

# ¿Por qué los estudiantes de una asignatura inicial de programación se convierten en repetidores?

María-Jesús Marco-Galindo, Julià Minguillón, David García-Solórzano, y Teresa Sancho-Vinuesa

**Title—Why do CS1 students become repeaters?**

**Abstract—Learning to program is hard for many students. As a result, CS1 courses have a significant percentage of repeaters. For this reason, the goal of this article is to analyze what factors affect repeaters so that a specific learning strategy for them can be performed. In this regard, a first analysis of a CS1 course shows there are two types of repeaters: (1) those who do (almost) nothing throughout the semester and drop out, and (2) those who work during the whole semester, but finally fail. According to repeaters' perceptions, they were motivated to learn to program, but it was difficult for them to keep up because of it was so hard to reconcile it with their personal context, so based on what they did the previous semester, they would prefer to continue from where they left off or change the pace of activities.**

**Index Terms—CS1 dropout factors, introductory programming course, performance analysis, repeaters' perception**

## I. INTRODUCCIÓN

DIVERSOS artículos –por ejemplo [1]– sostienen que para muchos estudiantes cursar una asignatura introductoria de programación es un gran reto debido a las dificultades que tienen que superar para dominar los contenidos y adquirir las competencias que en ella se trabajan. Esto se refleja en los resultados académicos de dichas asignaturas, cuyo comportamiento es casi bimodal [2], con un grupo de estudiantes que progresa adecuadamente logrando buenos resultados y otro formado por alumnos que suspenden tras hacer, en su mayoría, un gran esfuerzo durante el semestre. Como consecuencia, los índices de superación de estas asignaturas son bajos [3] y las tasas de abandono están en torno al 28-33 % [4]. Con este escenario es fácil entender por qué el porcentaje de repetidores en asignaturas iniciales de programación no es negligible.

En nuestro sistema educativo no es posible evitar que haya repetidores en las asignaturas introductorias de programación. Por ello el principal objetivo de esta investigación es entender las dificultades que experimentaron los estudiantes que no superaron la asignatura, para poder diseñar una intervención docente que les ayude a superarla cuando la cursen de nuevo. Concretamente, este artículo se estructura como sigue: en el apartado II se presenta un breve estado de la cuestión sobre los

factores que influyen en el abandono en las asignaturas introductorias de programación. También se recopilan, en el apartado III, las principales características que determinan, según la literatura, el perfil de los estudiantes repetidores de programación. En el apartado IV se plantean las preguntas de la presente investigación y se explica la metodología empleada para responderlas. Se muestran los resultados en el apartado V, que son discutidos en el apartado VI. El artículo concluye en el apartado VII con las posibles líneas de investigación futuras.

Se trata de un artículo que extiende la investigación publicada en la conferencia JENUI 2021 y que fue elegido para su envío a IEEE-RITA [5]. Respecto al artículo de partida, se amplía la revisión bibliográfica y se incorpora una tercera pregunta de investigación sobre la percepción de los estudiantes repetidores. Para responderla, se ha diseñado e implementado un cuestionario específico cuyos resultados se analizan y discuten de acuerdo con el propósito de la investigación.

## II. MOTIVOS DE ABANDONO

En este apartado se identifican los factores que, de acuerdo con la literatura, favorecen el abandono en las asignaturas introductorias de programación. Dichos factores están ordenados según las tres categorías definidas por Lee y Choi [6]: intrínsecos a la asignatura, al contexto y al estudiante. Así pues, la primera depende de la asignatura y su diseño, y las últimas, del alumno y sus circunstancias.

### A. Interrelación entre Conceptos

Una de las dificultades que presentan las asignaturas introductorias de programación es, como apunta Robins [7], el hecho de que los conceptos están fuertemente relacionados entre sí, de manera que los elementos introducidos más tarde dependen de aquellos adquiridos anteriormente. Este hecho, unido a la complejidad del planteamiento docente de la propia materia [8], provoca que sean bastantes los estudiantes que intentan avanzar sin asimilar correctamente todos los conceptos y competencias, hasta llegar a un punto en que han acumulado tantas carencias que se hallan en una situación de colapso y, en consecuencia, no les queda más remedio que abandonar. A esto se suma el hecho de que, como apunta [9], los conceptos presentados inicialmente tienden a ser tratados de nuevo y reevaluados indirectamente a lo largo del curso. En este sentido, cabe destacar que incluso metodologías que pretenden evitar que se aprenda un concepto sin dominar uno previo, como es el caso del *mastery learning*, pueden generar procrastinación [10].

Manuscrito recibido el día de mes de año; revisado día de mes de año; aceptado día de mes de año. English version received Month, day-th, year. Revised Month, day-th, year. Accepted Month, day-th, year.

M.-J. Marco-Galindo, J. Minguillón, D. García-Solórzano y T. Sancho-Vinuesa. Universitat Oberta de Catalunya, Barcelona, España ({mmarco, jminguillona, dgarciaso, tsancho}@uoc.edu.

### B. Lenguaje de Programación Utilizado

El objetivo de cualquier asignatura introductoria de programación debería ser *aprender a programar, no aprender C*, por ejemplo. Esto implica que la atención debe ponerse en la resolución de problemas así como en la comprensión y diseño de algoritmos. Esto supone también que el lenguaje de programación empleado durante la asignatura debe ser un mero vehículo para tal fin, es decir, debería ser tratado como un recurso didáctico más. Sin embargo, la realidad es que ni los lenguajes de programación están pensados para aprender a programar ni los estudiantes saben diferenciar entre *qué es programar* y *qué es el lenguaje de programación* [11]. Así pues, hay estudiantes que suspenden o abandonan, no por el diseño instruccional de la asignatura, sino por la complejidad que les supone el lenguaje de programación utilizado [12] y las dificultades inherentes al diseño de algoritmos.

### C. Tipología de las Actividades

Algunas actividades formativas –como puede ser un test de respuesta múltiple– pueden dar una sensación equivocada a los estudiantes sobre su grado de adquisición de los conceptos y competencias [13]. Del mismo modo, algunas actividades realizadas en parejas pueden suponer que el estudiante, sin ser consciente ello, tenga la sensación de que es capaz de terminarla individualmente, cuando en verdad sin la ayuda de su compañero no la hubiera finalizado. Es así como el estudiante puede tener una falsa percepción del conocimiento real que tiene de la materia, de la que no es consciente hasta realizar una actividad individual y evaluable.

### D. Calendarización de la Asignatura

Investigaciones como [14] evidencian que los estudiantes pueden necesitar más tiempo del estimado por el profesorado para adquirir los conocimientos de programación que se esperan en la asignatura. En este sentido, algunos trabajos [15] señalan que, a menudo, en el tramo final de la asignatura, el ritmo se acelera, siendo demasiado rápido para muchos alumnos. Quizás uno de los motivos de esta aceleración se deba a que, como apunta Luxton-Reilly [8], las expectativas de los profesores sean irreales y, por ello, se introduzcan demasiados conceptos nuevos al final del curso, suponiendo que los fundamentos han sido asimilados por los estudiantes.

### E. Adaptación a la Universidad

Como comentan Lowe y Cook [16], una cifra nada despreciable de estudiantes tienen problemas para adaptarse a la universidad debido a diversos factores. Por ejemplo, según dicho estudio, alrededor del 40 % sintieron que el trato con los profesores era más distante que en niveles de estudio previos, lo que dificultaba su acomodación al entorno universitario. Así, algunos estudiantes que abandonan la asignatura introductoria de programación, en realidad, han abandonado sus estudios.

### F. Estrategias de Aprendizaje Inadecuadas

Es sabido que la universidad exige a sus estudiantes una mayor autonomía en comparación a niveles de educación inferiores. Sin embargo, muchos no son capaces de tenerla. Según

el trabajo [16], casi un tercio de estudiantes tuvieron absoluta dificultad para realizar un aprendizaje autorregulado.

Por lo que respecta a las asignaturas introductorias de programación en sí, Petersen et al. [13] destacan que muchos estudiantes perciben que las estrategias de estudio que usan en otras asignaturas –más orientadas a leer los materiales docentes– no sirven para las asignaturas de programación. Según dichos autores, esto influye en el bajo rendimiento e incluso abandono de los alumnos. En este sentido, trabajos como [11] también señalan la diferencia que existe entre la manera de estudiar otras asignaturas (incluso en el colegio) y la necesaria para aprender a programar. Asimismo, una comparativa [17] entre estudiantes con alto y bajo rendimiento identificó que estos últimos tienden a memorizar soluciones concretas de código, en lugar de entender los conceptos subyacentes. Igualmente, Hawi [18] enfatiza que la estrategia de aprendizaje fue la principal causa de éxito y fracaso en su estudio. De hecho, indica que son muchos los estudiantes que se percatan de que existe una estrategia, basada en la práctica continua, que aumenta la posibilidad de aprobar.

### G. Falta de Tiempo

Algunos estudiantes ven que realizar las actividades de programación les exige demasiado tiempo y prefieren concentrarse en otras asignaturas [13], [19], [20]. De ellos, muchos no saben cómo realizar los ejercicios, se encallan invirtiendo mucho tiempo y terminan por abandonar [19].

Otros estudiantes se planifican mal y empiezan a hacer los ejercicios demasiado tarde, de modo que no tienen suficiente tiempo para finalizarlos antes de la fecha límite [19].

Finalmente, existen situaciones personales y/o profesionales del estudiante que pueden suponer una reducción de tiempo disponible para la asignatura y provocar el abandono.

### H. Falta de Confianza en Programar

Algunas investigaciones como [21] han visto que la desconfianza del estudiante acerca de su capacidad para programar puede contribuir a su abandono. A esto se suma que los alumnos evalúan sus progresos a menudo, pudiéndoles dar una sensación equivocada de su habilidad para programar, afectando directamente a su autoeficacia [22]. Durante dichas autoevaluaciones, los estudiantes tienen en cuenta ítems tales como las propias expectativas, el tiempo necesario para realizar las actividades, la comparación con otros, etc. Dichos factores pueden incluso hacerles creer que comprenden un concepto cuando en realidad no lo entienden o no lo dominan suficientemente [23].

### I. Falta de Motivación

La motivación, junto con la confianza, es un factor afectivo muy relevante en cualquier proceso de aprendizaje. En este sentido, muchos estudiantes se frustran al ver que no progresan pese a dedicar tiempo y esfuerzo a la asignatura [13], [19].

También hay alumnos que se sienten solos al no recibir ayuda suficiente del profesor o al llegar ésta tarde [13]. Esto les exige tener una alta motivación intrínseca que, en caso de no tenerla, puede frustrarles y hacerles abandonar.

TABLA I  
FACTORES DETERMINANTES PARA EL DISEÑO DE UNA INTERVENCIÓN

Factor	Descripción	Tipo
Motivación	Los estudiantes no buscan información adicional ni se enfrentan a los retos que se les presentan [17], [19], [33].	1
Creencias y actitudes	Los estudiantes asumen que abandonar una asignatura de programación es normal, o que aprender programación es muy difícil [13], [34].	1
Gestión del tiempo	Los estudiantes carecen de habilidades de autorregulación y planificación [17], [19].	1
Estrategia de aprendizaje	Los estudiantes utilizan técnicas de estudio inapropiadas para la programación (p.ej. memorización) [13], [17].	1
Dificultad	Los contenidos de la asignatura son complejos y/o abarcan demasiados conceptos [8], [35], [36].	2
Materiales teóricos	Los recursos no son claros, suficientes y/o no están bien organizados y actualizados [2].	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 [37].	2
Tipología de actividades	No hay suficiente variedad de actividades y estas no están contextualizadas ni ayudan a aprender a programar de manera progresiva [13], [19].	2
Planteamiento de las actividades	Los enunciados de las actividades no son claros ni comprensibles, los recursos y materiales disponibles no son suficientes para resolverlas ni la carga de trabajo es adecuada [38].	2
Calendario de actividades	El número y ritmo de las actividades no son adecuados [13].	2
Apoyo y guía del profesor	La ayuda y la retroalimentación del docente no es adecuada y/o llega tarde [13], [19].	2
Presencia social	Los canales de comunicación con profesores y entre compañeros no son adecuados para favorecer un sentido de pertenencia al grupo [39].	2
Conciliación	Incompatibilidad con responsabilidades laborales y/o familiares [40].	3
Imprevistos	Circunstancias sobrevenidas como enfermedades o cambios laborales [13].	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 [40].	3

hecho de haber suspendido la asignatura el semestre anterior.

Por otro lado, las actividades en las primeras semanas de curso pueden ser percibidas por los repetidores como una pérdida de tiempo. Así lo demuestran los resultados obtenidos por [28] 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 estudiantes que cursaban la asignatura por primera vez. En cambio, según [29], 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 diferentes soluciones para un mismo ejercicio con otro estudiante de nivel similar. En la misma línea, la investigación de Sheard y Hagan [26] también destaca que la utilidad del trabajo en grupo era mayor para los estudiantes repetidores que para los nuevos.

#### IV. METODOLOGÍA

Como parte de una investigación más amplia sobre el diseño y análisis de intervenciones en asignaturas introductorias de programación [30], en este trabajo planteamos responder las preguntas de investigación siguientes:

- RQ1: ¿Existen diferencias en los resultados de los estudiantes repetidores en función de lo que hicieron en la convocatoria anterior?
- RQ2: ¿Existen rasgos en el perfil de los repetidores que expliquen las posibles diferencias en sus resultados?
- RQ3: ¿Cuál es la percepción de los alumnos repetidores sobre los motivos por los que no superaron la asignatura?

A partir del análisis de los resultados de los estudiantes repetidores y una revisión de la literatura, se identificarán los factores que pueden ser motivo de no superar la asignatura la primera vez que se cursa. A partir de estos factores se diseñará una encuesta para conocer la percepción de los estudiantes repetidores sobre cuáles creen que fueron dichos factores.

final de la evaluación continua. Las actividades se acompañan de los recursos necesarios para resolverla: contenidos teóricos de algorítmica, indicaciones y ejemplos de codificación en C, y una máquina virtual con el entorno de desarrollo *Codelite*. Cada semana se entrega una actividad y a continuación se publica su solución que sirve como retroalimentación grupal. Una vez se completa un grupo de cuatro actividades, se recibe la calificación de cada una de ellas y también una retroalimentación personalizada del profesor. El profesor utiliza el tablón del aula virtual para comunicar cualquier cuestión relacionada con las actividades. Las dudas de los estudiantes, en cambio, se comparten y resuelven a través del foro del aula o del correo del profesor. En el foro se espera que los alumnos participen activamente, creando así conocimiento de forma colaborativa. Además, disponen de un aula específica, llamada «Laboratorio de C», donde se resuelven dudas de *Codelite* y del lenguaje C.

#### B. Análisis Exploratorio

Los datos para el análisis se obtuvieron del *Learning Record Store* (LRS) institucional [32], 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, sean repetidores o no, 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), lo que representa 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

TABLA II  
ENCUESTA SOBRE LOS FACTORES QUE INFLUYEN EN EL HECHO DE HABER REPETIDO LA ASIGNATURA

Categoría	Ítem	Pregunta
Dificultad e interés	Q2.1_1	Estaba motivado para aprender a programar
	Q2.1_2	Aprender a programar me pareció complicado
Recursos	Q2.2_1	Los materiales docentes eran claro y suficientes
	Q2.2_2	Los materiales docentes estaban bien organizados y se podía acceder fácilmente
	Q2.2_3	Pude instalar fácilmente el <i>software</i>
Actividades	Q2.3_1	El ritmo de las actividades era adecuado
	Q2.3_2	Los enunciados de las actividades eran claros y comprensibles
	Q2.3_3	Los materiales docentes eran suficientes para resolver las actividades
	Q2.3_4	Me veía capaz de resolver las actividades planificadas
Comunicación	Q2.4_1	El profesor/a respondía adecuadamente y en tiempo a mis dudas
	Q2.4_2	El <i>feedback</i> que recibía del profesor/a me ayudaba a entender los errores que había cometido
	Q2.4_3	El soporte que recibía del laboratorio me ayudó a resolver problemas
	Q2.4_4	El foro del aula me permitió resolver mis dudas
Contexto personal	Q2.5_1	Pude compaginar bien la asignatura con mi vida personal y profesional
	Q2.5_2	La combinación de asignaturas que matriculé no fue adecuada

datos, en función de sus resultados del semestre anterior. Para cada estudiante 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 - de 21 a 30 años (538, 44,6 %) / E3 - de 31 a 40 años (353, 29,3 %) / E4 - 41 años o más (274, 22,7 %).
- ¿Cursan la asignatura como parte del Grado de Ingeniería Informática o no?: SÍ (1.016, 84,2 %) / NO (190, 15,8 %).

Además, para cada estudiante y semestre, se dispone de los datos siguientes relacionados con su actividad académica:

- Número de asignaturas matriculadas y superadas a la vez (sin incluir «Fundamentos de Programación»).
- Resultados obtenidos en las actividades semanales.

### C. Instrumento de Medida: la Encuesta

Como tarea previa al diseño de la encuesta, a partir de la literatura y siguiendo la categorización de Lee y Choi [6] que identifica tres grupos de factores relacionados con el abandono –(1) intrínsecos al estudiante, (2) intrínsecos a la asignatura, e (3) intrínsecos al contexto–, se detectaron los elementos de la asignatura sobre los que se puede actuar, así como otros factores relativos al estudiante. La Tabla I sintetiza estos elementos, destacando aquellos trabajos donde se describen y su tipología según la clasificación anterior. No se han incluido factores socio-demográficos del estudiante ni otros vinculados a su situación académica, los cuales se pueden extraer del LRS.

Cada uno de los factores descritos en la Tabla I se integró como ítem en una encuesta orientada a conocer la percepción de los estudiantes repetidores sobre los motivos por los que suspendieron la asignatura. De este modo podría saberse, por ejemplo, si la retroalimentación actual es percibida diferente por estudiantes que suspenden o no se presentan, y, por tanto, es necesario que el profesor actúe de manera diferenciada según el perfil del estudiante. La encuesta consta de 15 preguntas divididas en cinco categorías: (1) dificultad e interés, (2) recursos, (3) actividades, (4) comunicación y (5) contexto personal del estudiante (ver Tabla II).

Además de las 15 preguntas que se responden de acuerdo a una escala de Likert de cinco niveles, una última pregunta solicita a los estudiantes que ordenen según la importancia que

tendría para ellos cada una de las siguientes frases:

- «Me gustaría poder continuar desde donde lo dejé».
- «Me gustaría poder adaptar el ritmo de actividades».
- «Me gustaría recibir un *feedback* más personalizado».
- «Me gustaría disponer de más materiales docentes».

La encuesta se envió a todos los estudiantes repetidores matriculados en la asignatura «Fundamentos de Programación» durante dos semestres consecutivos. Los estudiantes recibieron la encuesta al inicio del curso mediante un correo electrónico y la respondieron de manera anónima.

## V. RESULTADOS

Este apartado presenta los resultados obtenidos del análisis exploratorio y de la encuesta enviada a estudiantes repetidores.

### A. El Comportamiento en el Primer Semestre Es Relevante

La Figura 1 muestra la evolución de los estudiantes semestre a semestre. Un primer análisis visual indica que el comportamiento de los estudiantes repetidores depende de los resultados obtenidos anteriormente. Por ejemplo, se puede observar que pocos estudiantes volvieron a matricularse el siguiente semestre (69+17+46, un 23,1 %) y que la mayor parte de ellos volvieron a suspender (69+17, un 65,2 %). Estos datos cuestionan si el diseño actual de la asignatura es el más adecuado para los estudiantes repetidores, ya que la falta de una estrategia docente específica para ellos hace que pocos saquen provecho de su segundo intento.

TABLA III  
ESTUDIANTES SEGÚN SU ORIGEN Y RESULTADOS ACADÉMICOS EN SEMESTRES SUCESIVOS

Grupo	Origen	1ª vez	2ª vez	N (%)	A <sub>4</sub>	A <sub>8</sub>	Rematrícula
G1				1.051 (87,2 %)	3,10	5,64	108 (10,3 %)
G1-1		OK	-	560 (53,3 %)	3,91	7,70	-
G1-2				80 (7,6 %)	3,56	6,54	29 (36,3 %)
G1-2-1	Nuevos	KO	OK	17 (58,6 %)	3,53	6,94	-
G1-2-2		KO	KO	7 (24,1 %)	3,43	6,29	-
G1-2-3		KO	NP	5 (17,3 %)	1,60	3,00	-
G1-3				411 (39,1 %)	1,91	2,66	79 (19,2 %)
G1-3-1	NP	OK	OK	20 (25,3 %)	3,70	7,15	-
G1-3-2		OK	KO	8 (10,1 %)	3,63	6,75	-
G1-3-3		OK	NP	51 (64,6 %)	1,69	2,29	-
G2				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	7,50	3 (75,0 %)
G2-2-1		KO	OK	2 (66,7 %)	4,0	8,00	-
G2-2-2		KO	KO	1 (33,3 %)	4,0	8,00	-
G2-2-3		KO	NP	0 (0,0 %)	-	-	-
G2-3				10 (20,0 %)	2,40	3,30	2 (20,0 %)
G2-3-1	NP	OK	OK	2 (100,0 %)	3,50	7,50	-
G2-3-2		OK	KO	0 (0,0 %)	-	-	-
G2-3-3		OK	NP	0 (0,0 %)	-	-	-
G3				105 (37,1 %)	2,58	4,36	19 (18,1 %)
G3-1		OK	-	39 (37,1 %)	3,72	7,28	-
G3-2	NP			11 (10,5 %)	3,27	5,45	2 (18,2 %)
G3-2-1		KO	OK	2 (100,0 %)	4,00	7,50	-
G3-2-2		KO	KO	0 (0,0 %)	-	-	-
G3-2-3		KO	NP	0 (0,0 %)	-	-	-
G3-3				55 (52,4 %)	1,64	2,07	17 (30,9 %)
G3-3-1	NP	OK	OK	3 (17,6 %)	3,00	7,00	-
G3-3-2		OK	KO	1 (5,9 %)	4,00	7,00	-
G3-3-3		OK	NP	13 (76,5 %)	1,62	1,85	-

io que respecta a la rematrícula, de los 411 estudiantes que no se presentaron en su primer intento (G1-3), solo 79 (19,2 %) lo volvieron a intentar una segunda vez en el periodo analizado. En cambio, de los 80 estudiantes que suspendieron la primera vez (G1-2), 29 (36,4 %) lo volvieron a intentar, un porcentaje aún bajo pero considerablemente superior al anterior.

Otro dato de la Tabla III es el número de actividades entre-

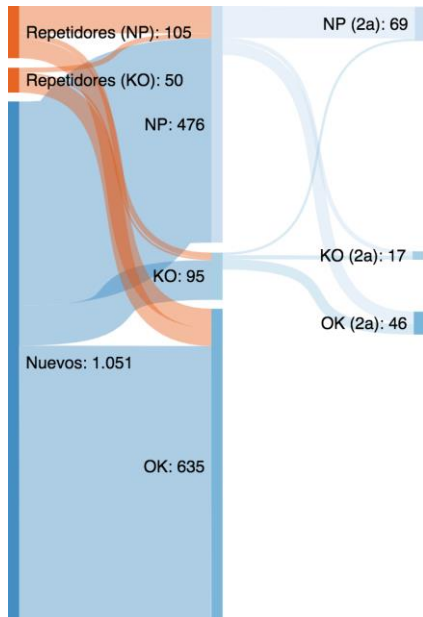


Fig. 1. Resultados obtenidos por los estudiantes (OK: aprobado, KO: suspendido, NP: no presentado).

*B. No Hay un Único Perfil de Estudiante Repetidor*

En relación con las características de los estudiantes que puedan explicar las diferencias encontradas en los resultados de los repetidores (RQ2), es interesante observar en la Tabla IV los indicadores seleccionados para los grupos más relevantes de la Tabla III. 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 número de asignaturas superadas, en ambos casos sin incluir la asignatura en cuestión. También se muestra el porcentaje de estudiantes que realizan todas las actividades 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. No obstante, las diferencias encontradas no son realmente importantes, aunque no deben ser descartadas completamente.

Por otra parte, los estudiantes que aprobaron la asignatura (G1-1) tuvieron mucho mejor rendimiento en las otras asignaturas matriculadas en el mismo semestre, y la mayoría las aprobaron todas (83,0 % de asignaturas aprobadas) a diferencia de los que lo intentaron y suspendieron (43,0 %) y los que no se presentaron (16,3 %), especialmente en este último caso. Esto puede indicar que los estudiantes que dejaron la asignatura podrían estar dejando todas las asignaturas de las que se habían matriculado, es decir, que abandonaron los estudios, lo que explica que no sea posible recuperarlos y que el indicador de rematrícula en un semestre posterior sea tan bajo.

Además, se puede ver también que los estudiantes que obtuvieron un «No presentado» (G1-3) son los que dejaron de entregar alguna actividad propuesta más pronto y no hicieron

casi ninguna. El porcentaje de estudiantes que no se presentaron, pero realizaron todas las actividades propuestas fue también considerablemente menor con respecto a los que suspendieron y los que aprobaron (6,7 % vs. 51,3 % y 81,8 % respectivamente). Concretamente, este abandono se produjo 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. De nuevo parece factible pensar que parte de los estudiantes que abandonaron la asignatura lo hicieron porque dejaron sus estudios al inicio del semestre, casi sin esperar a la retroalimentación del profesor de las cuatro primeras actividades.

*C. Resultados de la Encuesta*

La Tabla V muestra los resultados de la encuesta que los estudiantes repetidores respondieron respecto a su experiencia en el semestre anterior, en el cual no superaron la asignatura. Es importante destacar que la participación en la encuesta fue muy reducida (N = 38, un 20,1%), pero ya era lo esperado teniendo en cuenta el perfil de los estudiantes repetidores, los cuales suelen excusar su participación en cualquier actividad no evaluable por falta de tiempo.

TABLA IV  
CARACTERÍSTICAS DE LOS ESTUDIANTES EN FUNCIÓN DE SUS RESULTADOS ACADÉMICOS

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 %)	MATRICULADAS: 1,98 SUPERADAS: 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 %)	MATRICULADAS: 2,00 SUPERADAS: 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 %)	MATRICULADAS: 2,30 SUPERADAS: 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 %)	MATRICULADAS: 1,90 SUPERADAS: 0,31	6,7 %	2

Como se puede observar, destacan la alta motivación para aprender a programar de la mayoría de estudiantes repetidores que respondieron (Q2.1\_1,  $\mu = 4,37$ ), mientras que en el otro extremo se encuentra el poder compaginar el ritmo de actividades del semestre con el entorno personal y profesional (Q2.5\_1,  $\mu = 1,87$ ). El resto de indicadores están bastante centrados, aunque algunos tienen una mayor dispersión, como la valoración del *feedback* recibido (Q2.4\_2,  $\sigma = 1,42$ ).

Un análisis de las correlaciones entre ítems de la encuesta muestra que solo algunas son significativas ( $p < 0,01$ ). Hay una correlación positiva entre los recursos docentes y el soporte que dichos recursos ofrecen para realizar las actividades (Q2.2\_1 y Q2.3\_3, 0,52), entre la claridad de las actividades y el soporte / retroalimentación recibidos por parte del profesor (Q2.3\_2 y Q2.4\_1 / Q2.4\_2, 0,49 / 0,3 respectivamente), así como el soporte recibido por parte del laboratorio y la actividad del profesor (Q2.4\_1 / Q2.4\_2 y Q2.4\_3, 0,74 / 0,49 respectivamente). Como correlaciones negativas solo aparecen entre la percepción de dificultad de aprender a programar y la utilidad de los materiales docentes y la autoeficacia (Q2.1\_2 y Q2.3\_3 / Q2.3\_4, -0,33 / -0,45 respectivamente).

En relación a lo que los estudiantes repetidores desearían para volver a afrontar la asignatura una segunda vez, claramente lo más valorado es poder adaptar el ritmo de actividades (escogido por un 50,0 % de los estudiantes repetidores), seguido de poder continuar desde el punto donde se abandonó la asignatura (un 26,3 %) y tener más apoyo del profesor (un 15,8 %), mientras que disponer de más materiales docentes queda en última posición (7,9 %).

## VI. DISCUSIÓN

Antes de proceder con la discusión de los resultados obtenidos, es necesario establecer las limitaciones del estudio. Disponemos de los datos del LRS institucional y de las respuestas de los estudiantes a la encuesta. Dado que el número de respuestas obtenido en la encuesta ha sido muy reducido en números absolutos, los resultados solamente pueden ser usados de forma orientativa, sin posibilidad de hacer generalizaciones o de efectuar análisis más profundos.

Los resultados obtenidos nos permiten responder la primera pregunta de investigación (RQ1) acerca de las diferencias que existen entre los estudiantes que repiten la asignatura según su desempeño en la convocatoria anterior. 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. Esto es consistente con los resultados descritos por Porter y Zingaro [41]. Los datos muestran que los repetidores que suspendieron la asignatura tienen una segunda oportunidad que pueden aprovechar, y de hecho un porcentaje elevado así lo hace. 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 abandonar.

TABLA V  
RESULTADOS DE LA ENCUESTA. LAS PREGUNTAS FORMULADAS EN NEGATIVO SE INDICAN CON (NEG.)

Concepto	Ítem	Med.	$\mu$	$\sigma$
Motivación	Q2.1_1	5	4,37	0,82
Dificultad (neg.)	Q2.1_2	3	2,97	1,20
Recursos	Q2.2_1	3	3,00	1,27
Organización	Q2.2_2	4	3,58	1,15
Herramientas	Q2.2_3	4	3,76	1,38
Ritmo	Q2.3_1	3	2,63	1,00
Actividades	Q2.3_2	4	3,24	0,97
Soporte	Q2.3_3	3	2,84	1,03
Autoeficacia	Q2.3_4	3	2,68	1,16
Profesor	Q2.4_1	4	3,58	1,29
Retroalimentación	Q2.4_2	3	3,24	1,42
Laboratorio	Q2.4_3	3	3,53	1,18
Comunicación	Q2.4_4	3,5	3,53	1,06
Contexto personal	Q2.5_1	1,5	1,87	1,12
Matrícula (neg.)	Q2.5_2	3	2,79	1,28

En cuanto a la segunda pregunta (RQ2), los dos grupos están formados por estudiantes similares en sexo, edad, etc. Esto también coincide con los resultados descritos por [7] sobre 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 [42]. 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.

En relación con la percepción de los estudiantes (RQ3), los resultados de la encuesta, aun siendo limitados por la poca cantidad de respuestas obtenidas, ya dejan entrever algunos aspectos interesantes a tener en cuenta. Se observa, por ejemplo, que los estudiantes estaban muy motivados para aprender a programar, pero que les costó mucho compaginar la asignatura con su vida personal y profesional. No obstante, están más bien en desacuerdo con que su matrícula no fue adecuada, lo cual puede indicar que la asignatura les parecía que no requeriría tanta dedicación *a priori*, y una vez la están cursando mayoritariamente reconocen que seguir el ritmo de actividades propuesto resultó complicado. Esto se ve reforzado por el hecho de que poder ajustar el ritmo de actividades sería su primera opción si pudieran cambiar algo del diseño del curso. Y, de hecho, el ritmo semanal de actividades de evaluación, similarmente a lo descrito por [2], polariza a los estudiantes en los dos grupos, los que abandonan la asignatura –que normalmente la dejan ya en la segunda actividad–, y los que lo intentan, pero suspenden la asignatura –que suelen hacer hasta la quinta actividad. Los que aprueban, si fallan alguna actividad, suele ser la sexta de las ocho actividades. Este efecto de polarización también se describe en experiencias de *mastery learning* basadas igualmente en la realización de actividades continua-

das de evaluación [10]. Además, los estudiantes que consideran que aprender a programar es difícil, también se ven menos capacitados para hacer las actividades y son los que consideran que no tienen suficiente con los materiales para resolverlas. No obstante, no quieren más recursos. Quizás necesitan otros recursos más visuales, por ejemplo *screencasts*, que les muestren paso a paso cómo resolver un problema.

Finalmente, la detección temprana de estudiantes con problemas puede ser una estrategia general, de manera que no se tenga que esperar a que los repetidores se conviertan en tales o que no abandonen la asignatura prematuramente y que, en un segundo intento, estén mejor preparados para superarla.

Excepto los indicadores relativos al contexto personal, el resto son, en general, positivos. Así pues, los factores de diseño y estrategia docente de la asignatura están bien dirigidos.

## VII. CONCLUSIONES

En este trabajo hemos presentado un análisis de los resultados de los estudiantes repetidores de «Fundamentos de Programación» así como de su experiencia en la asignatura. Los repetidores muestran dos comportamientos claramente diferentes: (1) los que en su primer semestre lo intentan casi hasta el final y suspenden, y (2) 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. En cualquier caso, ofrecer la misma asignatura a los estudiantes que la repiten no garantiza que mejoren sus resultados y no repitan los errores de nuevo. Por ello, es necesario plantearse algún tipo de intervención diferente para estos estudiantes 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. Se ha diseñado y administrado una encuesta cuyos resultados arrojan luz sobre la perspectiva de los estudiantes en relación con su paso por la asignatura. Los resultados de la encuesta muestran que los estudiantes estaban motivados, pero no todos pudieron seguir el ritmo de la asignatura ni hacer las actividades propuestas. Esto se debió principalmente a circunstancias propias de su contexto personal y profesional.

A partir de los resultados de esta investigación exploratoria, el trabajo futuro pasa por el diseño, implementación y evaluación de una intervención que mejore los resultados de los repetidores en su segundo intento. Esta intervención debe abordar los problemas detectados, especialmente los relativos a la dificultad para seguir el ritmo semanal de las actividades y a la rentabilización de lo que ya hicieron en su primer semestre.

## AGRADECIMIENTOS

Los autores quieren agradecer el premio al mejor artículo al Comité Científico de las XXVII Jornadas sobre la Enseñanza Universitaria de la Informática (JENUI) así como su recomendación a la revista VAEP-RITA.

Esta investigación ha sido parcialmente financiada por el proyecto 2017SGR1619 de la Generalitat de Catalunya.

## REFERENCIAS

- [1] R. Lister, E. S. Adams, S. Fitzgerald, W. Fone, J. Hamer, M. Lindholm, R. McCartney, J. E. Moström, K. Sanders, O. Seppälä, B. Simon, y L. Thomas, "A multi-national study of reading and tracing skills in novice programmers," *SIGCSE Bull.*, vol. 36, no. 4, pp. 119–150, 2004. [Online]. Available: <http://doi.acm.org/10.1145/1041624.1041673>.
- [2] A. V. Robins, *Novice Programmers and Introductory Programming*, ser. Cambridge Handbooks in Psychology. Cambridge University Press, 2019, p. 327–376.
- [3] C. Watson y F. W. Li, "Failure rates in introductory programming revisited," en *Proceedings of the 2014 Conference on Innovation in Computer Science Education*, ser. ITiCSE '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 39–44. [Online]. Available: <https://doi.org/10.1145/2591708.2591749>.
- [4] J. Bennedsen y M. E. Caspersen, "Failure rates in introductory programming: 12 years later," *ACM Inroads*, vol. 10, no. 2, pp. 30–36, Apr. 2019. [Online]. Available: <http://doi.acm.org/10.1145/3324888>.
- [5] M. J. Marco-Galindo, J. Minguillón, D. García-Solórzano, y T. Sancho-Vinuesa, "¿Quién tropieza dos veces con la misma piedra en una asignatura inicial de programación?" en *Actas de las Jornadas sobre Enseñanza Universitaria de la Informática (JENUI)*, vol. 6, 2021, pp. 59–66.
- [6] Y. Lee y J. Choi, "A review of online course dropout research: Implications for practice and future research," *Educational Technology Research and Development*, vol. 59, no. 5, pp. 593–618, 2011.
- [7] A. Robins, "Learning edge momentum: a new account of outcomes in CS1," *Computer Science Education*, vol. 20, no. 1, pp. 37–71, 2010. [Online]. Available: <https://doi.org/10.1080/0893401003612167>.
- [8] A. Luxton-Reilly, *Learning to Program is Easy*. New York, NY, USA: Association for Computing Machinery, 2016, p. 284–289. [Online]. Available: <https://doi.org/10.1145/2899415.2899432>.
- [9] A. Petersen, M. Craig, y D. Zingaro, "Reviewing cs1 exam question content," en *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 631–636. [Online]. Available: <https://doi.org/10.1145/1953163.1953340>.
- [10] C. Ott, B. McCane, y N. Meek, "Mastery learning in cs1—an invitation to procrastinate?: Reflecting on six years of mastery learning," en *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*, 2021, pp. 18–24.
- [11] T. Jenkins, "On the difficulty of learning to program," en *Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences*, 2002, pp. 53–58.
- [12] B. N. Miller y D. L. Ranum, "Teaching an introductory computer science sequence with python," en *Proc. 38th Midwest Instructional and Computing Symposium, Eau Claire, Wisconsin, USA*, 2005.
- [13] A. Petersen, M. Craig, J. Campbell, y A. Tafliovich, "Revisiting why students drop CS1," en *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*, ser. Koli Calling '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 71–80. [Online]. Available: <https://doi.org/10.1145/2999541.2999552>.
- [14] D. Teague y R. Lister, "Longitudinal think aloud study of a novice programmer," en *Proceedings of the Sixteenth Australasian Computing Education Conference - Volume 148*, ser. ACE '14. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2014, pp. 41–50. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2667490.2667495>.
- [15] K. J. Whittington, D. P. Bills, y L. W. Hill, "Implementation of alternative pacing in an introductory programming sequence," en *Proceedings of the 4th Conference on Information Technology Curriculum*, ser. CITC4 '03. New York, NY, USA: Association for Computing Machinery, 2003, pp. 47–53. [Online]. Available: <http://doi.acm.org/10.1145/947121.947132>.
- [16] H. Lowe y A. Cook, "Mind the gap: Are students prepared for higher education?" *Journal of Further and Higher Education*, vol. 27, no. 1, pp. 53–76, 2003.
- [17] S. N. Liao, S. Valstar, K. Thai, C. Alvarado, D. Zingaro, W. G. Griswold, y L. Porter, *Behaviors of Higher and Lower Performing Students in CS1*. New York, NY, USA: Association for Computing Machinery,



- 2019, p. 196–202. [Online]. Available: <https://doi.org/10.1145/3304221.3319740>.
- [18] N. Hawi, "Causal attributions of success and failure made by undergraduate students in an introductory-level computer programming course," *Computers & Education*, vol. 54, no. 4, pp. 1127–1136, 2010. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0360131509003108>.
- [19] P. Kinnunen y L. Malmi, "Why students drop out CS1 course?" en *Proceedings of the second international workshop on Computing education research*, 2006, pp. 97–108.
- [20] U. Nikula, O. Gotel, y J. Kasurinen, "A motivation guided holistic rehabilitation of the first programming course," *ACM Trans. Comput. Educ.*, vol. 11, no. 4, nov 2011. [Online]. Available: <https://doi.org/10.1145/2048931.2048935>.
- [21] F. B. Tek, K. S. Benli, y E. Deveci, "Implicit theories and self-efficacy in an introductory programming course," *IEEE Transactions on Education*, vol. 61, no. 3, pp. 218–225, 2018.
- [22] J. Gorson y E. O'Rourke, "Why do cs1 students think they're bad at programming? Investigating self-efficacy and self-assessments at three universities," en *Proceedings of the 2020 ACM Conference on International Computing Education Research*, ser. ICER '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 170–181. [Online]. Available: <https://doi.org/10.1145/3372782.3406273>.
- [23] P. Kinnunen y B. Simon, "My program is ok – am I? Computing freshmen's experiences of doing programming assignments," *Computer Science Education*, vol. 22, no. 1, pp. 1–28, 2012. [Online]. Available: <https://doi.org/10.1080/08993408.2012.655091>.
- [24] A. Forte y M. Guzdial, "Motivation and nonmajors in computer science: identifying discrete audiences for introductory courses," *IEEE Transactions on Education*, vol. 48, no. 2, p. 248–253, May 2005.
- [25] M. Guzdial y A. E. Tew, "Imagining inauthentic legitimate peripheral participation: An instructional design approach for motivating computing education," en *Proceedings of the Second International Workshop on Computing Education Research*, ser. ICER'06. New York, NY, USA: Association for Computing Machinery, 2006, p. 51–58. [Online]. Available: <https://doi.org/bibliotecauoc.idm.oclc.org/10.1145/1151588.1151597>.
- [26] J. Sheard y D. 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*, ser. ITiCSE'98. New York, NY, USA: ACM, 1998, p. 223–227. [Online]. Available: <https://doi.org/bibliotecauoc.idm.oclc.org/10.1145/282991.283550>.
- [27] A. J. Gomes, A. N. Santos, y A. J. 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*, ser. ITiCSE '12. New York, NY, USA: ACM, 2012, p. 132–137. [Online]. Available: <https://doi.org/10.1145/2325296.2325331>.
- [28] N. G. Fonseca, L. Macedo, y A. J. Mendes, "CodeInsights: Monitoring programming students' progress," en *Proceedings of the 17th International Conference on Computer Systems and Technologies*, ser. CompSysTech '16. New York, NY, USA: ACM, 2016, p. 375–382. [Online]. Available: <https://doi.org/bibliotecauoc.idm.oclc.org/10.1145/2983468.2983492>.
- [29] K. Wood, D. Parsons, J. Gasson, y P. Haden, "It's never too early: Pair programming in CS1," en *Proceedings of the 15th Australasian Computing Education Conference*, ser. ACE'13, vol. 136. AUS: Australian Computer Society, Inc., 2013, p. 13–21.
- [30] M.-J. Marco-Galindo, J. Minguillón, y T. Sancho-Vinuesa, "Análisis de la progresión de los estudiantes en una asignatura introductoria a la programación mediante redes bayesianas," en *Actas de las Jornadas sobre Enseñanza Universitaria de la Informática (JENUI)*, vol. 5. Asociación de Enseñantes Universitarios de la Informática (AENUI), 2020, pp. 69–76.
- [31] M.-J. Marco-Galindo y J. Minguillón, "La evaluación formativa como factor decisivo en el aprendizaje online. intervención en una asignatura inicial de programación," en *Actas del VI Congreso Internacional sobre Aprendizaje, Innovación y Cooperación (CINAIC)*, 2021, p. 677–681.
- [32] J. Minguillón, J. Conesa, M. E. Rodríguez, y F. Santanach, "Learning analytics in practice: Providing e-learning researchers and practitioners with activity data," en *Frontiers of Cyberlearning*. Springer, 2018, pp. 145–167.
- [33] G. Kanaparan, R. Cullen, D. Mason et al., "Effect of self-efficacy and emotional engagement on introductory programming students," *Australasian Journal of Information Systems*, vol. 23, 2019.
- [34] Y. Qian y J. Lehman, "Students' misconceptions and other difficulties in introductory programming: A literature review," *ACM Transactions on Computing Education*, vol. 18, no. 1, Oct. 2017. [Online]. Available: <https://doi.org/10.1145/3077618>.
- [35] J. Sorva, *Visual program simulation in introductory programming education*. Aalto University, 2012.
- [36] A. Vihavainen, J. Airaksinen, y C. 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*, ser. ICER '14. New York, NY, USA: ACM, 2014, p. 19–26. [Online]. Available: <https://doi.org/10.1145/2632320.2632349>.
- [37] S. P. Roche y N. M. Martínez, "Evaluación de entornos de programación para el aprendizaje," en *Actas de las Jornadas sobre Enseñanza Universitaria de la Informática (JENUI)*, p. 83, 2011.
- [38] A. Luxton-Reilly y A. Petersen, "The compound nature of novice programming assessments," en *Proceedings of the Nineteenth Australasian Computing Education Conference*, 2017, pp. 26–35.
- [39] J. Warren, S. Rixner, J. Greiner, y S. Wong, "Facilitating human interaction in an online programming course," en *Proc. of 45th ACM Technical Symposium on Computer Science Education*, 2014, pp. 665–670.
- [40] J. Catterall, J. Davis et al., "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*, vol. 29, no. 5, 2013.

[41] L. Porter y D. Zingaro, "Importance of early performance in cs1: two conflicting assessment stories," en *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, 2014.

[42] J. Q. Dawson, M. Allen, A. Campbell, y A. Valair, "Designing an introductory programming course to improve non-majors' experiences," en *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 2018, pp. 26–31.

**María-Jesús Marco-Galindo** Licenciada en informática por la Universitat Politècnica de Catalunya (UPC). Doctora en Educación y TIC por la Universitat Oberta de Catalunya (UOC). Desde 1999 es profesora de la UOC, donde investiga, en el marco del grupo *Learning Analytics for Innovation and Knowledge Application in Higher Education* (LAIKA), en temas relacionados con la docencia de la programación y de las competencias transversales en entornos de virtuales.

**Julià Minguillón** Doctor Ingeniero informático por la Universitat Autònoma de Barcelona (UAB). Desde 2001 es profesor de la UOC, donde investiga, en el marco del grupo LAIKA, en temas relacionados con el análisis de los estudiantes en entornos virtuales, analíticas de aprendizaje y el uso de visualizaciones de datos interactivas para dar soporte a docentes y estudiantes.

**David García-Solórzano** Doctor por la Universitat Oberta de Catalunya (UOC) en 2013 e ingeniero informático por la Universitat Ramon Llull (URL). Desde 2008 es profesor de la UOC donde investiga, en el marco del grupo LAIKA, en temas relacionados con *learning analytics*, visualización de datos, aprendizaje autorregulado y didáctica de la programación en entornos online.

**Teresa Sancho-Vinuesa** Doctora Ingeniera electrónica por la Universitat Ramon Llull (URL) y licenciada en matemáticas por la Universitat de Barcelona. Desde 1998 es profesora de la UOC, donde investiga, en el marco del grupo LAIKA, en temas de educación matemática en línea, principalmente, mediante el uso de las analíticas de aprendizaje.