólo se usa para que los alumnos aprendan a través de la modificación (y creación) de pequeños programas.

A partir de la semana 4, y como parte de la programación orientada a objetos, se introduce la herramienta *Greenfoot* [22]. *Greenfoot* permite el desarrollo de juegos de forma simplificada y ha sido usado tanto para la elaboración de ejercicios relacionados con los contenidos principales de la semana, como de los laboratorios de las semanas 4 y 5. Esta herramienta debe ser descargada por el alumno e instalada en máquina, por lo que ha sido necesaria la inclusión de explicaciones acerca de la descarga e instalación *Greenfoot*. Al igual que con *Codeboard*, el profesorado ha creado pequeños ejercicios que los alumnos deben descargar y ejecutar. Además, se han realizado vídeos de apoyo sobre el uso de *Greenfoot* y como guía en la realización de las primeras actividades con esta herramienta (ver Figura 5).

MORE BRANCHING

We understand that branching may be a little bit tricky. Here you have another level so you can practice more. Take into account that in this level branches allow you to evaluate three different conditions. 7 blocks are allowed in this level but we are sure that you can do it with just 6.

(This activity may not be compatible with mobile devices: we recommend that you use a computer)

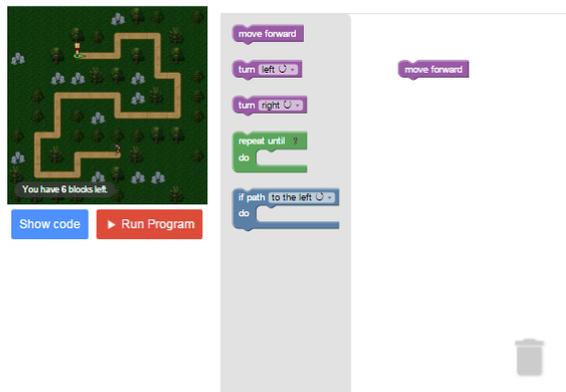


Fig. 3. Ejemplo de ejercicio con *Blockly* integrado en edX como parte del laboratorio de la semana 1 en el que los alumnos deben llegar al final de un laberinto utilizando bloques que representan movimientos, giros, bifurcaciones y bucles.

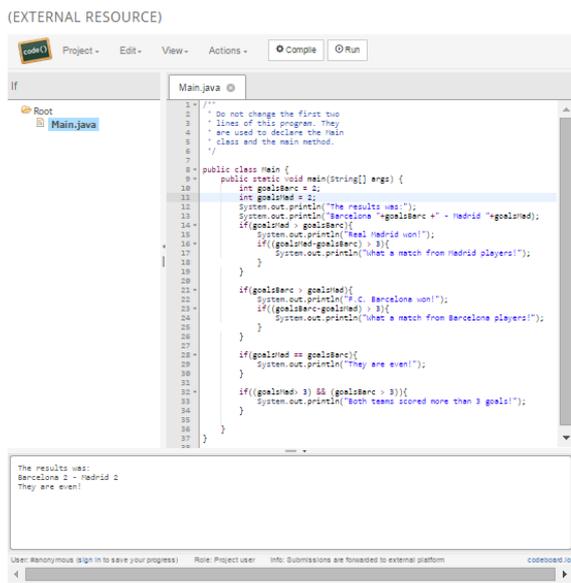


Fig. 4. Ejemplo de ejercicio con Codeboard integrado en edX como recurso externo a través de LTI.

E. Sistema de evaluación

El sistema de evaluación está basado exclusivamente en ejercicios realizados sobre herramientas propias de edX. Esta decisión se ha tomado para evitar depender en algo tan sensible de servicios ofrecidos por terceros ajenos a UC3M y edX. La evaluación sumativa consta de exámenes basados en ejercicios de corrección automática (5 exámenes), y de actividades más complejas de evaluación entre pares (2 actividades).

Cada semana los alumnos deben realizar un examen consistente en múltiples preguntas (entre 5 y 10), independientes entre sí, sobre los temas tratados durante la semana. Cada examen tiene un peso del 15% de la nota final, con lo que el peso total de los exámenes es de un 75% de la nota final. En la elaboración de estos ejercicios se han tenido en cuenta las limitaciones actuales de la plataforma. Por ejemplo, usando problemas de arrastrar y soltar (*drag and drop*) con múltiples opciones, si el alumno se equivoca en un solo campo, la pregunta es evaluada por la plataforma con un cero. Debido a esta limitación, se decidió minimizar el uso de preguntas de este tipo en los exámenes.

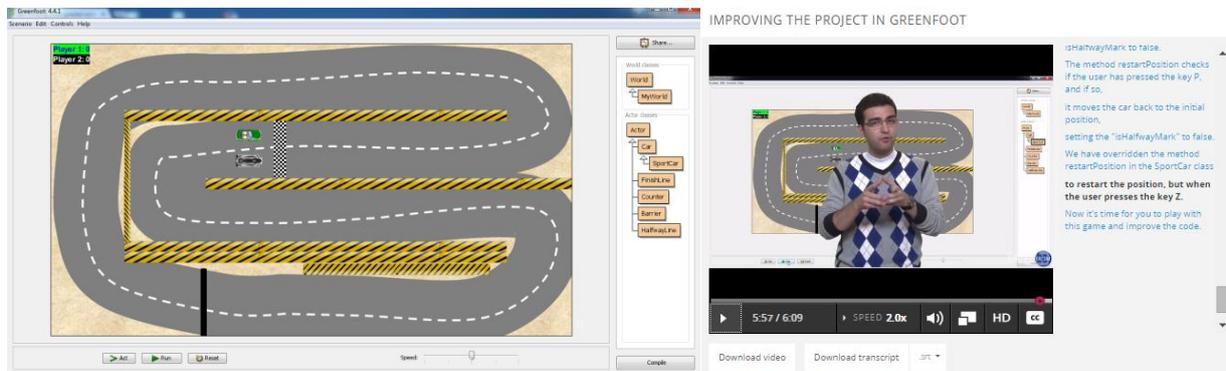


Fig. 5. Ejemplo de ejercicio realizado con Greenfoot (izquierda) en el que los alumnos desarrollan un pequeño juego de carreras de coches. Explicación del funcionamiento del ejercicio y del código a través de uno de los videos.

Además, se han programado dos actividades de evaluación entre pares, una en la semana 3 (10% de la nota final), justo antes de empezar con programación orientada a objetos, y otra en la semana 5 (15% de la nota final), al finalizar el curso. La actividad de evaluación entre pares de la semana 3 plantea un problema específico de los contenidos de esa semana, mientras que la actividad de la semana 5 plantea un problema global de todo el curso. En este tipo de actividades tanto el enunciado como la rúbrica de corrección se han cuidado especialmente. Además, al estar orientado el MOOC a neófitos en la programación, es muy probable que sea la primera vez que se enfrenten a la elaboración de un código desde cero, por lo que se han planteado actividades muy guiadas.

F. Comunicación con los alumnos

La comunicación con los alumnos se ha estructurado de la siguiente manera:

- **Mensaje semanal.** Al inicio de cada semana, y coincidiendo con la liberación nuevos materiales, se envía un mensaje (que reciben los alumnos en el correo que han proporcionado a la plataforma) que presenta un resumen de las actividades y conceptos más importantes que se cubrirán durante esa semana. Además, el mensaje sirve para recordar aspectos logísticos: plazos de entrega; la posibilidad de bajarse no sólo los vídeos y subtítulos, sino también las transparencias en PDF; la necesidad de consultar regularmente el “syllabus”, etc. A partir de la segunda semana los vídeos del Colegio Americano se anuncian también en el mensaje semanal. Se pretende con este mensaje motivar a los alumnos e invitarles a entrar en la plataforma para ver estos contenidos.
- **Información acerca del curso (Course Info).** Los mensajes semanales son publicados también en la página de información acerca del curso dentro de la plataforma. Esta página, además, contiene enlaces al “syllabus”, una guía de cómo usar los foros y un convertidor de hora local a UTC (que es el sistema seguido en el curso).
- **Uso de foros dentro de edX.** Se ha decidido usar como canal de comunicación principal con los alumnos los foros proporcionados por la propia plataforma.
- **Uso de twitter.** Para el curso se ha creado una cuenta específica en twitter (@javaedxuc3m) y se pide a los alumnos que usen el hashtag (#javaedxuc3m) para mantener conversaciones en esta red social.

III. ALGUNAS LECCIONES APRENDIDAS

Esta sección presenta algunas de las lecciones aprendidas por el equipo docente durante el proceso de diseño y puesta en marcha de este MOOC. Estas lecciones cubren tanto el proceso de diseño y puesta en marcha de un MOOC en general como la particularización para el caso de edX.

Una de las principales lecciones aprendidas se refiere a la *gestión del tiempo*. Es muy fácil subestimar la dedicación necesaria del profesorado en la creación de un MOOC. Es por ello necesario hacer una planificación concienzuda del tiempo a dedicar desde antes de comenzar la etapa de diseño (y ajustarse a ella), dejando tiempo suficiente para discutir el diseño del curso y generar los materiales, pero también para el testeado del curso y la subtitulación de los vídeos. Respecto a los vídeos es además muy importante tener en cuenta el precio tan costoso que supone realizar cambios a posteriori (volver a producir, cargar en la plataforma, ajustar el subtítulo...) por lo que deben revisarse concienzudamente antes de darlos por definitivos.

Este MOOC ha sido producido por un equipo de ocho docentes por lo que algunas de las lecciones aprendidas más valiosas se refieren a la *coordinación* del trabajo. A la hora de buscar la eficiencia es importante establecer reuniones específicas cada semana (por tareas o módulos) con las personas implicadas y que alguien asuma el rol de coordinador. Tener reuniones generales es importante al principio y al final para tener una visión global, pero en el día a día, y especialmente si la dedicación del profesorado no puede ser completa, es recomendable tratar de compartimentar. Además, es importante partir de un esquema inicial lo más detallado posible del contenido que se quiere ofrecer. Esto evita repetir ejercicios al repartir el trabajo y ayuda a guardar la coherencia a la hora de tratar ciertos conceptos. Es necesario hacer un buen reparto de ejercicios en grupos pequeños de trabajo y con revisiones cruzadas de personas que han desarrollado contenidos relacionados para minimizar los errores en dichos ejercicios. En esta línea, también es necesario tener un plan de contingencia ante posibles bajas del profesorado. En este curso se produjo la baja de tres de los nueve profesores iniciales de la universidad donde se impartía el MOOC y, aunque dos de ellos siguieron activamente participando en el MOOC, su grado de implicación no pudo ser el mismo que habían estimado inicialmente, generando una carga adicional para el resto de integrantes del equipo.

En cuanto a las *actividades* planteadas durante el curso cabe destacar la necesidad de tener en cuenta la importancia de disponer de herramientas fáciles de usar que permitan dotar al curso de contenidos relacionados con la materia que se quiere impartir. Por ejemplo, edX no cuenta a día de hoy con herramientas nativas que permitan trabajar con código Java (editar, compilar y ejecutar). Por esto motivo ha sido necesario integrar herramientas externas como *Codeboard*, o *Greenfoot*. Utilizar herramientas que pueden funcionar desde el navegador sin necesidad de instalar entornos pesados puede contribuir a disminuir el porcentaje de abandonos en cursos de iniciación como éste. Sin embargo, hay que tener en cuenta también las limitaciones de algunas herramientas a la hora de funcionar en ciertos dispositivos y navegadores incluyendo Blockly o Codeboard pero también los ejercicios de arrastrar y soltar (nativos en edX).

El hecho de plantear un *laboratorio*, proyecto o caso de estudio longitudinal al MOOC, es en general positivo, ya que permite reforzar conceptos y que los alumnos vayan encajando las piezas, construyendo soluciones incrementales, lo que suele tener un efecto positivo en la motivación de los alumnos. Por el contrario, una aproximación de estas características requiere una gran coordinación entre los diferentes ejercicios relacionados y cualquier modificación puede afectar transversalmente a todas las semanas.

Otros de los aspectos más importantes en un MOOC es la *componente social*. Es necesario dedicar tiempo durante la etapa de diseño a preparar una estrategia social para el curso. En esta estrategia se determinan las herramientas a utilizar (foros, Twitter...) pero también los roles de cada componente del equipo. El hecho de no haber planteado una estrategia social a tiempo puede suponer que se desaprovechen algunos de esos canales de comunicación con los alumnos. También es importante conocer las limitaciones que tienen cada uno de esos canales de comunicación. Por ejemplo, en el caso de edX, uno de los principales problemas encontrados en la comunicación con los alumnos en el foro de la propia plataforma es la desorganización de los diferentes hilos de discusión. En este sentido es muy difícil seguir una conversación sobre un tema, pues los hilos de discusión no pueden agruparse de forma sencilla. Para amortiguar este problema se decidió asociar un componente de discusión a cada unidad en edX, identificándolo mediante el título de la semana y el título de la unidad. Así, se ha conseguido, al menos, detectar problemas relacionados con vídeos y ejercicios concretos. Por otra parte, en edX el manejo y gestión del foro en sí mismo es bastante complejo (tanto para alumnos como para profesores). Por ejemplo, el profesor no puede reordenar los hilos de discusión para una mayor comprensión por parte de los alumnos. Además, cada vez que se abre un hilo, se mueve la barra de navegación lateral lo cual dificulta la lectura del foro. Esto contribuye a que haya alumnos que no usen de forma adecuada el foro, abriendo hilos a preguntas ya resueltas, y que éste (exceptuando la organización mínima dada mediante el uso de títulos decidida por el profesorado) esté bastante desorganizado. Todo esto provoca que el mantenimiento de este canal de comunicación propio de la plataforma sea muy costoso.

Otra lección aprendida consiste en valorar convenientemente el esfuerzo de los *contenidos y actividades adicionales*. En el caso de este curso se proporcionaban por ejemplo vídeos grabados con alumnos del Colegio Americano de Madrid como complemento. Sin embargo, la logística detrás de la grabación de esos contenidos fue costosa e involucró desplazamientos de un equipo de cámara y sonido más el profesor durante varias sesiones, el visionado de horas de grabación para resumir los aspectos más relevantes y finalmente la producción final de los vídeos.

Por último, es importante contar con una *unidad de apoyo* como la UTEID a la hora de realizar un MOOC. De esta forma el profesorado puede abstraerse de muchos de los detalles técnicos de la creación de contenidos audiovisuales, para centrarse en añadir valor y contexto a los contenidos creados. Además, una unidad como la UTEID puede realizar un seguimiento periódico del estado del curso para intentar mantener la planificación inicial, y asumir la comunicación con el proveedor de la plataforma (edX en este caso).

IV. CONCLUSIONES

Este artículo ha presentado el diseño del MOOC “Introducción a la Programación con Java – Parte 1: Empezando a Programar en Java”, incluyendo el ciclo de trabajo utilizado en la UC3M, la visión general del curso, el diseño de los vídeos y actividades, el sistema de evaluación y la componente social del curso. Este MOOC forma parte de una serie de tres módulos dedicados a introducir la programación utilizando el lenguaje Java. Para ello en este primer módulo se ha utilizado una aproximación *bottom-up*, desde la implementación más básica basada en instrucciones y programación imperativa hasta el diseño de alto nivel basado en el paradigma de orientación a objetos. Queda también pendiente reutilizar parcialmente los materiales generados en este MOOC para realizar una segunda aproximación de este primer módulo siguiendo una aproximación *top-down*, desde el diseño de alto nivel hasta la implementación de bajo nivel, para permitir a los alumnos elegir la aproximación a través de la cual desean aprender los principios de la programación utilizando el lenguaje Java, ya que no existe un consenso generalizado en la comunidad con respecto a estas dos aproximaciones.

Además, en este artículo se han recogido algunas de las lecciones aprendidas por el profesorado durante el diseño y puesta en marcha de este MOOC. Estas lecciones aprendidas se refieren: a la necesidad de hacer un uso adecuado del tiempo de dedicación al MOOC; a la necesidad de preparar una buena estrategia de coordinación para la elaboración y testeo de los contenidos generados, así como de un plan de contingencia por si se producen bajas en el equipo docente; a la necesidad de fomentar la interactividad en el curso a través del planteamiento de numerosas actividades, las cuales pueden utilizar herramientas nativas de las plataformas o herramientas de terceros (con los problemas derivados que esto puede conllevar); a la necesidad de plantear proyectos o laboratorios con una cierta continuidad a lo largo del curso; a la necesidad de gestionar de una forma organizada los canales de comunicación con los alumnos; a la necesidad de valorar el esfuerzo que supone generar recursos adicionales de apoyo al curso; y a la necesidad de buscar apoyo especializado dentro de la universidad donde se va a llevar a cabo el curso.

AGRADECIMIENTOS

Este artículo ha sido parcialmente financiado por la Comunidad de Madrid a través de la Red de Excelencia eMadrid (S2013/ICE-2715). Este trabajo también ha recibido financiación por parte del Ministerio de Economía y Competitividad a través de los proyectos EEE (TIN2011-28308-C03-01) y RESET (TIN2014-53199-C3-1-R) y las Ayudas para contratos para la Formación Posdoctoral 2013 (FPDI-2013-17411). Nos gustaría agradecer también especialmente a la Unidad de Tecnología Educativa e Innovación Docente (UTEID) de la UC3M por contribuir en el diseño y despliegue del MOOC y a edX, así como a Diego Villarán, Jonathan Almodóvar y Raquel M. Crespo por su colaboración en la realización de diferentes ejercicios del MOOC, y al Colegio Americano de Madrid y particularmente a Joaquín, Elisa, Nico, Mylo, Carlos y Eduardo por sus aportaciones en los vídeos de la visión de los estudiantes.

REFERENCIAS

- [1] D. J. Skiba, “Disruption in Higher Education: Massively Open Online Courses (MOOCs)”, *Nursing Education Perspectives*, vol. 33, n. 6, pp. 416-417, 2012.
- [2] A. McAuley, B. Stewart, G. Siemens, D. Cormier, “The MOOC Model for Digital Practice”, 2010. Último acceso mayo 2015: http://www.elearnspace.org/Articles/MOOC_Final.pdf
- [3] A. Fox, “From MOOCs to SPOCs”, *Communications of the ACM*, vol. 56, n. 12, pp. 38-40, 2013.
- [4] C. Delgado-Kloos, P. J. Muñoz-Merino, M. Muñoz-Organero, C. Alario-Hoyos, M. Pérez-Sanagustín, H. A. Parada G., J. A. Ruipérez, J. L. Sanz, “Experiences of Running MOOCs and SPOCs at UC3M”, *Proceedings of the 2014 IEEE Global Engineering Education Conference, EDUCON 2014*, 884-891, Estambul, Turquía, 2014.
- [5] P. J. Muñoz-Merino, E. Méndez Rodríguez, C. Delgado Kloos, “SPOCs for Remedial Education: Experiences at the Universidad Carlos III de Madrid”, *Proceedings of the 2nd MOOC European Stakeholders Summit, EMOOCs 2014*, 271-276, Lausana, Suiza, 2014.
- [6] edX. Último acceso mayo 2015: <https://www.edx.org>
- [7] MiríadaX. Último acceso mayo 2015: <https://www.miriadax.net>
- [8] C. Alario-Hoyos, M. Pérez-Sanagustín, C. Delgado-Kloos, H. A. Parada G., M. Muñoz-Organero, “Delving into participants’ profiles and use of social tools in MOOCs”, *IEEE Transactions on Learning Technologies*, vol. 7, n. 3, pp. 260-266, 2014.
- [9] UTEID, Unidad de Tecnología Educativa e Innovación Docente. Último acceso mayo 2015: <http://www.uc3m.es/uteid>
- [10] UTEID, “Guía de buenas prácticas para profesores para la creación de vídeos educativos”, UC3M, 2014. Último acceso mayo 2015: http://portal.uc3m.es/portal/page/portal/biblioteca/UTEID/UTEID_GuiaCreacionVideosEdu-v.5.pdf
- [11] UTEID, “Guía metodológica para la planificación, diseño e impartición de MOOCs y SPOCs”, UC3M, 2014. Último acceso mayo 2015: http://portal.uc3m.es/portal/page/portal/biblioteca/UTEID/UTEID_GuiaMetodologicaMOOC%20v3Octubre2014.pdf
- [12] C. Alario-Hoyos, M. Pérez-Sanagustín, C. Delgado-Kloos, I. Gutiérrez-Rojas, D. Leony, H. A. Parada G., “Designing your first MOOC from scratch: recommendations after teaching ‘Digital Education of the Future’”, *Proceedings of the 2nd MOOC European Stakeholders Summit, EMOOCs 2014*, 172-177, Lausana, Suiza, 2014.
- [13] C. Alario-Hoyos, M. Pérez-Sanagustín, C. Delgado-Kloos, P. J. Muñoz-Merino, “Recommendations for the design and deployment of MOOCs: Insights about the MOOC Digital Education of the Future deployed in MiríadaX”, *Proceedings of the 2nd International Conference on Technological Ecosystems for Enhancing Multiculturality, TEEM 2014*, 403-408, Salamanca, España, 2014.
- [14] Introduction to Programming with Java - Part 1: Starting to Program in Java. Último acceso mayo 2015: <https://www.edx.org/course/introduction-programming-java-part-1-uc3mx-it-1-1x>
- [15] Studio Edge (edX). Último acceso mayo 2015: <https://studio.edge.edx.org>
- [16] J. McGonigal, “Reality is Broken: Why Games Make us Better and How they Can Change the World”, United States: Penguin Group, 2011.
- [17] C. Fernández-Panadero, C. Delgado Kloos, “PhyMEL. A Framework to Integrate Physical, Mental and Emotional Learning in Meaningful Experiences and Multidimensional Reports”, *3rd European Immersive Education Summit*, 203-209, Londres, Reino Unido, 2013.
- [18] P. Guo, “Optimal Video Length for Student Engagement”, edX, 2013. Último acceso mayo 2015: <https://www.edx.org/blog/optimal-video-length-student-engagement>
- [19] IMS Global Learning Consortium. Learning Tools Interoperability. Último acceso mayo 2015: <http://www.imsglobal.org/lti/index.html>
- [20] Blockly. Último acceso mayo 2015: <https://developers.google.com/blockly/>
- [21] Codeboard. Último acceso mayo 2015: <https://codeboard.io/>
- [22] Greenfoot. Último acceso mayo 2015: <http://www.greenfoot.org/>