Actas de las Jenui, vol. 6. 2021.
Páginas: 59-66

¿Quién tropieza dos veces con la misma piedra en una asignatura inicial de programación?

Maria-Jesús Marco-Galindo
Universitat Oberta de Catalunya
Barcelona, Spain
mmarcog@uoc.edu

Julià Minguillón
Universitat Oberta de Catalunya
Barcelona, Spain
jminguillona@uoc.edu

David García-Solórzano
Universitat Oberta de Catalunya
Barcelona, Spain
dgarciaso@uoc.edu

Teresa Sancho-Vinuesa
Universitat Oberta de Catalunya
Barcelona, Spain
tsancho@uoc.edu

Resumen

Aprender a programar no es sencillo y es sabido que un porcentaje elevado de estudiantes fracasa en su primer intento. De hecho, en asignaturas introductorias de programación hay muchos estudiantes repetidores. La experiencia de "Fundamentos de Programación" muestra que hay dos tipos de estudiantes repetidores: los que no hacen nada o casi nada y abandonan, y los que lo intentan, pero suspenden. En este trabajo presentamos un análisis de los resultados de los estudiantes repetidores en esta asignatura y su participación en las actividades propuestas, con la finalidad de diseñar una intervención que les ofrezca un itinerario formativo adaptado a su situación y necesidades, y que tenga en cuenta sus logros previos. Basándonos en la literatura y el análisis realizado, identificamos los factores que explican el fracaso de los estudiantes en su primer intento.

Abstract

Learning to code is not easy and a high percentage of students fail on their first attempt. In fact, in introductory programming courses there are many students who are repeating the course. The experience of "Programming Foundations" shows that there are two types of students repeating the course: those who do nothing or almost nothing and drop out, and those who try but fail. In this paper we present an analysis of the results of the repeating students repeating and their participation in the proposed activities, with the ultimate objective of designing an intervention that provides them with a learning itinerary adapted to their situation and needs, taking into account their previous achievements. Based on the literature and the analysis carried out, we enumerate the factors that explain students' failure in their first attempt.

Palabras clave

Curso inicial de programación, estudiantes repetidores, análisis del rendimiento, diseño de un cuestionario, abandono temprano.

1. Introducción

Diversos artículos [11] ponen de manifiesto que muchos estudiantes tienen dificultades para dominar los contenidos y competencias de las asignaturas introductorias de programación. Esto conlleva que las tasas de abandono de estas asignaturas sean altas –alrededor del 28-33 % [1]– y los índices de superación, bajos [26].

Una de las dificultades que presentan estas asignaturas es, como apunta Robins [18], el hecho de que los conceptos están fuertemente relacionados entre sí, de manera que los introducidos más tarde dependen de aquellos adquiridos anteriormente. En consecuencia, la acumulación de «lagunas» impide a muchos estudiantes progresar en este tipo de aprendizajes. En este sentido, varias investigaciones [23] evidencian que los estudiantes pueden necesitar más tiempo del esperado para adquirir los conocimientos de programación. De hecho, algunos se quejan de que, en el tramo final de la primera asignatura de programación, el ritmo es demasiado rápido [27]. En este sentido, los repetidores pueden ser aquellos estudiantes que implícitamente han manifestado necesitar más tiempo para asimilar los conceptos y competencias de una asignatura introductoria de programación. El objetivo final es entender las dificultades que experimentaron los repetidores, para poder diseñar una intervención que les ayude a superar la asignatura cuando la repiten. Con este propósito nos proponemos, mediante un análisis exploratorio preliminar, delimitar cuáles son los factores que llevaron a los estudiantes repetidores a no superar la asignatura

en su primer semestre.

Este artículo se organiza de la siguiente manera: el apartado 2 presenta un breve estado de la cuestión sobre los factores que influyen en el abandono en asignaturas de programación y el perfil de los estudiantes repetidores. El apartado 3 describe la metodología usada en este trabajo. El apartado 4 presenta el análisis realizado y los resultados obtenidos. En el apartado 5 se discuten los resultados y se propone una categorización de los factores identificados como relevantes para diseñar una intervención con los estudiantes repetidores. Finalmente, el apartado 6 presenta las conclusiones y las líneas de investigación futuras de este estudio.

2. Asignaturas de programación

Las asignaturas introductorias de programación tienen, en general, altas tasas de abandono y un índice de superación menor que el de otras asignaturas de primer curso [26]. Además, la literatura muestra que los resultados académicos de estas asignaturas se comportan de manera casi bimodal, con un grupo de estudiantes (llamado *effective novices*) que progresa adecuadamente obteniendo buenos resultados y otro grupo de estudiantes (llamado *ineffective novices*) que suspenden tras hacer, muchos de ellos, un esfuerzo extraordinario durante el semestre [19]. Todo esto supone que el porcentaje de estudiantes repetidores en asignaturas de programación no sea negligible. Para hacer frente a esta problemática y poder crear estrategias específicas para los estudiantes repetidores, es importante, por un lado, identificar qué características definen a estos estudiantes y, por otro, qué factores favorecen el abandono en estas asignaturas.

2.1. Motivos de abandono

Estudios como el de Kinnunen y Malmi [8] y el de Petersen et al. [16] concluyen que los dos motivos principales que llevan a los estudiantes a abandonar una asignatura inicial de programación son la falta de tiempo y motivación. Respecto a la escasez de tiempo, existen factores tales como:

- Priorización de otras asignaturas.
- Eventos ineludibles durante el semestre.
- Actividades que exigen más tiempo del esperado.
- Falta de habilidades para gestionar el tiempo.
- Sensación de no poder seguir el ritmo si uno se queda rezagado en un tema.

En cuanto a la falta de motivación, se identifican los factores siguientes:

- Desequilibrio entre la carga de trabajo y el aprendizaje obtenido.
- Frustración al perder demasiado tiempo buscando y corrigiendo errores en el código.
- Ayuda insuficiente o a destiempo por parte del profesorado.

En [16] también se revisan aspectos que influyen en el bajo rendimiento e incluso abandono de los estudiantes. Por ejemplo, se destaca que muchos estudiantes perciben que las estrategias de estudio que utilizan en otras asignaturas –más orientadas a revisar los materiales docentes– no sirven para las de programación. En esta línea, una comparativa entre estudiantes con alto y bajo rendimiento [10] identificó que estos últimos tienden a memorizar soluciones concretas de código, en lugar de entender los conceptos subyacentes.

Asimismo, el diseño de algunas actividades formativas –como puede ser un test con múltiples intentos– puede dar una sensación equivocada a los estudiantes sobre su grado de adquisición de los conceptos y competencias [16]. Del mismo modo, algunos trabajos en parejas pueden suponer que el estudiante, sin ser consciente, tenga la sensación de que es capaz de terminarla individualmente, cuando en verdad sin la ayuda de su compañero no la hubiera finalizado creándole una falsa percepción de sí mismo, de la que no es consciente hasta realizar a una actividad individual y evaluable.

2.2. Perfil de los estudiantes repetidores

Trabajos como [21] señalan que muchos repetidores no tienen una gran motivación por la programación, ni siquiera por la informática, pero que escogieron la titulación, entre otras razones, por tener una mejor salida profesional o no tener nota de acceso suficiente para realizar la carrera deseada. El mismo trabajo [21] también muestra que la mayoría de los repetidores no quieren dedicarse a trabajos directamente relacionados con la programación. Este conocimiento puede ser utilizado para, por ejemplo, contextualizar los ejercicios en diferentes ámbitos (p.e. tales como financieros, artísticos, etc.), lo cual hace que los estudiantes perciban los conceptos como más relevantes y la tasa de éxito sea mayor [6]. Asimismo, gracias a dicha contextualización, se puede llegar a motivar al estudiante haciéndole ver que los conocimientos de programación pueden ser una ventaja profesional si se dedican a estos ámbitos. Un factor indudablemente a tener en cuenta y analizado en diferentes estudios como [5] es la autoestima, menor entre los repetidores, probablemente a causa de haber suspendido en el semestre anterior.

La actividad en las primeras semanas de curso puede ser percibida por los repetidores como una pérdida de tiempo. Así lo demuestran los resultados obtenidos por [4] a través de una herramienta de monitorización, donde la participación en los ejercicios iniciales fue menor por parte de los repetidores en comparación con los es-

tudiantes que cursaban la asignatura por primera vez. En cambio, según [28], el uso de la programación por parejas (*pair programming*) durante la realización de las actividades en las semanas iniciales favorece la motivación de los repetidores. Una de las posibles razones es el hecho de poder debatir con otro estudiante de nivel similar, por ejemplo, diferentes soluciones para el mismo ejercicio. En la misma línea, la investigación de Sheard y Hagan [21] también destaca que la utilidad del trabajo en grupo era mayor para los estudiantes repetidores que para los nuevos. Por otro lado, muchos trabajos como [22] analizan qué contenidos resultan más difíciles para los estudiantes. La identificación de dichos contenidos puede resultar de mucho interés en el diseño de actividades específicas para los estudiantes que repiten la asignatura.

3. Metodología

Como parte de un proyecto de investigación más amplio sobre el diseño y análisis de intervenciones en asignaturas introductorias a la programación [14], en este trabajo nos planteamos responder las preguntas de investigación siguientes:

- RQ1: ¿Existen diferencias en los resultados obtenidos por los estudiantes repetidores en función de lo que hicieron en la convocatoria anterior?
- RQ2: ¿Existen características del perfil de los estudiantes repetidores que permitan explicar posibles diferencias en sus resultados?

A partir del análisis de los resultados de los repetidores y una revisión de la literatura, seguidamente se identificarán los factores sobre los cuales se puede incidir para mejorar la experiencia de los repetidores.

3.1. Contexto

La asignatura "Fundamentos de programación" es obligatoria en los grados de Ingeniería Informática e Ingeniería de Tecnologías de Telecomunicación de la UOC, pero también un complemento de formación en algunos másteres especializados, y asignatura libre dentro de otros programas de la universidad. Cuenta con un perfil de estudiantes muy heterogéneo.

Como asignatura introductoria, introduce los principios básicos de la programación a partir de la algorítmica lo que combina con la práctica de ejercicios sencillos en lenguaje C que progresivamente van avanzando en dificultad. La evaluación es continua y se basa en una secuencia de actividades donde se combinan ejercicios de diseño algorítmico y de programación en lenguaje C. Las actividades 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 continua. Las actividades se acompañan de los recursos necesarios para resolverla: los contenidos teóricos de algorítmica, indicaciones y ejemplos de codificación en C y una máquina virtual con el entorno de desarrollo *Codelite* para programar. Cada semana se entrega un ejercicio y a continuación se publica su solución. Una vez completados los cuatro ejercicios que forman una actividad, se recibe la calificación y también una retroalimentación personalizada del profesor. Es, pues, una evaluación formativa.

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 estudiante. Se trata de un espacio donde se espera que los estudiantes participen compartiendo sus dudas y creando conocimiento de forma colaborativa.

3.2. Datos

Los datos para el análisis se obtuvieron del *Learning Record Store* (LRS) institucional [15], el cual almacena todas las evidencias relacionadas con las actividades que realiza el estudiante, desde su matrícula hasta las notas finales obtenidas en las asignaturas matriculadas, pasando por las actividades parciales entregadas. En el estudio se incluyen todos aquellos estudiantes que se han matriculado de la asignatura de Fundamentos de Programación al menos una vez desde el semestre 2017/2 hasta el semestre 2019/1 (cuatro semestres), así como los que habían cursado la asignatura antes del semestre 2017/2 y la repiten al menos una vez en el semestre 2017/2 o posteriormente, un total de 1,206 estudiantes. Se trata, por lo tanto, de un estudio realizado con toda la población de interés. Cada semestre los estudiantes son diferentes y las diferentes cohortes (semestres) no son completamente equivalentes, pero teniendo en cuenta que el diseño de la asignatura es el mismo desde el semestre 2017/2, el análisis se realizó para todos los estudiantes en un solo conjunto de datos, en función de sus resultados del semestre anterior.

Para cada estudiante y cada semestre se dispone de los datos siguientes: sexo (Hombres: 976, 80.9 %; Mujeres: 230, 19.1 %); grupo de edad (E1 - hasta 20 años: 41, 3.4 %; E2 - 21-30 años: 538, 44.6 %; E3 - 31-40 años: 353, 29.3 %; E4 - 41 años o más: 274, 22.7 %); y si está cursando la asignatura dentro del grado de Ingeniería Informática o lo hace como complemento de otro programa (SÍ: 1016, 84.2 %; NO: 190, 15.8 %). También se dispone del número de asignaturas matriculadas y superadas simultáneamente (sin incluir la analizada en este trabajo) y de los resultados obtenidos en las actividades de evaluación continua.

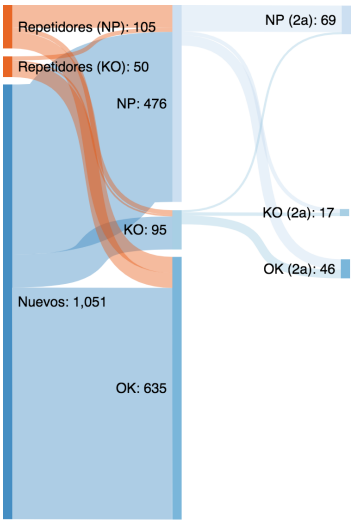

Figura 1: Resultados obtenidos por los estudiantes.

4. Resultados

En base a las preguntas de investigación se detalla a continuación el resultado de los análisis realizados.

4.1. Comportamiento de los estudiantes

Un primer análisis visual muestra que el comportamiento de los estudiantes depende de los resultados obtenidos anteriormente (OK = aprobado, KO = suspendido, NP = no presentado). La Figura 1 muestra resultado obtenido por los estudiantes en función del resultado anterior. Se puede observar que pocos estudiantes lo volvieron a intentar una segunda vez (69+17+46, 23.1 %) y que la mayor parte de ellos volvieron a suspender (69+17, 65.2 %). Estos primeros datos sugieren plantearse si el diseño actual de la asignatura es el más adecuado para los estudiantes repetidores.

De forma más detallada, el Cuadro 1 muestra el desglose de los estudiantes en función de los resultados que obtienen en cada uno de los intentos consecutivos. Desde el semestre 2017/2 (incluido), matricularon la asignatura por primera vez 1,051 estudiantes (Grupo G1), mientras que 155 la cursaron después de no haberla superado anteriormente, de los cuales 50 la suspendieron (G2) y 105 no se presentaron (G3). Distinguimos tres subgrupos para cada uno de ellos (los que aprueban (OK), los que suspenden (KO) y los que no se presentan (NP)), con el objetivo de analizar sus trayectorias. Así, tomando como referencia solamente los resultados de los estudiantes nuevos (G1), una primera inspección muestra que los estudiantes que no se habían presentado a la asignatura (G1-3) obtuvieron peores resultados que el resto. Pocos estudiantes lo intentaron una vez más, y además incluso obtuvieron peores resultados. Por otra parte, los estudiantes que habían suspendido la asignatura (G1-2) se matricularon más y además obtuvieron mejores resultados, pero parece que ya no lo intentaron de nuevo si no superaron la asignatura en su segundo intento. Por lo que respecta a la rematrícula, de los 411 estudiantes que no se presentaron en su primer intento (G1-3), sólo 79 (19.2 %) lo volvieron a intentar una segunda vez en el periodo analizado. En cambio, de los 80 estudiantes que suspendieron (G1-2), un total de 29 (36.4 %) lo volvieron a intentar, un porcentaje aún bajo pero considerablemente superior al anterior.

Otro dato que muestra el Cuadro 1 es el número de actividades entregadas (A_4, para las cuatro primeras, y A_8 para las ocho primeras). Los estudiantes que lo intentaron y suspendieron realizaron casi las mismas actividades que los que aprobaron, mientras que los que no se presentaron realizaron muy pocas de las cuatro primeras, y casi ninguna de las cuatro siguientes, después de la retroalimentación del profesor. Este hecho era también cierto para los estudiantes que ya venían de un intento anterior (G2 y G3), aunque en el caso de estudiantes no presentados el número de actividades realizadas era aún menor.

Los resultados obtenidos nos permiten responder la primera pregunta de investigación (RQ1). Efectivamente, los datos muestran que los repetidores que suspenden la asignatura tienen una segunda oportunidad que pueden aprovechar, y de hecho un porcentaje elevado así lo hizo. Puede ser que necesiten más tiempo (que un semestre) para superarla y que quizás tuvieron problemas al final, no al principio del semestre. Por otra parte, los estudiantes que no se presentaron en la convocatoria anterior se encontraron de nuevo con los mismos obstáculos y volvieron a dejar la asignatura, haciendo buena la frase (desde la perspectiva del equipo docente) «la estupidez es hacer lo mismo y esperar resultados diferentes». En este contexto, es necesario proporcionar alguna alternativa (formativa) para que estos estudiantes superen el obstáculo que les impide progresar y no repitan los mismos errores de nuevo.

4.2. Características de los repetidores

La segunda pregunta de investigación es si hay alguna característica de los estudiantes que nos permita explicar las diferencias encontradas en los resultados de los repetidores (RQ2). El Cuadro 2 muestra los indicadores seleccionados para los grupos más relevantes del Cuadro 1. Para cada grupo o subgrupo se muestra el porcentaje de hombres y mujeres, el desglose por grupo de edad, el porcentaje de estudiantes que cursan el grado de Ingeniería Informática, el número de asignaturas matriculadas simultáneamente y el numero de asignaturas superadas, en ambos casos sin incluir la asignatura en cuestión. También se muestra el porcentaje de estudiantes que realizan todas las actividades

Grupo	Origen	1a vez	2a vez	N (%)	A_4	A_8	Rematrícula
G1	Nuevos			1,051 (87.2 %)	3.10	5.64	108 (10.3 %)
G1-1		OK		560 (53.3 %)	3.91	7.70	—
G1-2		KO		80 (7.6 %)	3.56	6.54	29 (36.3 %)
G1-2-1			OK	17 (58.6 %)	3.53	6.94	—
G1-2-2			KO	7 (24.1 %)	3.43	6.29	—
G1-2-3			NP	5 (17.3 %)	1.60	3.00	—
G1-3		NP		411 (39.1 %)	1.91	2.66	79 (19.2 %)
G1-3-1			OK	20 (25.3 %)	3.70	7.15	—
G1-3-2			KO	8 (10.1 %)	3.63	6.75	—
G1-3-3			NP	51 (64.6 %)	1.69	2.29	—
G2	KO			50 (4.1 %)	3.58	6.76	5 (10.0 %)
G2-1		OK		36 (72.0 %)	3.89	7.64	—
G2-2		KO		4 (8.0 %)	3.75	7.5	3 (75.0 %)
G2-2-1			OK	2 (66.7 %)	4.00	8.00	—
G2-2-2			KO	1 (33.3 %)	4.00	8.00	—
G2-2-3			NP	0 (0.0 %)	—	—	—
G2-3		NP		10 (20.0 %)	2.40	3.30	2 (20.0 %)
G2-3-1			OK	2 (100.0 %)	3.50	7.50	—
G2-3-2			KO	0 (0.0 %)	—	—	
G2-3-3			NP	0 (0.0 %)	—	—	
G3	NP			105 (8.7 %)	2.58	4.36	19 (18.1 %)
G3-1		OK		39 (37.1 %)	3.72	7.28	—
G3-2		KO		11 (10.5 %)	3.27	5.45	2 (18.2 %)
G3-2-1			OK	2 (100.0 %)	4.00	7.50	—
G3-2-2			KO	0 (0.0 %)	—	—	—
G3-2-3			NP	0 (0.0 %)	—	—	—
G3-3		NP		55 (52.4 %)	1.64	2.07	17 (30.9 %)
G3-3-1			OK	3 (17.6 %)	3.00	7.00	—
G3-3-2			KO	1 (5.9 %)	4.00	7.00	—
G3-3-3			NP	13 (76.5 %)	1.62	1.85	—

Cuadro 1: Estudiantes según su origen y resultados académicos en semestres sucesivos.

propuestas y el índice (mediana) de la primera actividad que no entregan, un posible indicador de desconexión del ritmo de actividades propuesto.

Se puede observar que respecto al grupo de estudiantes que se matriculan de la asignatura por primera vez (G1), el grupo de los estudiantes que suspendieron la asignatura (G1-2) tenía una proporción ligeramente mayor de mujeres, así como de estudiantes más jóvenes, y de estudiantes que no estaban haciendo el grado de Ingeniería Informática. También estaban ligeramente matriculados de más asignaturas. Por otra parte, los estudiantes que aprobaron la asignatura (G1-1) obtuvieron mucho mejor rendimiento en las otras asignaturas de las que se habían matriculado en el mismo semestre, a diferencia de los que lo intentaron y suspendieron y los que no se presentaron, especialmente en este último caso. Esto puede indicar que los estudiantes que dejaron la asignatura podían estar dejando todas las asignaturas de las que se habían matriculado, es decir, que abandonaron los estudios, lo cual explica que no sea posible recuperarlos y que el indicador de rematrícula sea tan bajo. Además, se observa que los estudiantes que obtuvieron un no presentado (G1-3) son los que dejaron de entregar alguna actividad propuesta más pronto. Concretamente, fue en la segunda actividad, mientras que los que lo intentaron y suspendieron o aprobaron no lo hicieron hasta la quinta o la sexta actividad respectivamente, bien entrado el curso, después de la retroalimentación recibida por parte de su profesor. El porcentaje de estudiantes que no se presentaron pero realizaron los ejercicios propuestos fue también considerablemente menor con respecto a los que suspendieron y los que aprobaron (6.7 % vs 51.3 % y 81.8 % respectivamente). De nuevo parece factible pensar que parte de los estudiantes que abandonaron la asignatura lo hicieron porque estaban abandonando sus estudios, en realidad, al inicio del semestre, casi sin esperar a la retroalimentación del profesor de los cuatro primeros ejercicios.

5. Discusión

A partir de los resultados observados, queda claro que los estudiantes que no superan la asignatura pueden clasificarse en dos grupos, los que lo intentan hasta bien entrado el curso y suspenden, y los que no lo intentan y dejan el curso en las primeras actividades propuestas. Los dos grupos están compuestos por estudiantes similares en sexo, edad, etc., pero son claramente diferentes respecto a los resultados obtenidos, respondiendo a la segunda pregunta (RQ2). Esto es consistente con los resultados descritos por [18] so-

Grupo	Sexo	Edad	Informática	Asignaturas	Todas	Índice
G1	H: 838 (79.7 %) M: 213 (20.3 %)	E1: 38 (3.6 %) E2: 474 (45.1 %) E3: 303 (28.8 %) E4: 236 (22.5 %)	NO: 179 (17.0 %) SÍ: 872 (83.0 %)	MAT: 1.98 SUP: 1.61	50.1 %	3
G1-1	H: 454 (81.1 %) M: 106 (19.9 %)	E1: 19 (3.4 %) E2: 228 (40.7 %) E3: 172 (30.7 %) E4: 141 (25.2 %)	NO: 95 (17.0 %) SÍ: 465 (83.0 %)	MAT: 2.00 SUP: 1.66	81.8 %	6
G1-2	H: 57 (71.3 %) M: 23 (28.7 %)	E1: 8 (10.0 %) E2: 43 (53.8 %) E3: 17 (23.2 %) E4: 12 (15.0 %)	NO: 18 (22.5 %) SÍ: 62 (77.5 %)	MAT: 2.30 SUP: 0.99	51.3 %	5
G1-3	H: 327 (79.6 %) M: 84 (20.4 %)	E1: 11 (2.7 %) E2: 203 (49.4 %) E3: 114 (27.7 %) E4: 83 (20.2 %)	NO: 66 (16.1 %) SÍ: 345 (83.9 %)	MAT: 1.90 SUP: 0.31	6.7 %	2

Cuadro 2: Características de los estudiantes en función de sus resultados académicos.

bre el perfil socio-demográfico de los estudiantes de programación. No obstante, hay un subgrupo de estudiantes que provienen de otros grados, más jóvenes y con mayor proporción de mujeres, que deben cursar la asignatura como complemento de formación y que suspenden más, aunque esto también implica que lo intentan más. El hecho de ofrecer la misma asignatura a todos los estudiantes independientemente de su perfil puede causar que algunos no se adapten al ritmo y tipo de actividades propuesto [3]. Esta circunstancia debería tenerse presente en el diseño de una intervención con los estudiantes repetidores. Por lo tanto, dicho diseño debe orientarse a discriminar entre estudiantes que abandonan y los que lo intentan y suspenden, con el objetivo de corregir factores seguramente diferentes.

5.1. Factores determinantes

En términos generales, Lee y Choi [9] identificaron tres grupos de factores relacionados con el abandono: (1) intrínsecos al *estudiante*, (2) intrínsecos a la *asignatura* e (3) intrínsecos al *entorno o contexto*. El Cuadro 3 incluye los elementos de la asignatura sobre los que se puede actuar así como otros factores relativos al estudiante, destacando aquellos trabajos de la literatura donde se describen y su tipología de acuerdo a la clasificación anterior. Se indican en cursiva los factores que no pueden variar entre un semestre y otro. No se han incluido los factores socio-demográficos del estudiante y otros relacionados con su situación académica que se pueden extraer del LRS institucional. Estos factores determinarán el contenido de un cuestionario que se enviará a los estudiantes repetidores.

Cada uno de los factores descritos en el Cuadro 3 deberá desarrollarse en forma de preguntas de un cuestionario orientado a conocer la percepción de los estudiantes repetidores sobre los motivos por los que suspendieron la asignatura. Por ejemplo, con respecto al factor relativo al apoyo y guía del profesor, sería interesante conocer: ¿El profesor responde mis dudas a tiempo? ¿La retroalimentación que recibo me ayuda a entender los errores que he cometido en las actividades? ¿Las indicaciones que da el profesor me ayudan a seguir el calendario de actividades? De este modo podría saberse si el mecanismo de retroalimentación actual es diferente para estudiantes que suspenden o no se presentan, y si es necesario que el profesor actúe diferente según del perfil del estudiante, por ejemplo.

6. Conclusiones

En este trabajo hemos presentado un análisis de los resultados de los estudiantes repetidores de una asignatura de Fundamentos de Programación, en función de los resultados obtenidos. Los repetidores muestran dos comportamientos claramente diferentes: los que en su primer semestre lo intentan casi hasta el final y suspenden, y los que abandonan de forma temprana. El rendimiento de estos dos grupos en su segundo intento es claramente diferente, aprobando mucho más los que lo intentaron y suspendieron la primera vez. Así pues, ofrecer la misma asignatura a los estudiantes que la repiten es casi sinónimo de volver a fracasar, por lo que es necesario plantearse algún tipo de intervención basada en su experiencia previa, perfil y necesidades.

De acuerdo con la literatura científica sobre el comportamiento de los estudiantes repetidores y los problemas inherentes a la enseñanza de la programación en cursos iniciales, se han identificado algunos de los factores más relevantes que podrían explicar dichas diferencias, y que deberían servir para recoger más datos sobre los estudiantes y su situación en el momento de cursar la asignatura, con el objetivo de diseñar una intervención. Partiendo de los resultados de esta investigación exploratoria, el trabajo futuro pasa por el diseño, implementación y evaluación de una intervención que mejore el resultado de los repetidores en su segun-

Factor	Descripción	Tipo
Motivación	Baja motivación del estudiante, éste no busca información adicional y/o no intenta superar los retos que se le presentan [8, 10, 7].	1
Creencias y actitudes	Los estudiantes asumen que abandonar una asignatura de programación es normal, o que aprender programación es muy difícil [16, 17].	1
Gestión del tiempo	Los estudiantes carecen de habilidades de autorregulación y planificación [8, 10].	1
Estrategia de aprendizaje	Los estudiantes utilizan técnicas de estudio inapropiadas para la programación (p.e. memorización) [10, 16].	1
Dificultad	Los contenidos de la asignatura son complejos y/o abarcan demasiados conceptos [12, 22, 24].	2
Materiales teóricos	Los recursos son claros, suficientes y/o están bien organizados y actualizados [19].	2
Lenguaje y entorno de programación	El lenguaje de programación no es apropiado para aprender y/o el entorno de programación no es sencillo de instalar y utilizar [20].	2
Tipología de actividades	Hay suficiente variedad de actividades y éstas están contextualizadas y ayudan a aprender a programar de manera progresiva [8, 16].	2
Planteamiento de las actividades	Los enunciados de las actividades son claros y comprensibles, los recursos y materiales disponibles son suficientes para resolverlas y/o la cara de trabajo es adecuada[13].	2
Calendario de actividades	El número y ritmo de las actividades es adecuado. [16]	2
Apoyo y guía del profesor	La ayuda y la retroalimentación del docente no es adecuada y/o llega tarde [8, 16].	2
Presencia social	Los canales de comunicación con profesores y entre compañeros son adecuados para favorecer un sentido de pertinencia al grupo [25].	2
Conciliación	Incompatibilidad con responsabilidades laborales y/o familiares [2].	3
Imprevistos	Circunstancias sobrevenidas como enfermedades o cambios laborales[16].	3
Entorno de aprendizaje	Problemas de disponibilidad y accesibilidad al campus virtual en el caso de estudios en línea y/o dificultades para manejarse con el entorno[2].	3

Cuadro 3: Factores determinantes para el diseño de una intervención.

do intento. Para ello, se propone empezar por diseñar un cuestionario que permita conocer la perspectiva de los estudiantes repetidores en relación a su proceso de aprendizaje en la asignatura iniciado en su primera matrícula. El análisis de los datos recogidos nos permitirá definir una intervención específica, adecuada a sus necesidades, que mejore su experiencia en su segundo intento de superar la asignatura. Por descontado, y de acuerdo con el paradigma de investigación basada en el diseño, deberá analizarse dicha experiencia para introducir las mejoras necesarias en una segunda iteración.

Referencias

[1] Jens Bennedsen y Michael E. Caspersen. Failure rates in introductory programming: 12 years later. *ACM Inroads*, 10(2):30-36, abril de 2019.

[2] Janice Catterall, Janelle Davis y col. Supporting new students from vocational education and training: finding a reusable solution to address recurring learning difficulties in e-learning. *Australasian Journal of Educational Technology*, 29(5), 2013.

[3] Jessica Q. Dawson, Meghan Allen, Alice Campbell y Anasazi Valair. Designing an introductory programming course to improve non-majors' experiences. En *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, páginas 26-31, 2018.

[4] Nuno Gil Fonseca, Luıs Macedo y António José Mendes. Codeinsights: monitoring programming students' progress. En *Proceedings of the 17th International Conference on Computer Systems and Technologies*, CompSysTech '16, páginas 375-382, Palermo, Italy. ACM, 2016.

[5] Anabela Jesus Gomes, Alvaro Nuno Santos y António José Mendes. A study on students' behaviours and attitudes towards learning to program. En *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education*, ITiCSE '12, páginas 132-137, Haifa, Israel. ACM, 2012.

[6] Mark Guzdial y Allison Elliott Tew. Imagineering inauthentic legitimate peripheral participation: an instructional design approach for motivating computing education. En *Proceedings of the Second International Workshop on Computing Education Research*, ICER'06, páginas 51-58, Canterbury, United Kingdom. Association for Computing Machinery, 2006.

[7] Geetha Kanaparan, Rowena Cullen, David Mason y col. Effect of self-efficacy and emotional engagement on introductory programming students. *Australasian Journal of Information Systems*, 23, 2019.

[8] Päivi Kinnunen y Lauri Malmi. Why students drop out cs1 course? En *Proceedings of the second international workshop on Computing education research*, páginas 97-108, 2006.

[9] Youngju Lee y Jaeho Choi. A review of online course dropout research: implications for practice and future research. *Educational Technology Research and Development*, 59(5):593-618, 2011.

[10] Soohyun Nam Liao, Sander Valstar, Kevin Thai, Christine Alvarado, Daniel Zingaro, William G. Griswold y Leo Porter. *Behaviors of higher and lower performing students in CS1*. En *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*. Association for Computing Machinery, New York, NY, USA, 2019, páginas 196-202.

[11] Raymond Lister, Elizabeth S. Adams, Sue Fitzgerald, William Fone, John Hamer, Morten Lindholm, Robert McCartney, Jan Erik Moström, Kate Sanders, Otto Seppälä, Beth Simon y Lynda Thomas. A multi-national study of reading and tracing skills in novice programmers. *SIGCSE Bull.*, 36(4):119-150, 2004.

[12] Andrew Luxton-Reilly. *Learning to program is easy*. En *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*. Association for Computing Machinery, New York, NY, USA, 2016, páginas 284-289.

[13] Andrew Luxton-Reilly y Andrew Petersen. The compound nature of novice programming assessments. En *Proceedings of the Nineteenth Australasian Computing Education Conference*, páginas 26-35, 2017.

[14] Maria-Jesús Marco-Galindo, Julià Minguillón y Teresa Sancho-Vinuesa. Análisis de la progresión de los estudiantes en una asignatura introductoria a la programación mediante redes bayesianas. *Actas de las Jornadas sobre Enseñanza Universitaria de la Informática (JENUI), 5*:69-76, 2020.

[15] Julià Minguillón, Jordi Conesa, Maria Elena Rodríguez y Francesc Santanach. Learning analytics in practice: providing e-learning researchers and practitioners with activity data. En *Frontiers of Cyberlearning*, páginas 145-167. Springer, 2018.

[16] Andrew Petersen, Michelle Craig, Jennifer Campbell y Anya Tafliovich. Revisiting why students drop cs1. En *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*, Koli Calling '16, páginas 71-80, Koli, Finland. Association for Computing Machinery, 2016.

[17] Yizhou Qian y James Lehman. Students' misconceptions and other difficulties in introductory programming: a literature review. *ACM Trans. Comput. Educ.*, 18(1), octubre de 2017.

[18] Anthony Robins. Learning edge momentum: a new account of outcomes in CS1. *Computer Science Education*, 20(1):37-71, 2010.

[19] Anthony V. Robins. *Novice programmers and introductory programming*. En *The Cambridge Handbook of Computing Education Research*. Sally A. Fincher y Anthony V. Robins, edición. Cambridge Handbooks in Psychology. Cambridge University Press, 2019, páginas 327-376.

[20] Sonia Pamplona Roche y Nelson Medinilla Martınez. Evaluación de entornos de programación para el aprendizaje. *JENUI 2011*:83, 2011.

[21] Judy Sheard y Dianne Hagan. Our failing students: a study of a repeat group. En *Proceedings of the 6th Annual Conference on the Teaching of Computing: Changing the Delivery of Computer Science Education*, ITiCSE'98, páginas 223-227, Dublin, Ireland. ACM, 1998.

[22] Juha Sorva. *Visual program simulation in introductory programming education*. Aalto University, 2012.

[23] Donna Teague y Raymond Lister. Longitudinal think aloud study of a novice programmer. En *Proceedings of the Sixteenth Australasian Computing Education Conference - Volume 148*, ACE '14, páginas 41-50, Auckland, New Zealand. Australian Computer Society, Inc., 2014.

[24] Arto Vihavainen, Jonne Airaksinen y Christopher Watson. A systematic review of approaches for teaching introductory programming and their influence on success. En *Proceedings of the 10th Annual Conference on International Computing Education Research*, ICER '14, páginas 19-26, Glasgow, Scotland, United Kingdom. ACM, 2014.

[25] Joe Warren, Scott Rixner, John Greiner y Stephen Wong. Facilitating human interaction in an online programming course. En *Proc. of 45th ACM technical symposium on Computer science education*, páginas 665-670, 2014.

[26] Christopher Watson y Frederick W.B. Li. Failure rates in introductory programming revisited. En *Proceedings of the 2014 Conference on Innovation in Computer Science Education*, ITiCSE '14, páginas 39-44, Uppsala, Sweden. Association for Computing Machinery, 2014.

[27] Keith J. Whittington, Dianne P. Bills y Lawrence W. Hill. Implementation of alternative pacing in an introductory programming sequence. En *Proceedings of the 4th Conference on Information Technology Curriculum*, CITC4 '03, páginas 47-53, Lafayette, Indiana, USA. Association for Computing Machinery, 2003.

[28] Krissi Wood, Dale Parsons, Joy Gasson y Patricia Haden. It's never too early: pair programming in CS1. En *Proceedings of the 15th Australasian Computing Education Conference*, volumen 136 de *ACE'13*, páginas 13-21, Adelaide, Australia. Australian Computer Society, Inc., 2013.