



LOS RETOS DE LLEVAR UN CURSO DE PROGRAMACIÓN EN UNA FACULTAD DE INGENIERÍA A UNA PLATAFORMA PARA MOOCS

Mario Sánchez, Andrea Herrera, Andrés Melani, Marcela Hernández Hoyos

**Universidad de los Andes
Bogotá, Colombia**

Resumen

Indiscutiblemente la transformación más grande de la educación en Ingeniería en los últimos 10 años ha sido la virtualización del aula de clase primero debido a la aparición y auge de los MOOCs (Massive Open Online Courses) basados en plataformas de aprendizaje masivas como Coursera y edX, y luego por la pandemia ocasionada por el COVID-19. Sin embargo, incluso antes de que ocurriera la pandemia muchas voces apuntaban a que el futuro de la educación, en Ingeniería y en prácticamente todos los campos, iba a estar basada en tecnología de información y comunicaciones (TIC). Las señales sugerían que era inevitable la virtualización del aula, entendida como la desaparición definitiva del modelo en que profesor y estudiantes están siempre ubicados en el mismo espacio físico y la adopción de TICs para mediar en todas sus interacciones. Dentro de este escenario, que amenaza directamente la supervivencia de las Instituciones de Educación Superior (IES), empezó una transformación que no ha sido fácil y que tomará muchos años lograr. Detrás de esa aparente lentitud hay varios motivos, pero frecuentemente se atribuye a falta de recursos (dinero, tiempo, recursos tecnológicos, etc.). Sin embargo, un motivo posiblemente más importante es que la educación virtual requiere transformaciones profundas en la enseñanza (del lado de los profesores) y del aprendizaje (del lado de los estudiantes): transformar un curso tradicional en un curso virtual necesita una reflexión profunda sobre lo que se quiere lograr, lo que realmente se puede lograr y los medios disponibles para alcanzar esos objetivos. Convertir un curso de una IES en su versión virtual no es algo que se pueda hacer rápidamente ni siguiendo siempre el mismo proceso.

Recientemente los autores de esta ponencia participaron en el proyecto para convertir un curso existente de una Facultad de Ingeniería en el curso equivalente para una plataforma de aprendizaje

masivo (Coursera). Este proyecto incluyó el análisis y ajuste de los objetivos de aprendizaje, la revisión y rediseño de todas las actividades de aprendizaje y de evaluación, la producción del material de instrucción, y la adaptación e implementación de herramientas para evaluar automáticamente una parte del trabajo de los estudiantes. Como resultado de todo este esfuerzo, además del curso que se dicta presencialmente a casi 1000 estudiantes por semestre, ahora hay un curso virtual en el que más de 3500 estudiantes de todo el mundo se han inscrito durante los primeros cinco meses al aire.

Esta ponencia presenta experiencias vividas en el proyecto y se concentra en los aspectos más retadores para construir un curso virtual, así se tenga un curso presencial como punto de partida. Se espera que estas experiencias sirvan como guía a quienes estén pensando en hacer lo mismo con otros cursos en Ingeniería para que dimensionen mejor los esfuerzos que van a requerir y para que sepan a qué retos podrían enfrentarse durante el proceso.

Palabras clave: MOOC; educación en ingeniería; enseñanza de la programación

Abstract

The virtualization of engineering courses, first by going remote by choice, then by the introduction of MOOCs (Massive Open Online Courses), and recently by going remote by force due to the COVID-19 pandemic, is undoubtedly a game changer for engineering education. Even before the COVID-pandemic, the virtualization of classrooms seemed inevitable and threatened the survival of higher education institutions. Although these institutions have begun to adapt, it has been a slow process. On the one hand, there is a chronic lack of resources (money, time, tools, etc.). More importantly, however, the migration to virtual delivery mechanisms for instruction requires a deep reflection on what the ultimate goals of instruction are and what can be achieved. Even starting with mature offline courses, creating virtual versions is not easy and is not a repeatable process.

The authors of this paper were part of the project that brought an existing CS1 course to a MOOC platform (Coursera): the project included the analysis and adjustment of the course objectives, the review and redesign of all learning and assessment activities, and the implementation and adaptation of tools to automatically grade students' work. The result is a new MOOC with more than 3,500 students enrolled in its first five months, in addition to the existing offline course, which has more than 1,000 students enrolled per semester.

This paper presents experiences faced by the project and focuses on the most challenging aspects of building a MOOC, even when using a complete face-to-face course as starting point. These experiences should serve as a guide to all those who are thinking of carrying out similar projects with other existing engineering courses so that they can better dimension the efforts they will require and know the challenges they may face during the process.

Keywords: MOOC; engineering education; teaching programming



1. Introducción

En años recientes, las Instituciones de Educación Superior (IES) han hecho grandes esfuerzos para empezar a impartir cursos remotos a través de plataformas electrónicas y así llegar a poblaciones más amplias y dispersas. Inicialmente, estos cursos estaban dirigidos a grupos relativamente pequeños, compuestos exclusivamente por estudiantes matriculados en la institución. Sin embargo, con la aparición en 2012 de plataformas como Coursera y edX, las IES empezaron también a construir MOOCs: cursos abiertos masivos en-línea (Massive Open Online Courses). Estos dos esquemas tienen algunas cosas en común, pero también tienen importantes diferencias. Una de ellas tiene que ver con el perfil y comportamiento de los estudiantes (Hennis et al., 2016): en el primer caso, los estudiantes tienen características similares y hay un cierto control sobre lo que deben ser capaces de hacer antes de inscribirse a cada curso, mientras que en los MOOCs no se puede asumir mucho con respecto a los conocimientos previos que tengan y hay mucha más variedad en la edad, ubicación geográfica, intereses, motivaciones y formación académica. Por otro lado, en los cursos cerrados (remotos) los grupos de estudiantes son relativamente pequeños y siguen un cronograma bien definido, que suele incluir varias actividades síncronas. De esta manera los instructores pueden ir monitoreando permanentemente el progreso de los estudiantes, ajustando elementos sobre la marcha, y complementando cosas que no sean totalmente claras dentro del material de instrucción. Por el contrario, en los MOOCs los grupos de estudiantes son mucho más grandes y cada uno puede llevar su propio ritmo. Esto hace que no sea viable incluir actividades síncronas y limita enormemente el tipo de actividades que se pueden realizar: por ejemplo, las sesiones prácticas de un curso presencial podrían mantenerse para un curso remoto, pero tienen que transformarse radicalmente para usarse en un MOOC porque la cantidad de estudiantes desborda la capacidad de cualquier equipo dentro de una IES (profesores, monitores, asistentes, etc.).

Durante el 2020, un equipo de profesores del Departamento de Ingeniería de Sistemas y Computación de la Universidad de los Andes tuvo que transformar un curso presencial en un curso remoto, debido a la pandemia ocasionada por el COVID-19, y paralelamente también vivió la experiencia de adaptar ese mismo curso a una plataforma para MOOCs (Coursera). Estos dos procesos dejaron muchos aprendizajes que podrían ser de utilidad en iniciativas similares, especialmente en proyectos en los que cursos presenciales existentes se lleven a plataformas para educación masiva.

El resto del artículo está organizado de la siguiente manera. La sección 2 presenta brevemente el curso de "Introducción a la Programación" haciendo énfasis en los aspectos que lo diferencian de cursos equivalentes en otras universidades y en los ajustes que fueron necesarios para impartirlo de forma remota. Luego, la sección 3 presenta la parte central de la ponencia, es decir los principales aprendizajes que dejó el proyecto de creación del MOOC, organizados en 6 aspectos. Finalmente, la sección 4 concluye el artículo y resume las principales recomendaciones para quienes se enfrenten a proyectos similares.



2. Introducción a la Programación: presencial y remoto

CS1 es el término que se utiliza en todo el mundo para referirse de forma genérica al primer curso de programación dentro de una IES, independientemente de que sea dictado por un departamento de ciencias de la computación (CS) o ingeniería. Sin embargo, no existe ni una estructura ni un conjunto de contenido que pueda considerarse verdaderamente estándar: cada curso tiene sus propias características que dependen de factores como la estructura de los programas académicos de la institución, las tecnologías disponibles, y asuntos históricos a nivel de la universidad y del país (Pears et al., 2007). En el caso de la Universidad de los Andes, la última reforma al curso CS1 se había realizado en el 2004 y el resultado había sido un curso de programación orientada a objetos con Java (APO1), basado en técnicas de aprendizaje activo y en el uso de casos (Villalobos et al., 2008). En 2018 inició un proyecto para reformar este curso, adaptándolo a las nuevas necesidades de los programas de las facultades de Ingeniería y Ciencias, y actualizando las tecnologías empleadas. El resultado de este proceso fue un curso totalmente renovado llamado Introducción a la Programación (IP).

Las principales características que distinguen a IP del curso anterior, y de cursos equivalentes en otras IES, son: (i) el foco en el desarrollo de habilidades algorítmicas, (ii) la descomposición de problemas, y (iii) el uso del lenguaje de programación Python sin emplear el paradigma orientado a objetos. Metodológicamente, IP conserva muchos elementos de APO1: utiliza técnicas de aprendizaje activo, está basado en 4 niveles incrementales, y el método de evaluación es prácticamente idéntico al de APO1 (un proyecto, una evaluación práctica y una evaluación teórica por nivel). Adicionalmente, para el curso de IP se construyó material de referencia a la medida (presentaciones, notas de clase y una plataforma de calificación automática de ejercicios de programación - Senecode) que permitió garantizar la homogeneidad entre las múltiples secciones que se dictan en cada semestre (primero 2 secciones, luego 6 secciones y luego más de 40 secciones de 25 estudiantes a partir del primer semestre de 2020).

Precisamente a mitad del primer semestre masivo de IP se declaró la pandemia del COVID-19 y el curso tuvo que pasar de ser presencial a ser remoto. Contrario a lo que se esperaría, no fue necesario hacer mayores ajustes al curso. En primer lugar, se reemplazaron las sesiones síncronas presenciales por sesiones síncronas remotas a través de plataformas como Zoom, Webex, Teams y Blackboard Collaborate. Cada profesor se encargó de solucionar los aspectos técnicos, con el apoyo de la Universidad, y la coordinación del curso se encargó de supervisar que los ajustes de índole académica se implementaran de manera uniforme entre todas las secciones.

Estos cambios académicos no requirieron demasiado esfuerzo y fueron relativamente sencillos de implementar. El monitoreo realizado al curso muestra que, en general, se están obteniendo los resultados que se esperan del mismo y que el material que se construyó para la versión presencial ha sido igualmente útil para las clases remotas.



3. El MOOC – Programación en Python

En 2019, la Universidad de los Andes inició un proyecto para la virtualización de programas de maestría, dentro del cual surgió la necesidad de producir un curso que sirviera para que personas que nunca hubieran programado desarrollaran habilidades básicas de programación. En ese momento se decidió que el nuevo curso sería un MOOC sobre la plataforma Coursera y que el curso de IP sería la base para su construcción. El primer motivo para esto es que los objetivos del curso de IP parecían coincidir con lo que requería el MOOC, el segundo motivo es que el material del curso presencial parecía ser justo lo que necesitaba el MOOC y el tercer motivo es que el equipo de trabajo ya tenía experiencia en la construcción de un MOOC a partir del curso APO1 (Hernández et al., 2017).

A continuación, se discutirán 5 aspectos relacionados con el desarrollo del MOOC, y uno relacionado con su operación, que agrupan las principales lecciones aprendidas durante el proyecto. También se explica el esfuerzo real que se requirió para montar y operar la versión MOOC del curso de IP, que tiene el nombre Programación en Python (PP).

3.1 Objetivos de aprendizaje

Al igual que en cursos tradicionales, el diseño de un curso en versión MOOC debería partir de la definición de los objetivos de aprendizaje, aún cuando en muchos casos esto no ocurra (Margaryan et al., 2015). En el caso del curso PP, la base fueron los objetivos del curso de IP con algunos ajustes, debido a que el alcance del curso virtual está definido por los requerimientos del programa de maestría. Como resultado, los objetivos del curso PP son más amplios que los del curso de IP. Se trata entonces de un curso bastante ambicioso que requiere un esfuerzo comparativamente alto por parte de los estudiantes: el estimado conservador es que un estudiante requiere 60 horas de trabajo total, pero podría ser más dependiendo de su formación.

Una segunda razón para ajustar los objetivos de aprendizaje es que en un MOOC es difícil alcanzar exactamente los mismos objetivos que en un curso presencial. Esto puede variar entre disciplinas, cursos y niveles de complejidad, pero la dificultad para realizar actividades en grupo, el tipo de interacción que puede darse entre estudiantes, los tipos de evaluaciones, y la falta de retroalimentación directa y personalizada por parte de los instructores, hace necesario que los objetivos sean analizados con cuidado. En particular, es necesario revisar los objetivos relacionados con los niveles más avanzados de la taxonomía de Bloom (analizar, evaluar, crear) (Anderson et al., 2001).

Por último, los objetivos donde la retroalimentación sea fundamental para el proceso de aprendizaje también deben revisarse, así como los objetivos relacionados con la realización de trabajos en grupo: aunque plataformas como Coursera ofrecen mecanismos para la interacción entre estudiantes, el hecho de que cada uno pueda seguir el curso a su propio ritmo y tenga sus propios intereses y motivaciones, lleva a situaciones frustrantes en las que un estudiante puede quedar bloqueado porque ninguno de sus “compañeros” está dispuesto a trabajar en una ventana de tiempo específica. Vale la pena resaltar que antes del año 2015 muchos MOOCs tenían fechas de inicio y calendarios estrictos, lo cual permitía formar “cohortes” de estudiantes que podían tener



un nivel de interacción mucho más alto con los instructores y entre ellos que lo que pueden tener ahora en los llamados “self-paced courses” (cada estudiante elige cuando iniciar el curso y es libre de definir su propio ritmo) (Shah, 2015).

3.2 Actividades de aprendizaje

Una vez ajustados los objetivos de aprendizaje es fundamental revisar las actividades del curso para asegurar que sean adecuadas para alcanzar dichos objetivos y que sean compatibles con la plataforma del MOOC. Durante la creación del curso PP, se identificaron dos tipos de actividades: individuales guiadas e individuales independientes.

Las actividades individuales guiadas son las que un estudiante realiza por su cuenta, pero en las que los instructores tienen un fuerte nivel de control y pueden estimar con precisión el esfuerzo que van a requerir. Estas actividades incluyen revisar videos, hacer lecturas y resolver evaluaciones formativas pequeñas. Estas actividades remplazan la mayoría de las actividades que se realizarían de forma síncrona en un curso tradicional. En el caso de la creación del curso PP, la preparación del material para estas actividades requirió grandes esfuerzos que se describirán en la sección 3.3.

Por otro lado, las actividades individuales independientes son aquellas en las que los instructores escriben un enunciado y luego tienen muy poco control sobre lo que haga un estudiante o cuánto se demore. En el caso del curso PP, esto corresponde a la realización de 4 proyectos (uno por nivel) y la resolución de múltiples problemas de programación con calificación automática (ver secciones 3.3 y 3.5). Siguiendo buenas prácticas para el diseño de cursos que buscan el desarrollo de habilidades, el curso PP incluye una gran cantidad de ejercicios de programación, los cuales permiten poner en práctica los temas cubiertos en los videos y lecturas.

Finalmente, la experiencia previa con la creación de un MOOC basado en APO1 hizo que no se consideraran actividades grupales y que las actividades de revisión entre pares se limitaran al máximo: como ya se mencionó, estas actividades llevan a situaciones desafortunadas donde el proceso de un estudiante queda bloqueado por la falta de colaboración de sus “compañeros”. El curso PP sólo incluye una de estas actividades por nivel (revisión del proyecto), pero configurada para que no haga parte de la calificación y no sea una actividad requerida para avanzar y terminar el curso.

3.3 Recursos de instrucción

Los aspectos anteriores se concentraron en el diseño del curso. Esta sección describe la construcción propiamente dicha de los recursos de instrucción necesarios. En el caso del curso PP, estos recursos incluyen lecturas, presentaciones, videos y enunciados de ejercicios. Las características genéricas que este material debería cumplir, incluso para cursos presenciales y remotos, pero que en el caso de los MOOCs se vuelven mucho más importantes debido a la mayor cantidad y diversidad de estudiantes, y a la falta de interacción directa con los instructores, son las siguientes:

- Alta calidad y claridad: es indispensable que el material sea revisado con mucho cuidado y sea muy claro porque los estudiantes no tienen la oportunidad de solicitar aclaraciones.



- Auto-contenido: en lo posible, cada elemento debe incluir toda la información necesaria para entender un tema, resolver una tarea, etc. Aunque pueden haber referencias explícitas entre documentos, esto puede llevar a inconsistencias si algún documento o la estructura del curso cambia en el futuro.
- Estabilidad: dado que el esfuerzo de elaboración del material es alto, es importante garantizar que sea relevante durante un tiempo relativamente largo. Para eso se recomienda evitar: mencionar fechas, hacer referencia a eventos particulares y depender de aspectos tecnológicos más específicos de los estrictamente necesarios.
- Neutralidad: en el caso particular de cursos en español, se sugiere usar un lenguaje y una terminología que sea lo más neutral posible. En el caso del curso PP, se usaron las versiones en inglés de términos técnicos como estrategia para evitar traducciones al español que sean locales.
- Consistencia estética: para facilitar la comprensión del material, se recomienda apelar también a asuntos estéticos de forma consistente. En el caso del curso PP, el material se construyó siguiendo la paleta de colores y las bibliotecas de fuentes que se usan en el curso de IP, lo cual permite ubicar cualquier artefacto rápidamente dentro del curso.
- Accesibilidad: aunque para el curso PP no era obligatorio, Coursera y otras plataformas están muy interesadas en que todo el material tenga características que le faciliten a cualquiera participar en el curso, independientemente de sus capacidades cognitivas o físicas. Esto es particularmente relevante para la selección de las paletas de colores para las presentaciones, enunciados y capturas de pantalla, pero también tiene implicaciones sobre la organización de las diapositivas y la inclusión de meta-información para todas las imágenes que se utilicen.

Con respecto a los videos propiamente dichos, durante el desarrollo del curso PP se siguieron las recomendaciones que propone Coursera (Hickey et al., 2020). En particular, los videos del curso tienden a tener una duración cercana a los 7 minutos y sólo se extienden para los videos llamados "Manos a la obra", en los que se desarrollan ejemplos prácticos acompañados de explicaciones detalladas. En la mayoría de los videos se habla de un tema puntual que se expone y se ilustra con uno o más ejemplos. Un gran porcentaje de los videos incluyen también "preguntas en video", las cuales se presentan al estudiante en un punto predeterminado y lo obligan a responder una pregunta – usualmente sencilla - antes de seguir adelante.

La preparación de los videos fue la parte más demandante del proceso. Para cada uno es necesario preparar una presentación, diseñar las preguntas en video (enunciado, respuestas y explicaciones) y el punto donde se van a introducir, escribir un guion para el instructor, y grabar el video. En el curso PP, en la mayoría de los videos no se puede ver la cara del instructor, lo cual facilita enormemente la producción. Además, los instructores mismos se encargaron de aprender a utilizar herramientas de grabación y edición sencillas pero eficaces, lo cual redujo enormemente los costos de producción del material y permitió hacer las grabaciones en las casas de cada uno. Finalmente, se utilizó como política conservar los archivos fuentes de todo el material (presentaciones, guiones, audios, proyectos del software de edición), lo cual permitió hacer correcciones pequeñas sobre algunos videos sin tener que volver a grabarlos completamente.

Por último, el material también incluyó guías técnicas para ayudar a los estudiantes a instalar en sus computadores el software que el curso requiere. Para la versión presencial este material también



existe y disminuye los problemas de los estudiantes. Para el curso PP también ha resultado de utilidad, aunque con la cantidad de estudiantes también ha aumentado la variedad de problemas que se presentan durante los procesos de instalación. El lado negativo de estas guías es que se desactualizan con frecuencia, cada vez que aparece una nueva versión de las herramientas.

3.4 Evaluaciones

Aunque la elaboración de los recursos de instrucción fue la parte más dispendiosa del proceso, el diseño del sistema de evaluación y la construcción de las evaluaciones propiamente dichas fue la experiencia más retadora. Es importante considerar que no todo lo que se evalúa en un curso presencial puede evaluarse en un MOOC: si se tiene en cuenta una vez más la cantidad de estudiantes que podría tener un MOOC, la única salida posible es que la evaluación dependa enteramente de mecanismos automáticos.

En el caso del curso PP, la evaluación se basa en 4 exámenes, uno por nivel, y cada uno tiene una combinación de preguntas de selección múltiple y problemas de programación. Las preguntas de selección múltiple (con una respuesta o con múltiple respuesta) son un mecanismo relativamente sencillo para preparar una evaluación que se califique automáticamente. Sin embargo, no es fácil construir una buena evaluación con este tipo de preguntas para cualquier tema y mucho menos para evaluar el desarrollo de una habilidad. En el caso del curso PP, este tipo de preguntas se utilizaron en los exámenes para validar el conocimiento de temas puntuales o a través de preguntas relativamente complejas en las cuales las opciones posibles son cada una un programa.

Por otra parte, las evaluaciones también incluyen problemas de programación, que se evalúan de forma automática igual que las tareas (ver sección 3.5). Aunque el mecanismo es muy efectivo porque permite certificar la capacidad del estudiante para resolver cierto tipo de problemas de programación, tiene el inconveniente de ser drástico: si el programa que escribe el estudiante no está perfecto, será rechazado. Además, aunque se hizo un esfuerzo importante por construir un sistema capaz de dar retroalimentación inteligente, es importante tener presente que este mecanismo nunca será equivalente a la retroalimentación que daría un profesor y le permitiría al estudiante corregir sus errores.

Con respecto a los exámenes también vale la pena agregar que la cuidadosa redacción y verificación de los ejercicios es de suma importancia: los estudiantes ponen mucha atención a los enunciados de los exámenes y está casi garantizado que cualquier frase que sea susceptible de ser malinterpretada, será malinterpretada. Eso es particularmente crítico para los ejercicios de programación, donde la interpretación equivocada de un detalle puede llevar a un programa que nunca será considerado correcto por el calificador automático.

El otro aspecto que hace parte de la evaluación del curso PP son los proyectos. El problema es que la evaluación automática sólo llega a evaluar criterios "fáciles" y no puede evaluar procesos cognitivos más complejos. Por ejemplo, la evaluación automática verifica que el estudiante haya implementado funciones con nombres específicos y que esas funciones produzcan los resultados correctos, pero no puede revisar que la implementación no sea ineficiente, use malas prácticas de programación, o apele a una mala descomposición de los problemas. Revisiones más



“inteligentes”, requerirían de la aplicación de técnicas más complejas que son actualmente tema de investigación. Por este motivo, la revisión de los proyectos se dejó en manos de los pares y se excluyó del sistema de calificación del curso.

3.5 Plataforma

Conocer de antemano la plataforma donde será desplegado el MOOC es muy importante: aunque las grandes funcionalidades son las mismas, hay detalles pequeños que terminan generando diferencias importantes. Por ejemplo, en el caso del curso PP, el diseño de las preguntas en video y de los exámenes se basó en los tipos de preguntas soportados por Coursera. De igual forma, inicialmente no se iban a realizar prácticas de programación para la parte del curso que trata de análisis de datos, pero una actualización de la plataforma agregó el soporte para las librerías que nos hacían falta.

El mecanismo utilizado para automatizar la evaluación de problemas de programación merece una explicación aparte. En primer lugar, la forma en la que se evalúan las soluciones a problemas de programación enviadas por un estudiante se basa en el mecanismo que utiliza la plataforma Senecode que se desarrolló específicamente para el curso IP (Sánchez, et al., 2020). Durante la etapa inicial del diseño del curso PP, se consideró la posibilidad de integrar a Senecode, como una herramienta externa, para hacer la evaluación de estas soluciones. Sin embargo, la experiencia previa con el MOOC creado a partir de APO1 llevó a la conclusión de que esta era una mala idea: si actividades dentro de un MOOC dependen de sistemas externos, los sistemas externos deben garantizar una disponibilidad similar a la de la plataforma del MOOC. Es decir, que tienen que funcionar 24 horas al día, durante todo el año.

Al descartar la opción de integrar una herramienta externa, fue necesario desarrollar un componente para que se ejecutara dentro de Coursera y realizara la evaluación. Aunque podría haberse utilizado la tecnología conocida como “contenedores”, se optó por la solución más sencilla posible y más resistente a cambios tecnológicos. Es decir, se optó por depender únicamente del lenguaje de programación que soporta la plataforma (Python) y no utilizar ninguna librería adicional.

Finalmente, es importante entender y tener en cuenta que, aunque podría pensarse que son plataformas maduras, todas las plataformas para MOOCS están en permanente evolución. Eso puede llevar a que presenten errores esporádicamente y será más evidente en la medida en que se utilicen características avanzadas.

3.6 Operación y actualización

Como último aspecto para considerar está la operación y actualización del MOOC. Una vez el curso PP salió al aire, se volvió necesario monitorear permanentemente los foros de discusión a través de los cuales los estudiantes intentan comunicarse con los instructores o con otros compañeros para pedir aclaraciones sobre algunos puntos particulares, o a través de los cuales reportan problemas que hayan encontrado en el material. Mantener la vigilancia diaria sobre estos foros es muy importante para garantizar la calidad del curso.



Además, una vez el curso está al aire es necesario monitorear que todo funcione correctamente. Actualizaciones sobre la plataforma podrían hacer que ciertas características dejen de funcionar. Por ejemplo, así como una actualización sobre la plataforma de Coursera habilitó la ejecución de programas que hicieran análisis de datos, otra actualización generó un pequeño problema en el reporte de los errores del examen de un estudiante.

4. Conclusiones

A partir de las experiencias presentadas en este artículo, es posible extraer importantes conclusiones que esperamos sean de gran utilidad para proyectos que intenten preparar cursos remotos y MOOCS a partir de cursos existentes. La primera conclusión es que todo el proceso es más sencillo si se parte de un curso con un diseño explícito y consistente, es decir un curso donde los objetivos de aprendizaje estén declarados y sea posible establecer un mapeo preciso entre esos objetivos y todas las actividades de aprendizaje y evaluación.

La segunda conclusión es que no hay mucha diferencia, desde el punto de vista del diseño del curso y del material, entre un curso presencial y uno remoto – para cursos con características similares al del caso de estudio. Aunque es claro que las actividades donde hay contacto entre instructores y estudiantes deben replantearse, la estructura general del curso, los objetivos de aprendizaje y el material de soporte no deberían sufrir grandes cambios.

Por otro lado, los aspectos que tienen un impacto muy grande en el rediseño de los cursos tienen que ver con la cantidad de estudiantes y la desaparición de las actividades síncronas. Estos dos puntos son absolutamente centrales para la migración hacia un MOOC y son la base para las siguientes recomendaciones que resumen el artículo:

1. Revise y ajuste los objetivos de aprendizaje: probablemente no serán los mismos en el MOOC y en el curso presencial.
2. Acomode las actividades a los objetivos y sólo conserve las actividades que tengan sentido dentro de la plataforma, procurando que sean individuales. Incluya actividades para poner en práctica todos los temas posibles y complementar las lecturas y videos.
3. Haga una planeación cuidadosa de las actividades para construir lecturas, videos, presentaciones y enunciados, incluyendo actividades de aseguramiento de la calidad.
4. Replantee los mecanismos de evaluación: conserve sólo lo que se pueda automatizar y diseñe nuevos mecanismos para evaluar lo que no quede cubierto. Tenga en cuenta que tal vez no podrá evaluar lo mismo que en el curso presencial.
5. Conozca bien la plataforma para aprovecharla al máximo: tal vez tenga funcionalidades que faciliten su trabajo y le dejen evaluar cosas que parecían imposibles.
6. Desde el inicio del proyecto planee cómo realizará la operación del curso y establezca el horizonte de tiempo en que se dará soporte al curso.

Finalmente, este artículo ha cubierto sólo una parte de los aspectos que se deben tener en cuenta para desarrollar un MOOC, y se ha concentrado en aspectos relacionados con la migración de



cursos existentes. Hay muchos más elementos que deberían tenerse en cuenta, como por ejemplo otros mecanismos de evaluación, diferencias entre disciplinas, y técnicas y recomendaciones más avanzadas para la preparación del material.

5. Referencias

- Anderson L., Krathwohl D., Airasian P., Cruikshank K., Mayer R., Pintrich P., Raths J., Wittrock M. (2001). A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives., Longman, New York.
- Hennis, T., Topolovec, S., Poquet, O., & de Vries, P. (2016). Who is the Learner: Profiling the Engineering MOOC Student. In SEFI 44th Annual Conference, Tampere, Finland.
- Hernández M., Reyes J., Gómez A. (2017). Introducción a la programación orientada a objetos en Java. <https://www.coursera.org/learn/introduccion-programacion-java/>
- Hickey A., Urban A., Karsten E. (2020). Drivers of Quality in Online Learning, Coursera, pp. 37.
- Margaryan A., Bianco M., Littlejohn A. (2015). Instructional quality of Massive Open Online Courses (MOOCs). Computers & Education, Vol. 80, 2015, pp. 77-83.
- Pears A., Seidman S., Malmi L., Mannila L., Adams E., Bennedsen J., Devlin M., and Paterson J. (2007). A survey of literature on the teaching of introductory programming. SIGCSE Bull. 39, No. 4, pp. 204–223.
- Sánchez M. and Salazar P. (2020). A feedback-oriented platform for deliberate programming practice. Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education, pp. 531- 532.
- Shah, D (2015). MOOC Trends in 2015: Rise of Self Paced Courses. Class Central.
- Villalobos J. and Vela M. (2008) CUIP2-An Active Learning and Problem Based Learning Approach to Teaching Programming. Proceedings of ALE'08 Workshop, Bogota, Colombia.

Sobre los autores

- **Mario Sánchez:** Ingeniero de Sistemas y Computación, Magíster en Ingeniería de Sistemas y Computación, Ph.D. de Vrije Universiteit Brussel, Bélgica. Profesor Asociado. mar-san1@uniandes.edu.co
- **Andrea Herrera:** Ingeniera de Sistemas y Computación, Magíster en Ingeniería de Sistemas y Computación, Ph.D. de University of Auckland, New Zealand. Profesora Asistente. a-herrer@uniandes.edu.co
- **Andrés Melani:** Ingeniero de Sistemas y Computación, Magíster en Tecnologías de Información para el Negocio. Instructor. af.melani3365@uniandes.edu.co
- **Marcela Hernández:** Ingeniera de Sistemas y Computación, Magíster en Ingeniería de Sistemas y Computación, Ph.D. de Institut National des Sciences Appliquées de Lyon, Francia. Profesora Asociada. marc-her@uniandes.edu.co



Los puntos de vista expresados en este artículo no reflejan necesariamente la opinión de la Asociación Colombiana de Facultades de Ingeniería.

Copyright © 2021 Asociación Colombiana de Facultades de Ingeniería (ACOFI)

