

La evaluación formativa como factor decisivo en el aprendizaje online. Intervención en una asignatura inicial de programación. Formative assessment as a key factor in online learning. Intervention in a CS1 course.

Maria-Jesús Marco-Galindo, Julià Minguillón Alonso
mmarcog@uoc.edu, jminguillona@uoc.edu

Estudis d'Informàtica Multimèdia i Telecomunicació
Universitat Oberta de Catalunya
Barcelona, España

Resumen- Un factor clave en el aprendizaje online es un diseño instruccional que garantice que el estudiantado mantenga un ritmo de aprendizaje adecuado y constante durante todo el curso. Esto es especialmente relevante cuando se requiere un aprendizaje fundamentalmente práctico y progresivo como es el caso de las asignaturas iniciales de programación. En este artículo se describe una intervención realizada en una asignatura de primer curso del grado de Ingeniería Informática llamada "Fundamentos de programación". La enseñanza de programación plantea muchos retos relacionados con la introducción de conceptos abstractos, la realización de ejercicios de programación en un lenguaje concreto y el seguimiento del ritmo de actividades de aprendizaje propuestas, de forma que los estudiantes sean capaces de realizar un aprendizaje progresivo y adecuado. A partir de un análisis de los resultados académicos de los estudiantes durante varios semestres, se decidió realizar una intervención para cambiar el ritmo de aprendizaje y el retorno recibido consiguiendo que más estudiantes sigan el modelo de actividades propuesto y superen la asignatura satisfactoriamente.

Palabras clave: *aprendizaje online, curso inicial de programación, evaluación formativa, ritmo de aprendizaje, retorno.*

Abstract- One key factor in online learning is an instructional design that guarantees that the student body maintains an adequate and constant learning rhythm throughout the course. This is especially relevant when fundamentally practical and progressive learning is required, as is the case with initial programming subjects. This article describes an intervention carried out in a subject in the first year of the Computer Engineering degree called "Fundamentals of programming". The teaching of programming poses many challenges related to the introduction of abstract concepts, the realization of practical exercises and the rhythm of the proposed learning activities, so that students can adequately follow the subject. Based on an analysis of the academic results of the students, it was decided to carry out an intervention to change the pace of student learning and the return received, thus allowing more students to continue with the proposed activity model and pass the course satisfactorily.

Keywords: *online learning CS1, formative assessment, learning pace, feedback.*

La experiencia demuestra que aprender a programar es difícil y que la enseñanza de la programación es un reto, tal y como demuestra el elevado número de publicaciones sobre el tema, como se describe por ejemplo en (Luxton et al., 2018) donde se analizan 1.666 trabajos sobre el aprendizaje de la programación. Así, la literatura refleja un acuerdo generalizado sobre el hecho de que aprender a programar es un proceso difícil para la mayoría de los estudiantes a todos los niveles. En particular, el abandono en cursos iniciales de programación en estudios universitarios generalmente es alto y los índices de superación, bajos (Watson et al., 2014). Y, a pesar de la extensa literatura sobre el tema y de décadas de experiencia e investigación, muchas preguntas siguen aún abiertas. ¿Por qué es tan difícil aprender a programar para muchos estudiantes?, ¿Qué factores permiten determinar el éxito de un estudiante en un curso inicial de programación? Parece ser que no hay ningún factor determinante que permita predecir el éxito (Carter et al., 2017), pero sí que está claro que uno de los elementos que más inciden positivamente en la eficacia del aprendizaje son las actividades prácticas de programación. Se vislumbra una necesaria e imprescindible combinación entre conocimientos básicos (saber) y prácticos (saber aplicar). Los aspectos abstractos de la algorítmica deben ponerse en práctica con la codificación en un lenguaje concreto de programación a través de una buena estrategia de actividades de laboratorio que permita a los estudiantes ir trabajándolos, primero uno a uno para después ir integrándolos progresivamente en un proyecto de mayor envergadura (Watson et al., 2014).

Desde la perspectiva del estudiante, muchos estudios analizan qué contenidos resultan más difíciles y la carga cognitiva que conllevan (Sorva, 2012), así como las estrategias de enseñanza-aprendizaje más adecuadas para reducirla (Hoda et al., 2014), entre las que se incluye el retorno individualizado, que es importante que se reciba en los momentos cruciales donde puede ser más efectivo (Ott et al., 2011). Uno de estos momentos críticos son las primeras semanas del curso en las que hay que estar muy atento para que la experiencia inicial del estudiante sea positiva, facilitando su aprendizaje y prestando ayuda rápidamente a quien muestre signos de abandono si, por ejemplo, no entrega la primera actividad o no participa en el aula (Porter et al., 2014). Relacionado con esto último, es

1. INTRODUCCIÓN

también importante estudiar los factores que inciden favorablemente en que el estudiante se involucre en la asignatura y participe, empezando a trabajar en las actividades, ya desde el principio y “enganchándose” al aprendizaje sin perder el interés ni disminuir su dedicación a lo largo de todo curso. En este sentido, son relevantes los estudios de (Kanaparan et al., 2013) relacionados con el *engagement* en un primer curso de programación, expresado en función de los tres indicadores que lo determinan: el esfuerzo, la persistencia y la búsqueda de soporte.

Este artículo se estructura de la siguiente manera: en el apartado siguiente se describe el escenario de la asignatura sobre la cual se realiza la intervención para mejorar el ritmo de aprendizaje. A continuación, el apartado 3 detalla la intervención. Los resultados obtenidos se muestran en el apartado 4 mientras que las limitaciones que se derivan se describen en el último apartado junto con las principales conclusiones y líneas futuras de investigación.

2. CONTEXTO

En esta sección se describe el escenario en el cual se ha planteado y realizado la intervención propuesta. La asignatura "Fundamentos de programación", obligatoria en los grados de Ingeniería Informática (INF) e Ingeniería de Tecnologías de Telecomunicación (TEL) de la Universitat Oberta de Catalunya (UOC), es también un complemento de formación en algunos másteres especializados y asignatura de libre elección dentro del programa abierto de la universidad.

Cuenta, pues, con un perfil de estudiantes muy heterogéneo, aunque todos la cursan del mismo modo y en las mismas aulas en un entorno de aprendizaje completamente virtual. De forma resumida, el 82.17 % de los estudiantes son hombres, y la mediana de edad en el momento de cursar la asignatura es de 30 años (el grupo de edad más popular es el de 31-40 años, seguido del de 26-30 años). La mayoría de los simultáneamente entre dos y tres asignaturas, una situación típica de estudiantes a tiempo parcial.

A. Diseño instruccional. Modelo de evaluación.

Como asignatura introductoria, parte de los principios básicos de la algorítmica combinados con la práctica de ejercicios sencillos en lenguaje C. Se combina la evaluación continuada basada en tres actividades de diseño algorítmico con dos prácticas de programación en lenguaje C y un examen final presencial. Las tres pruebas de evaluación continua (PEC) son opcionales, así que el estudiante decide cuántas y cuáles realiza, teniendo en cuenta que cada una de ellas representa una parte de la calificación final de la evaluación continuada. Las dos prácticas (PR), en cambio, son obligatorias, así como también el examen final (EX).

Las actividades se acompañan de los recursos necesarios para resolverla: los contenidos teóricos de algorítmica, ejemplos e indicaciones para la codificación en C y una máquina virtual con el entorno de desarrollo *Codelite* para programar. Una vez finaliza el plazo de entrega de una PEC o de una PR se publica su solución y cada estudiante recibe la calificación de su ejercicio. Las actividades de evaluación continuada se cualifican siguiendo un sistema alfabético: A (muy buena), B (buena), C+ (suficiente), C- (baja), D (muy baja) y N (no presentada). La cualificación final, sin embargo,

es numérica, de acuerdo con el sistema de calificaciones español.

El profesor utiliza el tablón del aula virtual para comunicar cualquier cuestión relativa al desarrollo de las actividades. Las dudas de los estudiantes, en cambio, se comparten y resuelven desde el foro del aula o a través del buzón personal del profesor y del estudiante. El foro es un espacio donde se espera que los estudiantes participen compartiendo sus dudas y creando conocimiento de forma colaborativa. Para la realización de la parte práctica, el estudiantado cuenta con el soporte del laboratorio de programación en C donde un profesor de laboratorio resuelve las dudas y problemas con el entorno de programación y el código en C (Marco et al., 2002).

B. Resultados

Con este planteamiento, el número de estudiantes que abandonaban la asignatura era bastante alto y el rendimiento final bajo, siempre alrededor del 30 %. Resultados habituales, acordes con la literatura y casi “esperados” y ya aceptados como normales en una asignatura inicial de programación, aún más tratándose de una docencia completamente en línea. Analizando más en detalle la actividad de los estudiantes durante diversos semestres, se observaba que el problema principal era el abandono durante las primeras semanas del semestre que atribuimos a que el aprendizaje de la programación:

1. Tiene una *curva de aprendizaje inicial alta*. Al principio cuesta entender los conceptos abstractos básicos de la algorítmica que son la base de la programación. Así que, muchos estudiantes, se pierden ya desde el inicio y, al con conseguir resolver las primeras actividades más sencillas, abandonan.
2. Es un *aprendizaje progresivo*, así que si el estudiante no logra entender un concepto inicial y resolver las primeras actividades ya es difícil que pueda continuar con los siguientes conceptos y actividades.
3. En un entorno de aprendizaje en línea, el aprendizaje se articula a través de las actividades de evaluación continua que se proponen al estudiante durante el semestre. Estas actividades marcan el *ritmo de trabajo* del estudiante a través del estudio y la práctica y también porque es a través de la retroalimentación como recibe indicaciones para solventar las dificultades y continuar progresando. Si no se sigue el ritmo, es fácil abandonar.

Así que, inspirados por el éxito de cambios precedentes en las asignaturas de matemáticas de la titulación que consiguieron cambiar esta tendencia, el curso 2017-2018 nos planteamos una intervención para suavizar la curva de aprendizaje y marcar un ritmo de trabajo más continuado, con el objetivo de reducir el abandono y conseguir que un número mayor de estudiantes consiguieran acabar con éxito la asignatura.

3. DESCRIPCIÓN

Con la intervención realizada, la asignatura no cambió ni sus objetivos, ni los recursos docentes ni el temario ni tampoco los resultados de aprendizaje esperados, así que las competencias a alcanzar antes y después de la intervención son exactamente las mismas. El rediseño del modelo instruccional se centró en modificar substancialmente el modelo y número de actividades

de evaluación continuada que se proponen hacer a los estudiantes a lo largo de todo el semestre.

Se perseguía conseguir un ritmo de trabajo más continuado por parte del estudiante, aumentar su motivación y ganas de programar y conseguir que quedase “enganchado” a la dinámica de trabajo desde el inicio del semestre. Basados en la máxima de que a programar se aprende programado, se reforzaron y replantearon las actividades de evaluación continua en varios aspectos:

- actividades *más cortas*, cada actividad incorpora y trabaja un único concepto fundamental;
- actividades *más frecuentes*, una actividad cada semana;
- todas las actividades, tanto las pruebas de evaluación continua (PEC) como las prácticas (PR), están *vinculadas entre sí* a través de un mismo contexto (caso práctico);
- el estudiante recibe una *retroalimentación* de cada actividad.

El cambio en la planificación supuso, principalmente, dividir cada PEC anterior en diversas PEC más cortas, de forma que las entregas fueran más frecuentes. El tipo de problemas incluidos seguía siendo exactamente el mismo. Simplemente se reorganizaron los mismos ejercicios en tres bloques con múltiples entregas. La PAC1 se dividió en 4 (las nuevas PEC1 a PEC4), la PEC2 en 4 (las nuevas PEC5 a PEC8) y la PEC3 se partió en 2 (las nuevas PEC9 y PEC10). Con esta nueva planificación, las entregas pasan a ser semanales, recibiendo el estudiante un retorno cada semana con la publicación de la solución. La retroalimentación individualizada del profesor y la calificación se han seguido entregando en tres momentos del semestre, después de cada bloque, es decir justo después de la cuarta, la octava y la décima actividad (PEC4, PEC8 y PEC10), respectivamente.

Así, el nuevo modelo de evaluación continua consiste en una secuencia de actividades cortas y semanales cada una de las cuales contiene un único ejercicio que combina teoría (diseño de algoritmos) y práctica (programación en C). En concreto, se proponen en total diez *actividades optativas* como pruebas evaluación continua que se combinan con la realización de dos *ejercicios prácticos obligatorios* de programación más completos que integran todos los contenidos del curso (PR1 y PR2). Más en detalle, la secuencia de actividades es la siguiente:

- PEC1: Ejercicio para trabajar tipos básicos de datos.
- PEC2: Ejercicio de expresiones con tipos básicos.
- PEC3: Ejercicio para practicar con la estructura alternativa.
- PEC4: Ejercicio de uso de la estructura iterativa.
- PEC5: Ejercicio que requiere el uso de tuplas.
- PEC6: Ejercicio de uso de acciones y funciones (modularidad).
- PEC7: Ejercicio que requiere del paso de parámetros.
- PEC8: Ejercicio que requiere el uso de tablas.
- PR1: Práctica de integración de todo lo aprendido en un proyecto de programación en C de más envergadura.
- PEC9: Ejercicio que introduce un tipo abstracto de datos sencillo: pila, cola o lista.

- PEC10: Ejercicio para practicar operaciones con pilas, colas y listas.
- PR2: Añadir al proyecto de la PR1 una estructura de un tipo de datos abstracto.

Se mantuvieron igual tanto los dos ejercicios prácticos (PR) como la evaluación final presencial a través de un examen (EX).

Las pruebas o actividades semanales están alineadas con la práctica y forman parte del mismo contexto (caso práctico), de tal modo que los ejercicios de programación semanales acaban siendo utilizados después en parte de la práctica y así se simplifica la comprensión del contexto de la práctica y se facilita su resolución. Aunque la evaluación continua es optativa, esta práctica es de realización obligatoria y es también indispensable superarla para aprobar la asignatura.

4. RESULTADOS

Para el análisis del resultado de la intervención descrita en el apartado anterior se ha optado simplemente por comparar los diferentes indicadores usados habitualmente, dado que se trata de comparar dos poblaciones diferentes solamente relacionadas por el hecho de cursar la misma asignatura de programación del grado, pero en semestres diferentes.

Se ha contado con los datos de cuatro semestres: dos previos a la modificación de la planificación de actividades (segundo semestre del curso 2016-17 y primero del curso 2017-18) y dos posteriores a los cambios (segundo semestre del curso 2017-18 y primero del curso 2018-19). De esta forma se puede comparar un curso entero antes y después de la intervención realizada. Los datos provienen del *Learning Record Store institucional* (Minguillón et al., 2018). Se ha contado con un total de 1,487 observaciones, 721 anteriores al cambio y 766 posteriores, donde cada observación es el registro de calificaciones de un estudiante matriculado en alguno de los cuatro semestres analizados.

Antes de proceder con la discusión de los resultados, es necesario comentar que el estudio presentado en este artículo se limita a realizar una comparación entre poblaciones que, aun siendo parecidas, no se pueden considerar iguales desde un punto de vista de significancia estadística. De todos modos, a pesar de las limitaciones mencionadas, la intervención realizada muestra claramente un aumento del porcentaje de estudiantes que supera la asignatura y un acusado descenso del porcentaje de no presentados en la evaluación continua.

Cada observación describe el perfil y calificaciones de un estudiante en un semestre. Las principales variables analizadas se pueden agrupar en distintas categorías:

- Perfil sociodemográfico: sexo, grupo de edad.
- Perfil de estudios: grado en el que está matriculado, vía de acceso a la titulación, si proviene de estudios de formación profesional o de estudios universitarios, número de asignaturas matriculadas en el semestre, número de semestres que lleva matriculado en el grado, veces que ha repetido la asignatura.
- Calificaciones obtenidas: calificaciones de cada PEC y de PR1 y PR2, calificación final de prácticas y de evaluación continua.

La entrega de la primera práctica (PR1) es uno de los indicadores más claros del rendimiento final del estudiante en la asignatura. De las personas que entregan la PR1, aprueban las prácticas el 83.4% y superan la evaluación continua el 93.3%. Solo el 21% de los que no entregan la práctica superan la evaluación continua de la asignatura (aunque no pueden aprobar la asignatura al ser la práctica un requisito). Así, el porcentaje de PR1 presentadas ha aumentado en los semestres posteriores al cambio en la planificación pasando del 48.13% al 60.70%. La tabla 1 muestra el desglose de notas obtenidas por los estudiantes antes y después de la intervención.

Se puede observar un incremento de los aprobados, especialmente de las calificaciones *Notable (B)* y *Excelente (A)*. Y, sobre todo, hay un importante descenso de no presentados (*N*). El cambio en la planificación parece haber mejorado también las calificaciones finales de las prácticas, incrementado el número de aprobados.

Tabla 1
Distribución de notas primera práctica (PR1)

	A	B	C+	C-	D	N
pre	9 %	15 %	8 %	2 %	12 %	54 %
post	27 %	23 %	6 %	2 %	3 %	39 %

A partir de los resultados y de la interpretación realizada, es posible pensar en las implicaciones que este análisis puede tener para el futuro de la asignatura. Por una parte, parece razonable pensar que el incremento de actividades suaviza la curva de aprendizaje inicial y consigue un ritmo de trabajo más continuado de tal modo que el estudiante se “engancha” a la asignatura desde el principio. Estos dos factores incrementan la posibilidad de superar la asignatura y también redundan en mejores resultados. Como muestra la tabla 2, el rendimiento de la asignatura mejoró significativamente los semestres posteriores a la intervención.

Tabla 2
Evolución del rendimiento y del seguimiento de la evaluación continua

	2016.2	2017.1	2017.2	2018.1
Rendimiento	34.3 %	23.5 %	52.8 %	48.7 %
Seguimiento EC	60.2 %	61.9 %	86.2 %	87.0 %

La percepción recogida a través de una encuesta realizada el primer semestre de la intervención también apunta mayoritariamente a una valoración positiva por parte de los estudiantes que argumentan que el hecho de aprender poco a poco, pero de forma constante es mejor que hacerlo de golpe y de manera intermitente. Algunos explican que al principio al ver tantas actividades programadas se asustaron e incluso esta planificación les llegó a estresar, pero que al hacer las primeras ya fueron cogiendo la dinámica y constatando las ventajas. En la nueva propuesta ven también aspectos a mejorar, como por ejemplo recibir la nota y el retorno de cada actividad antes sin tener que esperar a que acabe cada bloque.

No obstante, a pesar de los buenos resultados, hay que tener en cuenta que la intervención ha implicado también un incremento importante de trabajo para los profesores especialmente porque necesitan más tiempo para las

correcciones y por la complicación que conlleva preparar actividades de evaluación continuada vinculadas entre sí y relacionadas todas con el mismo contexto que las prácticas. Además, para algunos estudiantes es complicado seguir un ritmo de actividades semanal especialmente si cursan simultáneamente otras asignaturas con un planteamiento similar. Y, aunque los estudiantes no vieron incrementada la carga docente, sino que ésta se repartió de forma más gradual a lo largo del semestre sería interesante conocer el impacto que tiene esta intervención sobre las otras asignaturas que cursan simultáneamente.

5. CONCLUSIONES

En resumen, lo que resulta más relevante para la realización de la actividad obligatoria (y en consecuencia para la superación de la asignatura) es que el estudiante se comprometa con la asignatura, que se “enganche” desde el principio haciendo las actividades de evaluación continua. De esta forma, el acompañamiento más efectivo es el que se da justo en las primeras semanas del curso: por un lado, proporcionando un retorno personalizado cuanto antes mejor, que ayude y anime al estudiante a progresar hacia las siguientes actividades; y por otro, detectando a los estudiantes que no presentan las primeras actividades y ofrecerles el apoyo que necesiten para ponerse en marcha cuanto antes.

Partiendo de los resultados de esta investigación, el trabajo futuro pasa por el diseño, implementación y evaluación de otra intervención que mejore el retorno y acompañamiento más individualizado a los estudiantes en las primeras actividades, con el propósito de recuperar cuanto antes a los estudiantes que están en situación de riesgo de abandonar la asignatura o de no superarla. Y también, en aras de la sostenibilidad de la intervención, por evitar una carga excesiva para el profesorado diseñando un sistema de evaluación en el que, por ejemplo, algunas de las actividades sean de corrección automática. Por último, a nivel metodológico, es importante usar técnicas tipo PSM (*Propensity score matching*) para corregir los posibles sesgos derivados de la comparación de poblaciones distintas, aunque muy similares.

REFERENCIAS

- Carter A. S., Hundhausen C. D., & Adesope O. (2017). Blending measures of programming and social behavior into predictive models of student achievement in early computing courses. *ACM Transactions on Computing Education (TOCE)*, 17(3):1–20.
- Hoda R., & Andreae P. (2014). It’s not them, it’s us! why computer science fails to impress many first years. In *Proceedings of the Sixteenth Australasian Computing Education Conference*, vol. 148, 159–162.
- Kanaparan G., Cullen R., & Mason D. (2013). Self-efficacy and engagement as predictors of student programming performance. In PACIS, 282.
- Luxton-Reilly A., Albluwi I., Becker B. A., Giannakos M., Kumar A. N., Ott L., Paterson J., Scott M. J., Sheard J., & Szabo C. (2018). Introductory programming: a systematic literature review. In *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*.

- Marco-Galindo, M.J., & Prieto-Blazquez J. (2002). Necesidades específicas para la docencia de programación en un entorno virtual. *Actas JENUI*, 5-12.
- Minguillón J., Conesa J., Rodríguez M.E. & Santanach F. (2018). Learning analytics in practice: providing e-learning researchers and practitioners with activity data. In *Frontiers of Cyberlearning*, 145-167. Springer.
- Ott C., Robins A., & Shephard K. (2016). Translating principles of effective feedback for students into the cs1 context. *ACM Transactions on Computing Education (TOCE)*, 16(1):1– 27.
- Porter L., & Zingaro D. (2014). Importance of early performance in cs1: two conflicting assessment stories. In *Proceedings of the 45th ACM technical symp. on Computer science education*.
- Sorva J. (2012). Visual program simulation in *introductory programming education*. Aalto University.
- Watson C. & Li F.W. Failure rates in introductory programming revisited. (2014). In *Proceedings of the 2014 Conference on Innovation in Computer Science Education, ITiCSE '14*, page 39–44, New York, NY, USA. Association for Computing Machinery