

# Análisis de la progresión de los estudiantes en una asignatura introductoria a la programación mediante redes bayesianas

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

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

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

## Resumen

Una metodología adecuada para la adquisición de competencias de programación se basa en el trabajo continuado de conceptos teóricos, combinado con la realización de ejercicios prácticos. Es importante que la curva de aprendizaje permita a los estudiantes ir avanzando de forma progresiva, incluyendo elementos que promuevan la reflexión sobre su aprendizaje. Este artículo analiza la trayectoria seguida por los estudiantes en cuanto a la realización de los ejercicios prácticos de evaluación continua, organizados como una secuencia que combina ejercicios obligatorios para superar la asignatura, con otros optativos que pueden decidir entregar o no a lo largo del semestre. Dicho análisis se ha llevado a cabo mediante el uso de redes bayesianas. Los resultados muestran que el perfil de los estudiantes es poco determinante para la superación de la asignatura, pero sí lo es su nivel de seguimiento de la primera actividad.

## Abstract

To foster the acquisition of computer programming competencies, combining theoretical concepts with practical exercises in a continuous sequence seems to be an appropriate methodology. It is important that the learning curve allows students to advance in a progressive manner, including elements that promote reflection. This paper analyzes learners' paths with respect to the practical exercises, which are organized as a sequence combining mandatory and optional exercises, so learners can decide which exercises they want to submit along the semester. Such analysis was performed using Bayesian networks. Results show that learners's profile seems to be of no relevance to determine their performance, but it is for their engagement level in the first exercise.

## Palabras clave

Redes bayesianas, Programación, Retorno personalizado, Evaluación formativa, Aprendizaje práctico

## 1. Introducción

La enseñanza de programación en la educación superior es un tema que despierta mucho interés, tal y como demuestra un exhaustivo trabajo reciente, donde se analizan 1.666 trabajos [13]. Dos de los aspectos más estudiados son cómo se aprende el primer lenguaje de programación y cómo mejorar dicho aprendizaje. Y, a pesar de la extensa literatura sobre el tema y de décadas de experiencia e investigación, muchas preguntas siguen aún abiertas. Así, la literatura refleja un acuerdo generalizado sobre el hecho de que aprender a programar es un proceso difícil para la mayoría de 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 [26]. A la vez, se considera que uno de los elementos que más inciden positivamente en la eficacia del aprendizaje son las actividades prácticas de programación. Los aspectos conceptuales de la algorítmica deben estar anclados en una amplia experiencia práctica a través de una buena estrategia de actividades de laboratorio que permita a los estudiantes ir trabajando, primero una a una para después ir integrándolas progresivamente [26]. Por otro lado, cada vez más estudiantes que no cursan una ingeniería en informática deciden hacer cursos de introducción a la programación, ya sea como complemento de formación o por interés personal [22]. Ello causa una alta diversidad de perfiles en una misma aula [14] y la adaptación de actividades a otros contextos profesionales [8].

Para analizar la trayectoria seguida por los estudiantes en relación al seguimiento de las actividades propuestas en un curso inicial de programación, se han utilizado redes bayesianas [7]. Dichas redes permiten estimar las probabilidades condicionadas entre diferentes sucesos encadenados en el tiempo y un objetivo final, en este caso relacionado con la realización de la actividad práctica obligatoria para superar la asignatura. Concretamente, las preguntas de investigación que aborda este trabajo son las siguientes:

- RQ1: ¿Existe alguna relación entre el perfil de los

estudiantes y su grado de participación (entendida como *engagement*) en la asignatura?

- RQ2: ¿Qué decisiones tomadas por los estudiantes determinan su seguimiento de las actividades de evaluación continua?
- RQ3: ¿Es adecuado el retorno proporcionado a los estudiantes tal y como está previsto?

Este artículo se organiza de la siguiente manera: la sección 2 presenta trabajos previos relacionados con la enseñanza de la programación. La sección 3 describe los aspectos metodológicos de este trabajo. La sección 4 responde las preguntas de investigación y, finalmente, la sección 5 resume las principales aportaciones de este artículo y presenta futuras líneas investigación.

## 2. La enseñanza de programación en cursos iniciales

Aunque la investigación en la enseñanza de la programación *Computing Education Research* (CEdR) tiene ya 50 años de recorrido, es a partir de 2005 con la primera *Computing Education Research Conference* (ICER) cuando se consolida como un importante ámbito de investigación en educación. Uno de los temas que más interés han despertado en la CEdR como se refleja en la reciente estado del arte de [15] es analizar cómo se aprende a programar en un curso inicial de programación un estudiante novel, (conocido como CS1 *Computer Science 1*).

La experiencia demuestra que aprender a programar es difícil. De hecho, los cursos introductorios de programación tienen, en general, altas tasas de abandono y un índice de superación menor que el de otras asignaturas de primer curso. Y, aunque la literatura sobre el tema es extensa y relevante, muchas preguntas continúan abiertas y sin un consenso claro: ¿Por qué es tan difícil aprender a programar para algunos estudiantes y en cambio resulta más fácil para otros?, ¿Cuál es el mejor lenguaje de programación para aprender a programar?, ¿Qué factores permiten determinar el éxito de un estudiante en un curso inicial de programación? o ¿Qué contenidos se deberían tener un curso introductorio? entre otras.

La literatura describe de manera persistente y generalizada importantes tasas de abandono e índices bajos de superación, como se describe en [26], que muestra la similitud de resultados entre 51 instituciones, variando ligeramente en función del país pero no del lenguaje de programación, resultados que no mejoran con el tiempo. Trabajos similares se detallan en [13]. También se estudia si, una vez superado un curso inicial, los estudiantes han alcanzado el nivel de programación esperado. Diversos estudios como el de [2] muestran que muchos estudiantes aprueban el primer curso sin con-

seguir la soltura suficiente para crear programas sencillos lo que conlleva dificultades en los cursos posteriores, aumentando la percepción de que programar es difícil en todos los niveles.

Resulta más complejo aún si atendemos a diversas investigaciones como es el caso de [4] que a lo largo de los años describen una distribución bimodal de los resultados, con un grupo de estudiantes con tasas más bajas de la media y otro grupo, por contra, con resultados más altos, con pocos estudiantes en el rango medio. Así que, se dibujan habitualmente dos poblaciones muy distintas en referencia a los resultados. Esto abre un amplio abanico de investigaciones para analizar qué factores tanto del perfil del estudiante como del proceso de aprendizaje inciden en que éste caiga en uno u otro grupo y, si es posible, de algún modo, predecir el resultado. Parece ser que no hay ningún factor determinante que permita predecir el éxito [2], pero sí que está claro que, una gran parte de estudiantes que aprenden a programar no lo hacen con la intención de convertirse en programadores profesionales, sino que únicamente necesitan saber hacer programas sencillos para su trabajo diario. En este contexto, parece lógico pensar que en este caso no es adecuado un curso inicial de programación estándar [5], y que hay que proporcionar opciones más acordes a sus necesidades [11].

Otro de los aspectos más estudiados y debatidos es cuál debería ser el objetivo y los contenidos de un curso inicial de programación. Un estudio iniciado en 2003 y continuado recientemente [20] los caracteriza según tres dimensiones (conocimiento, estrategias y modelos) aplicadas a las tres etapas de desarrollo de un programa: diseño, implementación y prueba. Y, aunque categorizaciones hay muchas y diversas cada cual con bondades y carencias, en todas se vislumbra una necesaria e imprescindible combinación entre conocimientos básicos (saber) y prácticos (saber aplicar).

Desde la perspectiva del estudiante, muchos estudios analizan qué contenidos le resultan más difíciles y la carga cognitiva que representan [24], así como las estrategias de enseñanza-aprendizaje más adecuadas para reducirla [6] entre las que se incluye el retorno individualizado, que es importante que se dé en los momentos cruciales donde puede ser más efectivo [17, 19]. 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 [18]. Relacionado con esto último, es 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. En este sentido, son rele-

vantes los estudios de [9] 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.

### 3. Metodología

#### 3.1. Contexto

La asignatura "Fundamentos de programación", obligatoria en los grados de Ingenierías Informática (INF) e Ingeniería de Tecnologías de Telecomunicación (TEL) de la Universitat Oberta de Catalunya es también un complemento de formación en algunos másteres especializados y asignatura libre dentro del programa Ingeniería Informática de la universidad. Es, pues, una asignatura con un perfil de estudiantes muy heterogéneo, aunque todos la cursan del mismo modo y en las mismas aulas.

Es una asignatura introductoria a la programación, que parte de una base inicial de fundamentos algorítmicos que se combina, ya desde un inicio, con la práctica de ejercicios sencillos de programación en lenguaje C. Así, se avanza de manera secuencial y progresiva con el objetivo de conseguir un dominio básico a la vez que sólido de la programación en C. El diseño instruccional se sustenta en tres elementos fundamentales: el aula virtual combinada con un laboratorio virtual de programación en C donde se trabajan aspectos de instalación y uso del IDE (instalación, edición y ejecución); el entorno de programación con el que los estudiantes realizan las prácticas, que en este caso se proporciona a través de una máquina virtual; y el modelo de evaluación.

La evaluación es continua y está formada por una secuencia de actividades que combinan teoría (diseño de algoritmos) y práctica (programas en C). Se trata de una evaluación continua que, a la vez, es formativa dado que se proporciona un retorno al estudiante de las diferentes actividades mediante una calificación acompañada de comentarios individuales así como la publicación de la solución de cada una de ellas.

En concreto, se proponen ocho actividades optativas (A1, A2, ..., A8) como pruebas evaluación continua que se combinan con la realización de un ejercicio práctico de programación más complejo y obligatorio que integra todos los contenidos del curso (PR). Más en detalle, la secuencia previa de actividades es la siguiente: (A1)-Tipos básicos de datos, (A2)-Expresiones, (A3)-Estructura alternativa, (A4)-Estructura iterativa, (A5)-Tuplas, (A6)-Modularidad, (A7)-Paso de parámetros, (A8)-Tablas y (PR)-Integración de todo lo aprendido en un programa de más envergadura. Las pruebas o actividades semanales están alineadas con la práctica y forman parte del

mismo contexto, de tal modo que los ejercicios de programación semanales acaban siendo después parte de la práctica. Esta práctica es de realización obligatoria y es también indispensable superarla para aprobar la asignatura. De ahí que sea importante conocer qué factores inciden en que un estudiante llegue a realizar la práctica correctamente.

El profesor acompaña y guía al estudiante durante su aprendizaje y finalmente lo evalúa y le retorna el *feedback* de cada una de las actividades. La publicación de la solución se hace semanalmente, y la calificación y el retorno individuales se facilitan en dos momentos concretos del semestre (R1 y R2): justo después de la cuarta y de la octava actividad, respectivamente.

#### 3.2. Datos

Los datos referentes a los estudiantes de la asignatura Fundamentos de programación se extraen del *Learning Record Store* institucional [16], involucrando un total de 1,043 registros de tres semestres académicos consecutivos (2017/2, 2018/1 y 2018/2). Cada registro describe la experiencia de un estudiante en un semestre, y contiene los campos siguientes:

- Datos socio-demográficos: sexo (SEXE), grupo de edad (GE).
- Experiencia académica: grado (PROG), vía de acceso (VIA), si el estudiante es repetidor o no y qué hizo en su anterior experiencia (REPFA), si se trata de un nuevo estudiante o no (NOU), número de asignaturas matriculadas ese semestre (NASS).
- Semestre en curso: para cada actividad (An, n=1..8) se conoce si el estudiante la entrega o no y la nota obtenida. La variable objetivo (PR) es el hecho de superar la parte práctica obligatoria, indispensable para superar la asignatura. También se dispone de dos indicadores (R1 y R2) que representan el retorno que recibe el estudiante respecto a los 4 primeros y los 4 siguientes ejercicios propuestos, respectivamente.

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 de 31-40 años, seguido de 26-30 años). La mayoría de estudiantes la cursan durante su primer semestre (58.49 %) y lo hacen en el grado en Informática (INF, 69.89 %) pero también hay estudiantes del grado en Telecomunicaciones (TEL, 15.15 %) y otros que la cursan como complementos de formación de un máster (CFM, 10.26 %) o como asignatura libre (ALT, 4.70 %). La mayoría de estudiantes cursan al mismo tiempo entre 2 y 3 asignaturas, típico de estudiantes a tiempo parcial. Esta asignatura tiene históricamente un bajo rendimiento, por lo que el número de estudiantes que la repiten es elevado (un 17.45 %), y de hecho un

42.03 % de los estudiantes que no están en su primer semestre son repetidores. Por lo tanto, la diversidad de perfiles presentes en la misma asignatura es evidente.

### 3.3. Redes bayesianas

Una red bayesiana [7] es un grafo acíclico dirigido que representa las asociaciones existentes entre variables de acuerdo a algún criterio preestablecido, generalmente relacionado con la probabilidad condicional entre una variable y sus antecedentes o predictoras.

El uso de redes bayesianas para el análisis ofrece muchas posibilidades, como por ejemplo obtener una representación gráfica, explícita e interpretable de un conocimiento incierto presente en un conjunto de datos. Por ello, han sido utilizadas para realizar predicciones en escenarios muy diferentes [1], siendo una de las herramientas más usadas en el ámbito de la minería de datos aplicada a la educación [10, 12], principalmente por su potencial interpretación de acuerdo a hipótesis preestablecidas. Por ejemplo, un trabajo que usa inferencia bayesiana para predecir el éxito de los estudiantes en un curso introductorio a la programación es [3], donde los autores comparan diferentes métodos de clasificación usando los resultados obtenidos por los estudiantes en los primeros ejercicios planteados.

Es importante destacar, sin embargo, que el objetivo de este trabajo no es predecir el resultado de un estudiante en la asignatura en función de sus resultados parciales, sino explorar las posibles estrategias que siguen los estudiantes a lo largo del semestre: cuándo deciden entregar o no una de las actividades propuestas, y si siguen la secuencia de actividades propuesta.

## 4. Resultados

### 4.1. Seguimiento de la asignatura

El objetivo principal del trabajo es relacionar las entregas que hacen los estudiantes de las actividades propuestas con el resultado obtenido en la actividad obligatoria. Por este motivo se analizaron, en primer lugar, los resultados obtenidos por los estudiantes. El Cuadro 1 muestra el número y porcentaje de estudiantes que supera la actividad obligatoria por semestre. Se puede observar cierta variabilidad entre semestres aunque un test  $\chi^2$  no muestra asociación entre el semestre académico y el hecho de superar o no la asignatura ( $p = 0,2774$ ,  $V = 0,0496$ ). Los estudiantes que superan la actividad obligatoria son una ligera mayoría (56.2 %), por lo que no se ha considerado balancear el conjunto de datos respecto a esta variable.

La Figura 1 muestra las diferencias por semestre y la probabilidad de entregar cada actividad propuesta.

Semestre	Resultado		Total
	Suspende	Supera	
2017/2	120 (40.3 %)	178 (59.7 %)	298
2018/1	216 (46.2 %)	252 (53.8 %)	468
2018/2	121 (43.7 %)	156 (56.3 %)	277
<b>Total</b>	457 (43.8 %)	586 (56.2 %)	1,043

Cuadro 1: Probabilidad de superar la actividad obligatoria por semestre.

Las curvas indicadas por 1, 2 y 3 representan los resultados de los tres semestres consecutivos para los estudiantes que no superan la actividad obligatoria, mientras que las indicadas por 4, 5 y 6 representan los resultados para los que sí la superan. Se puede observar una clara diferencia entre grupos por resultado ya incluso desde la primera actividad. De hecho, en el primer retorno los estudiantes que no superan la actividad obligatoria obtienen un promedio de 1.40 puntos (sobre 4 posibles), con una desviación típica de 1.32, mientras que los estudiantes que sí la superan obtienen 3.32 puntos con una desviación de 0.68 (Welch t-test  $t = 28,287$ ,  $p < 0,001$ ). En el segundo retorno esta diferencia es aún más acusada, 1.97 vs 6.38 (sobre 8 posibles) con unas desviaciones típicas de 2.09 y 1.39 respectivamente (Welch t-test  $t = 38,942$ ,  $p < 0,001$ ). Igualmente, el porcentaje de estudiantes que no superan la actividad obligatoria cae de forma más o menos constante en cada actividad, mientras que los estudiantes que sí la superan se mantienen constantes hasta la actividad A5, donde empiezan a relajarse y participar en menor porcentaje. De hecho, es en la actividad A6 donde se empiezan a trabajar los conceptos más complejos y donde reciben el primer retorno.

### 4.2. Modelos construidos

Se han construido dos modelos diferentes, con objetivos complementarios. El primero pretende establecer una relación entre las variables socio-demográficas y la experiencia académica (datos del estudiante conocidos antes de iniciar la asignatura) con el hecho de iniciar la secuencia de actividades propuesta, para poder responder RQ1. El segundo modelo pretende establecer la relación causal entre el hecho de hacer o no cada una de las actividades propuestas con las siguientes, siguiendo el calendario de la asignatura, y culminando con la realización de la actividad obligatoria para su superación, usando también las variables identificadas como relevantes en el primer modelo, para responder RQ2 y RQ3. En ambos casos se usó un algoritmo de

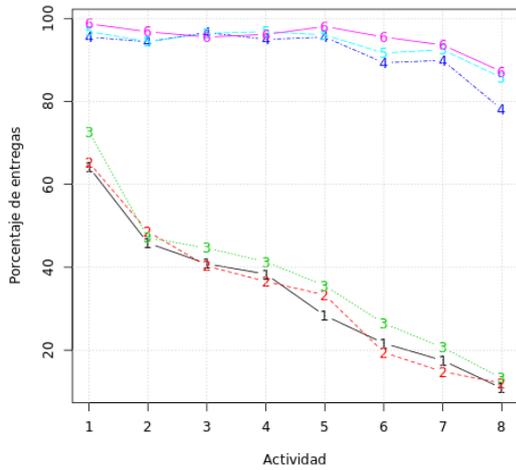


Figura 1: Porcentaje de entregas por semestre y resultado final.

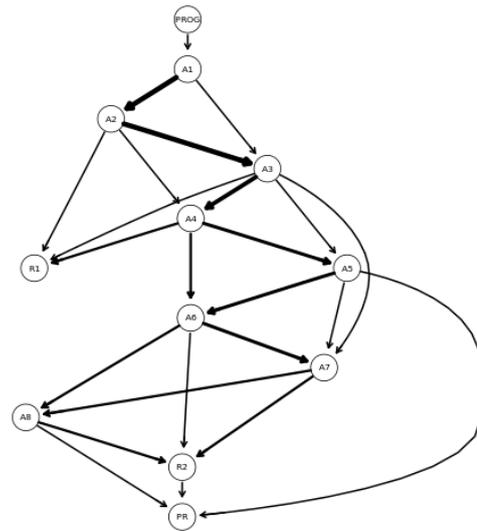


Figura 3: Red bayesiana con la actividad obligatoria como objetivo.

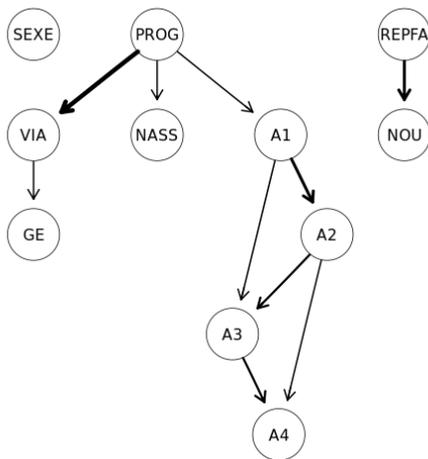


Figura 2: Red bayesiana con datos previos al inicio de la asignatura.

*hill climbing* [25] junto con el criterio de información bayesiano (BIC) para extraer la estructura subyacente de la red a partir del conjunto de datos.

La Figura 2 muestra la primera red bayesiana creada mediante el package de R 'bnlearn' [23]. Cada nodo es una variable y el grueso de cada arco entre dos nodos representa la fuerza relativa de la relación entre ellos. Se puede observar que la única variable que incide sobre el hecho de realizar la primera actividad propuesta es PROG (el grado que está cursando el estudiante), aunque se trata de una relación débil. Por el contrario, el perfil socio-demográfico del estudiante parece no tener ninguna relación causal con el hecho de realizar la primera actividad propuesta, respondiendo así a la primera pregunta de investigación RQ1. El hecho que la variable SEXE (el sexo del estudiante) esté tan

desbalanceada explica seguramente que quede aislada del resto. Curiosamente, la variable NASS (número de asignaturas que cursa el estudiante) y que *a priori* podría afectar a su capacidad de participación tampoco resulta relevante.

De hecho, los resultados académicos corroboran esta constatación, dado que el rendimiento (entendido como superación de la asignatura sobre matriculados) de los estudiantes del grado de Informática es siempre muy superior al de los estudiantes del grado de Tecnologías de Telecomunicación. Para estos últimos, la programación no es una competencia objetivo y, aunque la cursan obligatoriamente, no se encuentra entre sus intereses principales, al contrario de lo que ocurre con los estudiantes de informática. Esta diferencia de expectativas e intereses puede explicar la mayor o menor predisposición a empezar a realizar las actividades al inicio del curso.

A partir de aquí y de cara a la construcción del segundo modelo, se añadió la variable PROG como único antecedente de la primera actividad. La Figura 3 muestra el segundo modelo construido usando la misma metodología. Este modelo revela diversos hechos relevantes sobre las relaciones estimadas entre las actividades propuestas, el retorno obtenido en forma de calificación y el hecho de superar o no la actividad obligatoria. Como era de esperar según el modelo anterior, la variable PROG solamente influye en la realización de la primera actividad A1.

Se puede observar que los arcos más gruesos, que representan las posibles relaciones causales más importantes, se dan entre actividades  $A_n$  ( $n=1..8$ ) consecutivas, decreciendo a lo largo del tiempo. Cada actividad  $A_n$  influye en las dos siguientes, excepto A3 que tam-

bién precede a A7 y obviamente A7 y A8 que son las últimas actividades. Una inspección de los arcos muestra que el peso de A3 a A7 es tan solo de 0.3411, un valor de un orden de magnitud o dos inferior al resto, por lo que se puede considerar residual. Las primeras actividades son las más importantes a la hora de conseguir que el estudiante se aproveche del diseño del curso y no quede descolgado, adquiriendo cierta inercia.

Por otra parte, el primer retorno obtenido (R1) después de las cuatro primeras actividades no parece afectar al hecho de que los estudiantes decidan continuar o no con el resto de actividades, ni al hecho de superar o no la actividad obligatoria. Esto puede indicar que este retorno no se da en el momento adecuado. En cambio, el segundo retorno obtenido (R2) sí que permite saber al estudiante si está en condiciones o no de realizar la actividad obligatoria de forma satisfactoria (o reforzar su percepción al respecto), aunque este hecho ya se produce seguramente muy tarde en el curso y no permite al estudiante replantearse nada.

### 4.3. Estimación con evidencias firmes

Una de las posibilidades que ofrecen las redes bayesianas es la estimación de la probabilidad de un evento en función de unos antecedentes conocidos (también llamados evidencias firmes), de forma que es posible plantearse escenarios del tipo “¿Y si...?”. Para ello se usa una simulación de Monte Carlo con  $10^6$  iteraciones para cada una de las condiciones a validar.

Por ejemplo, el Cuadro 2 muestra la probabilidad de realizar la primera actividad en función del grado cursado, estimada a partir del modelo construido en la Figura 3. Se puede observar que hay una diferencia substancial entre grados, especialmente en los estudiantes de Telecomunicaciones y los que la cursan como asignatura libre, con un porcentaje claramente inferior al promedio ( $\chi^2 = 29,547, p < 0,001$ ).

A1	Grado			
	INF	TEL	CFM	ALT
NO	0.1454	0.2468	0.0654	0.3469
SÍ	0.8546	0.7532	0.9346	0.6531

Cuadro 2: Probabilidad de realizar la primera actividad según el grado.

Por el contrario, para cada estudiante, pasada la primera actividad, el grado que está cursando deja de tener relación con el hecho de superar la actividad obligatoria, tal y como muestra el Cuadro 3 ( $\chi^2 = 0,0610, p = 0,9961$ ), de acuerdo al segundo modelo construido, mostrado en la Figura 3.

Por lo tanto, la primera actividad consigue “enganchar” a los estudiantes independientemente del grado

A1	Grado			
	INF	TEL	CFM	ALT
Prev.	0.5685	0.5190	0.6062	0.4733
A1=NO	0.1510	0.1512	0.1556	0.1489
A1=SÍ	0.6372	0.6347	0.6385	0.6394

Cuadro 3: Probabilidad de superar la actividad obligatoria por grado según la realización de la primera actividad.

An	No entrega An	Sí entrega An
A1	0.1659	0.6364
A2	0.1204	0.6969
A3	0.0877	0.7361
A4	0.0749	0.7612

Cuadro 4: Probabilidad de superar la actividad obligatoria según las actividades iniciales realizadas.

del que provengan. De la misma forma es posible estimar la probabilidad de superar la actividad obligatoria en función de las primeras actividades  $A_n$  entregadas, en orden consecutivo. El Cuadro 4 muestra cómo esta probabilidad se reduce conforme aumenta el número de actividades consecutivas no entregadas. El objetivo es estimar en qué momento se debería realizar una intervención para intentar recuperar los estudiantes que se han descolgado de las actividades propuestas.

Se puede observar que conforme aumenta el número de actividades consecutivas no entregadas, la probabilidad de superar la actividad obligatoria disminuye. Resulta interesante que después de no entregar las tres primeras actividades, la probabilidad ya no cambia considerablemente, por lo que podría pensarse que después de la tercera actividad ya es demasiado tarde para recuperar a los estudiantes que no han realizado ninguna de ellas. En cambio, para los estudiantes que van entregando todas las actividades, la probabilidad de superar la actividad obligatoria va aumentando ligera pero constantemente. Es decir, el hecho de no entregar las primeras actividades es mucho más determinante que el hecho de entregarlas.

Este resultado es coherente con los obtenidos en [21] en asignaturas de cálculo y estadística respectivamente. En ambos casos la metodología docente también se basa en la actividad continuada del estudiante, en este caso mediante cuestionarios con evaluación y retorno automático semanales. En estudio presenta, además, una asociación significativa entre el resultado del examen final y la realización de dichos cuestionarios.

### 4.4. Limitaciones

Las principales limitaciones de este trabajo son causadas por la falta de robustez de las redes bayesianas,

especialmente frente a los posibles *outliers* presentes en el conjunto de datos, así como a cambios en la distribución de los mismos. No obstante, el ratio entre el tamaño del conjunto de datos y el tamaño de la red Bayesiana es aceptable, dado el reducido número de nodos de la misma. Por otra parte, la red bayesiana está completamente determinada por el diseño del curso y por lo tanto su uso generalizado en otras asignaturas no es posible. Aun así, es factible reproducir la misma metodología en escenarios diferentes donde haya una secuencia de eventos significativos.

## 5. Conclusiones

Las redes bayesianas han sido extensivamente usadas para la clasificación de estudiantes en situación de riesgo y la predicción de su rendimiento académico, en función de su perfil socio-demográfico y su participación en las actividades propuestas [12]. En este artículo se han utilizado también para analizar cómo el diseño de una asignatura de introducción a la programación y el retorno recibido ayuda o influye en la trayectoria de aprendizaje y seguimiento del curso por parte de los estudiantes, obteniendo conocimiento relevante.

Por un lado, en sintonía con lo que ya apuntaba Carter [2] sobre la no existencia clara de un factor o combinación de factores que pueda predecir el éxito de un estudiante en un primer curso de programación, se ha constatado que la única condición del perfil del estudiante que tiene una cierta influencia en la realización de la primera actividad optativa es el programa que está cursando. El estudio ha constatado diferencias significativas entre los estudiantes matriculados en ingeniería informática y los del resto de programas. Esto coincide con los resultados de diversos estudios recientes [5, 14], los cuales apuntan que los estudiantes de otras carreras (no de informática) están habitualmente insatisfechos con los cursos de introducción a la programación pensados para estudiantes de informática. Ello redundaría en el hecho que se involucra y participa menos, y obtiene peores resultados.

Por otro lado, el hecho de realizar o no la primera actividad optativa es un buen indicador de si acabará entregando o no la actividad práctica, obligatoria y decisiva para superar la asignatura. Así, resulta crucial que los estudiantes realicen esta primera actividad. La participación de los estudiantes puede verse como un comportamiento con beneficios acumulativos, donde lo más importante parece ser “empezar”, es decir, implicarse desde el principio y realizar la primera actividad, que es la que inicia la secuencia de actividades que conducen a la realización de la actividad práctica obligatoria final y decisiva para superar la asignatura. Este resultado confirma los que se indica en la literatura [6, 18] en relación con la importancia del compromi-

so con la asignatura (*engagement*) justo en las primeras semanas del curso.

Sin embargo, el retorno tal y como está planteado parece no tener demasiada incidencia en el hecho de realizar la actividad obligatoria y, en consecuencia, superar la asignatura. Tal y como describen [17, 19], no se trata de proporcionar más retorno, sino que esté relacionado con el nivel de participación, sea más efectivo y se produzca en el momento más adecuado.

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 una intervención que mejore el retorno y acompañamiento 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. Así mismo, y de cara a ofrecer una formación adecuada a los diferentes perfiles, es necesario investigar a fondo las causas de la diferencia de comportamiento según el programa cursado en relación con la participación en la primera actividad. Ello proporcionaría la justificación necesaria para diseñar una asignatura introductoria a la programación específica para cada uno de los distintos programas, tal y como se apunta en [11].

## Referencias

- [1] Concha Bielza y Pedro Larrañaga. Discrete bayesian network classifiers: a survey. *ACM Computing Surveys (CSUR)*, 47(1):5, 2014.
- [2] Adam S. Carter, Christopher D. Hundhausen, y Olusola Adesope. 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, 2017.
- [3] Evandro B Costa, Balduino Fonseca, Marcelo Almeida de Araújo Santana, Fabrísia Ferreira, y Joilson Rego. Evaluating the effectiveness of educational data mining techniques for early prediction of students’ academic failure in introduc-

- tory programming courses. *Computers in Human Behavior*, 73:247–256, 2017.
- [4] Saeed Dehnadi y Richard Bornat. The camel has two humps (working title). *Middlesex University, UK*, 2006.
- [5] Andrea Forte and Mark Guzdial. Motivation and non-majors in computer science. *nd): n. pag. Co-web. cc. gatech. edu. Georgia Tech. Web*, 1, 2016.
- [6] Rashina Hoda y Peter Andreae. It's not them, it's us! why computer science fails to impress many first years. En *Proceedings of the Sixteenth Australasian Computing Education Conference-Volume 148*, páginas 159–162, 2014.
- [7] Finn V. Jensen. *An introduction to Bayesian networks*, volume 210. UCL press London, 1996.
- [8] Brian Jiménez-García, Cristina Pérez Solà, Pau Andrio Balado, and M. Jesús Marco Galindo. Aprendiendo a programar. nuevos retos, nuevas propuestas. En *Actas de las XXV JENUI*, volume 4, páginas 71–78, 2019.
- [9] Geetha Kanaparan, Rowena Cullen, y David DM Mason. Self-efficacy and engagement as predictors of student programming performance. En *PACIS*, páginas 282, 2013.
- [10] Tanja Käser, Severin Klingler, Alexander G Schwing, y Markus Gross. Dynamic bayesian networks for student modeling. *IEEE Transactions on Learning Technologies*, 10(4):450–462, 2017.
- [11] Andrew J Ko, Robin Abraham, Laura Beckwith, Alan Blackwell, Margaret Burnett, Martin Erwig, Chris Scaffidi, Joseph Lawrance, Henry Lieberman, Brad Myers, et al. The state of the art in end-user software engineering. *ACM Computing Surveys (CSUR)*, 43(3):1–44, 2011.
- [12] Carmen Lacave, Ana I. Molina, Miguel A. Redondo, y Manuel Ortega. Redes bayesianas para identificar perfiles de estudiante. aplicación al estudio del abandono de las titulaciones de informática en la universidad de castilla-la mancha. En *Actas de las XXII JENUI*, pages 85–92. Universidad de Almería, 2016.
- [13] Andrew Luxton-Reilly, Ibrahim Albluwi, Brett A Becker, Michail Giannakos, Amruth N Kumar, Linda Ott, James Paterson, Michael James Scott, Judy Sheard, y Claudia Szabo. Introductory programming: a systematic literature review. En *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 2018.
- [14] Bruce A. Maxwell y Stephanie R. Taylor. Comparing outcomes across different contexts in cs1. En *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, 2017.
- [15] R. P. Medeiros, G. L. Ramalho, y T. P. Falcão. A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Transactions on Education*, 62(2):77–90, May 2019.
- [16] J Minguillón, J Conesa, ME Rodríguez, y F Santanach. Learning analytics in practice: Providing e-learning researchers and practitioners with activity data. En *Frontiers of Cyberlearning*, páginas 145–167. Springer, 2018.
- [17] Claudia Ott, Anthony Robins, y Kerry Shephard. Translating principles of effective feedback for students into the cs1 context. *ACM Transactions on Computing Education (TOCE)*, 16(1):1–27, 2016.
- [18] Leo Porter y Daniel Zingaro. Importance of early performance in cs1: two conflicting assessment stories. En *Proceedings of the 45th ACM technical symp. on Computer science education*, 2014.
- [19] Margaret Price, Karen Handley, y Jill Millar. Feedback: Focusing attention on engagement. *Studies in higher education*, 36(8):879–896, 2011.
- [20] Anthony Robins. Learning edge momentum: a new account of outcomes in cs1. *Computer Science Education*, 20(1):37–71, 2010.
- [21] Teresa Sancho-Vinuesa, Ramon Masià, Marc Fuertes-Alpiste, y Núria Molas-Castells. Exploring the effectiveness of continuous activity with automatic feedback in online calculus. *Computer Applications in Engineering Education*, 26(1):62–74, 2018.
- [22] Linda J Sax, Kathleen J Lehman, y Christina Zavala. Examining the enrollment growth: non-cs majors in cs1 courses. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, páginas 513–518, 2017.
- [23] Marco Scutari. Learning bayesian networks with the bnlearn r package. *arXiv preprint arXiv:0908.3817*, 2009.
- [24] Juha Sorva. *Visual program simulation in introductory programming education*. Aalto University, 2012.
- [25] Ioannis Tsamardinos, Laura E Brown, y Constantinos F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.
- [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, page 39–44, New York, NY, USA, 2014. Association for Computing Machinery.