

Evaluación de una intervención de reorganización del calendario de actividades en una asignatura inicial de programación

Alberto Gómez, Maria-Jesús Marco-Galindo, Julià Minguillón

Abstract—A key factor in online learning is an instructional design that ensures that students maintain an adequate and constant learning pace throughout the course. This is especially relevant when a fundamentally practical and progressive learning approach is required, such as in introductory programming courses. This article describes an intervention conducted in a first-year subject of the Computer Engineering degree called "Programming Fundamentals". This subject poses many challenges related to the introduction of abstract concepts, the completion of programming exercises in a specific language, and the monitoring of the pace of proposed learning activities so that students can achieve adequate learning. Based on academic results from several semesters, it was decided to make an intervention that modified the planning of learning activities to maintain motivation and learning pace throughout the semester, while reducing the time between completing the activities and receiving feedback. An analysis of the results following the change shows that more students complete the core activities, with a decrease in dropouts from continuous assessment and an increase in the number of students passing the course. Data analysis has been validated using *propensity score matching*, a method for evaluating interventions with a quasi-experimental design.

Index Terms—CS1, activity planning, formative assessment, intervention evaluation, performance analysis, Propensity Score Matching.

I. INTRODUCCIÓN

La experiencia demuestra que aprender a programar es difícil y que la enseñanza de la programación es un reto, tal y como demuestra el elevado número de publicaciones sobre el tema, como se describe, por ejemplo, en [1] donde se analizan 1666 trabajos sobre el aprendizaje de la programación. Así, la literatura refleja un acuerdo generalizado sobre el hecho de que aprender a programar es un proceso difícil para la mayoría de los estudiantes a todos

los niveles. En particular, el abandono en cursos iniciales de programación en estudios universitarios generalmente es alto y los índices de superación, bajos [2]. Y, a pesar de la extensa literatura sobre el tema y de décadas de experiencia e investigación, muchas preguntas siguen aún abiertas. ¿Por qué es tan difícil aprender a programar para muchos estudiantes?, ¿qué factores permiten determinar el éxito de un estudiante en un curso inicial de programación? Parece ser que no hay ningún factor determinante que permita predecir el éxito [3], pero sí está claro que uno de los elementos que más inciden positivamente en la eficacia del aprendizaje son las actividades prácticas de programación. Se vislumbra una necesaria e imprescindible combinación entre conocimientos básicos (saber) y prácticos (saber aplicar). Los aspectos abstractos de la algoritmia deben ponerse en práctica con la codificación en un lenguaje concreto de programación a través de una buena estrategia de actividades de laboratorio que permita a los estudiantes ir trabajándolos, primero uno a uno, para después ir integrándolos progresivamente en un proyecto de mayor envergadura [2].

Desde la perspectiva del estudiante, muchos estudios analizan qué contenidos resultan más difíciles y la carga cognitiva que conllevan [4], así como las estrategias de enseñanza-aprendizaje más adecuadas para reducirla [5], entre las que se incluye el retorno (*feedback*) individualizado, que es importante que se reciba en los momentos cruciales donde puede ser más efectivo [6]. Uno de estos momentos críticos son las primeras semanas del curso en las que hay que estar muy atento para que la experiencia inicial de cada 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 [7]. 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 y comprometiéndose con su aprendizaje sin perder el interés ni disminuir su dedicación a lo largo de todo curso. En este sentido, son relevantes los estudios de Kanaparan [8] relacionados con el *engagement* en un primer curso de programación, expresado en función de los tres indicadores

Alberto Gómez es profesor en la Universidad de Extremadura y colabora con la Universitat Oberta de Catalunya (correo electrónico: agomezma@uoc.edu).

Maria-Jesús Marco Galindo es profesora en la Universitat Oberta de Catalunya (correo electrónico: mmarcog@uoc.edu).

Julià Minguillón es profesor en la Universitat Oberta de Catalunya (correo electrónico: jminguillona@uoc.edu).

que lo determinan: el esfuerzo, la persistencia y la búsqueda de soporte.

Los objetivos concretos a corto plazo favorecen la motivación del estudiante [9]. Por eso, las actividades de evaluación a lo largo del curso mantienen el trabajo constante con los materiales, además de servir para la evaluación formativa y sumativa [10], [11]. En los estudios virtuales, donde desafortunadamente el abandono es muy alto, el diseño de una planificación de actividades debe buscar el mantenimiento de la motivación, el esfuerzo y la participación del estudiante [12].

En [13] se presenta una revisión de 32 trabajos que describen intervenciones en asignaturas de introducción a la programación y valoran los cambios que influyen más en las mejoras. Estos cambios consisten, principalmente, en aumentar la colaboración entre estudiantes, cambiar y contextualizar los contenidos, crear cursos previos, modificar el sistema de calificaciones y aumentar la tutorización. Una de las principales conclusiones de los autores de esta revisión es que, en general, todas las intervenciones mejoran los resultados si se comparan con enfoques tradicionales, aunque no hay diferencias estadísticamente significativas en la efectividad entre la mayoría de ellas.

Uno de los problemas al analizar la mejora obtenida tras una intervención es que se deben comparar resultados obtenidos en semestres distintos, con poblaciones de estudiantes diferentes. En general, se asume que las características de las poblaciones se mantienen constantes en los cursos comparados, aunque esto no tiene por qué ser así. En nuestro caso, hemos utilizado una técnica cuasiexperimental para validar la comparación de los datos de cuatro semestres distintos.

Este artículo extiende la investigación presentada y publicada en la conferencia CINAIC 2021 que fue elegido para su envío a IEEE-RITA [14]. Respecto al artículo de partida, se amplía la revisión bibliográfica y el análisis de los resultados, y se presenta la validación de la intervención mediante la técnica de diseño cuasiexperimental denominada *Propensity Score Matching*.

Este trabajo se estructura de la siguiente manera: en el apartado siguiente se describe el diseño cuasiexperimental. A continuación, en el apartado 3, se detalla la metodología seguida, con una descripción del contexto, los datos empleados y la medida usada para evaluar la intervención. En el siguiente apartado se describe la situación de partida y la intervención. El apartado 5 presenta los principales resultados obtenidos, junto con su validación. En el último apartado se describen las principales conclusiones y líneas futuras de investigación.

II. ANÁLISIS DE INTERVENCIONES

Para evaluar el efecto que ha tenido una intervención en una asignatura no basta con comparar los indicadores de resultado obtenidos por el estudiantado en los semestres previos y posteriores al cambio, ya que estos estudiantes pertenecen a

poblaciones distintas. Al comparar directamente los datos de diferentes semestres se corre el riesgo de que haya diferencias significativas en los perfiles de los estudiantes en cada semestre y, por tanto, los resultados de la evaluación estén sesgados por diferencias significativas en los conjuntos de datos y no solo por la influencia de la intervención realizada.

A. Diseño cuasiexperimental

Para poder comparar directamente los resultados de varios semestres se debería llevar a cabo un experimento aleatorio sobre la misma población, con un grupo de control en el que se hubiera mantenido la planificación anterior y otro con la nueva distribución de actividades, y un proceso aleatorio de asignación a cada grupo en cada semestre.

En campos como la medicina, el diseño experimental debe ser siempre aleatorio, donde el grupo de control y el grupo sujeto del estudio se formen siguiendo una metodología estricta que asegure que no influyen otros factores y que el estudio no pierde validez [15]. En la investigación en educación es difícil, y en algunos casos poco ético, realizar estudios aleatorios con grupos de control [16].

Para los casos en los que no se pueden llevar a cabo ensayos aleatorios, existen técnicas estadísticas cuasiexperimentales que intentan controlar, en la medida de lo posible, la influencia de otras variables distintas a la que se pretende analizar [17]. Los diseños cuasiexperimentales se basan en seleccionar elementos similares de los dos grupos de datos (el grupo de los que experimentan la intervención y el grupo de los que no), para que las dos muestras finales que se comparen se puedan considerar equivalentes y similares a las que se obtendrían con un diseño experimental aleatorio. Con estos métodos se pretende reducir el posible sesgo provocado por otras variables de confusión que intervienen en la pertenencia de los individuos a un grupo u otro. Cuando la asignación a los grupos es aleatoria, el efecto de los factores de confusión se equilibra y el sesgo deja de ser significativo.

B. Propensity Score Matching

Entre las técnicas cuasiexperimentales más utilizadas destacan los métodos de emparejamiento por índice o puntajes de propensión [18] o, en inglés, *Propensity Score Matching* (PSM) [19]. El índice de propensión (PS) de una observación es la probabilidad o propensión de formar parte o no del grupo que ha sido sometido a la intervención, calculada a partir de las variables explicativas con las que se cuenta. Si dos observaciones tienen los mismos valores en las covariables consideradas, se supone que la probabilidad de que hubieran participado en la intervención sería la misma. Si se elige una del grupo de control y otra del grupo donde se ha llevado a cabo la intervención, se obtiene un resultado similar al que hubiera producido una asignación aleatoria de las observaciones.

Generalmente, los PS se calculan mediante un modelo de regresión logística con la participación o no en la intervención como resultado y algunas variables del modelo como predictoras. Posteriormente, se emparejan los datos con PS

similares de los dos conjuntos de datos iniciales, rechazándose los elementos que no se puedan emparejar. Con los elementos emparejados ya se pueden analizar el impacto de la intervención como si los datos se hubieran conseguido de un estudio aleatorio. El emparejamiento entre los dos conjuntos puede hacerse atendiendo a distintos criterios de similitud que darán lugar a distintos conjuntos de datos emparejados.

Este método tiene limitaciones, ya que la reducción del sesgo puede ser parcial si no se cuenta con variables explicativas significativas suficientes o si se han perdido demasiadas observaciones en el emparejamiento y los datos emparejados ya no son representativos [20]. Por eso es importante comprobar, tras el emparejamiento, que no se han perdido demasiadas observaciones y que las características de los conjuntos que reciben o no la intervención están equilibradas en la mayoría de las variables consideradas [21].

Harris presenta en [19] una amplia revisión de la literatura donde se usa PSM en educación superior y un caso de estudio donde se comparan los resultados en unas pruebas según el programa de estudios en el que estuvieran admitidos los estudiantes. Como los requisitos de acceso y las características de cada programa de estudios son distintos, las calificaciones de acceso, el perfil de los estudiantes o su motivación pueden ser factores muy influyentes en los resultados, más allá del programa de estudios que se haya seguido. La comparación directa de los resultados, sin aplicar PSM, supondría no tener en cuenta esos factores diferenciadores de los programas y se podrían obtener conclusiones erróneas. En [22] se usa PSM para eliminar el sesgo al no haber podido formar el grupo de control de manera aleatoria, estando el alcance de los resultados siempre limitado por los posibles factores de confusión que se tienen en consideración en el estudio.

Se pueden utilizar diversos algoritmos para emparejar las observaciones que tienen PS similares. En este trabajo se han utilizado varios de los algoritmos más usuales para comparar los conjuntos de datos emparejados que se consiguen y seleccionar los que presentaban un grado mayor de equilibrio. Existen diversos paquetes de R que ayudan en el proceso de PSM (por ejemplo, MatchIt [23] o Matching [24]) y en la visualización e interpretación de los resultados (por ejemplo, Cobalt [25] o PSAgraphics [26]). En este análisis se ha usado MatchIt y Cobalt, principalmente.

Los algoritmos de emparejamiento que se han usado, según el nombre que se utiliza en el paquete MatchIt, son los siguientes:

- *Exact Matching*: se emparejan las observaciones que tienen exactamente los mismos valores en todas las variables.
- *Subclassification*: se forma un número determinado de clases disjuntas en las que la distribución de covariables es tan similar como sea posible.
- *Nearest Neighbor Matching*: se usa el PS para emparejar cada observación de un conjunto con el que tenga un valor más parecido en el otro. Se va realizando la asignación en orden, con un algoritmo voraz, por lo que a veces empareja observaciones con valores muy distintos. En este caso se

puede usar una opción adicional, un calibre (*caliper*) que obliga a que la diferencia entre los valores emparejados sea menor que la proporción indicada de la desviación típica de las distancias calculadas. Además, se puede combinar este tipo de emparejamiento con el emparejamiento exacto en algunas variables seleccionadas.

- *Full Matching*: se forman tantas clases disjuntas como sea necesario. En cada clase hay una observación de uno de los dos conjuntos y tantas del otro conjunto como coincidan con la primera.

III. METODOLOGÍA

En esta sección se describe el escenario en el cual se ha planteado y realizado la intervención propuesta de acuerdo con la metodología que se detalla a continuación.

El cambio principal de la intervención consistió en dividir las pruebas de evaluación continua para que fueran más cortas y tuvieran que entregarse cada semana, buscando un trabajo más continuo y con un *feedback* general más frecuente, de acuerdo a lo visto en la literatura.

A. Contexto

La asignatura "Fundamentos de programación", obligatoria en los grados de Ingeniería Informática y de Ingeniería de Tecnologías de Telecomunicación de la Universitat Oberta de Catalunya, es también un complemento de formación en algunos másteres especializados y asignatura de libre elección dentro del programa abierto de la universidad.

Al tratarse de una universidad virtual, cuenta con un perfil de entrada de estudiantes muy heterogéneo. Sin embargo, todos cursan la asignatura del mismo modo y en las mismas aulas en un entorno de aprendizaje completamente virtual.

De forma resumida, el 82.17% de los estudiantes son hombres, y la mediana de edad en el momento de cursar la asignatura es de 30 años (el grupo de edad más popular es el de 31-40 años, seguido del de 26-30 años). La mayoría cursa simultáneamente entre dos y tres asignaturas, una situación típica de estudiantes a tiempo parcial.

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

El profesorado utiliza el tablón del aula virtual para comunicar cualquier cuestión relativa al desarrollo de las actividades. Las dudas de los estudiantes, en cambio, se comparten y resuelven desde el foro del aula o a través del buzón personal del profesor y del estudiante. El foro es un espacio donde se espera que los estudiantes participen

compartiendo sus dudas y creando conocimiento de forma colaborativa. Para la realización de la parte práctica, el estudiantado cuenta con el soporte del laboratorio de programación en C donde un profesor de laboratorio resuelve las dudas y problemas con el entorno de programación y el código en C [27].

Como asignatura introductoria de programación, parte de los principios básicos de la algoritmia combinados con la práctica de ejercicios sencillos en lenguaje C. Se combina la evaluación continuada basada en tres actividades de diseño algorítmico con dos prácticas de programación en lenguaje C y un examen final presencial. Las tres pruebas de evaluación continua (PEC) son opcionales, así que el estudiante decide cuántas y cuáles realiza, teniendo en cuenta que cada una de ellas representa una parte de la calificación final de la evaluación continua. Las dos prácticas (PR1 y PR2), en cambio, son obligatorias, así como también el examen final (EX).

B. Datos utilizados

Para analizar si los cambios introducidos en esta intervención han conseguido reducir el abandono y mejorar los resultados académicos, se ha contado con los datos de cuatro semestres: dos previos a la modificación de la planificación de actividades (segundo semestre del curso 2016-17 y primero del curso 2017-18) y dos posteriores a los cambios (segundo semestre del curso 2017-18 y primero del curso 2018-19). De esta forma se puede comparar un curso entero antes y después de la intervención realizada. Los datos provienen del almacén de datos de aprendizaje institucional (*Learning Record Store*) [28]. Se ha contado con un total de 1487 observaciones, 721 anteriores al cambio y 766 posteriores, donde cada observación es el registro de un estudiante matriculado en alguno de los cuatro semestres analizados.

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

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

C. Medidas para la evaluación

El objetivo principal del análisis es determinar si el cambio realizado ha influido positivamente en el rendimiento de los estudiantes, reduciendo el alto abandono de la asignatura. El diseño instruccional y el modelo de evaluación es comparable en ambos casos hasta la entrega de la primera práctica. A partir de ese punto, transcurrida ya una gran parte del

semestre, hay algunos cambios en la consideración de la PR2 y algunas modificaciones menores en el temario.

La superación de la asignatura está marcada principalmente por la superación de la PR1. La entrega de la primera práctica (PR1) es uno de los indicadores más claros del rendimiento final del estudiante en la asignatura. De las personas que entregan la PR1, aprueban las prácticas el 83.4% y superan la evaluación continua el 93.3%. Solo el 21% de los que no entregan la práctica superan la evaluación continua de la asignatura (aunque no pueden aprobar la asignatura al ser la PR1 un requisito).

Se ha decidido, por tanto, analizar de qué manera han influido los cambios realizados en las tasas de entrega de la PR1.

IV. MARCO DE LA INTERVENCIÓN

A continuación, se detalla la situación de partida y los cambios fundamentales introducidos por la intervención realizada.

A. Situación de partida

Con la planificación de actividades descrita en el contexto, el número de estudiantes que abandonaban la asignatura era bastante alto y el rendimiento final bajo, siempre alrededor del 30%. Estos resultados son habituales, acordes con la literatura [29][30][31] y ya aceptados como normales en una asignatura inicial de programación, aún más tratándose de una docencia completamente en línea. Analizando más en detalle la actividad de los estudiantes durante varios semestres, se observaba que el problema principal era el abandono durante las primeras semanas del semestre, que atribuimos a distintos factores propios del aprendizaje de la programación:

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

Así que, inspirados por el éxito de cambios precedentes en las asignaturas de matemáticas de la titulación que consiguieron cambiar esta tendencia [32], el curso 2017-2018 se planteó una intervención para suavizar la curva de

aprendizaje y marcar un ritmo de trabajo más continuado, con el objetivo de reducir el abandono y conseguir que un número mayor de estudiantes pudieran finalizar con éxito la asignatura.

B. Descripción de la intervención

En esta intervención la asignatura no cambió ni sus objetivos, ni los recursos docentes, ni el temario, ni tampoco los resultados de aprendizaje esperados, así que las competencias que se debían alcanzar antes y después de la intervención eran exactamente las mismas. Tampoco cambió el equipo de profesorado.

El rediseño del modelo instruccional se centró en modificar substancialmente el modelo y número de actividades de evaluación continua que se proponen a los estudiantes a lo largo de todo el semestre.

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

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

El cambio en la planificación supuso, principalmente, dividir cada PEC anterior en diversas PEC más cortas, de forma que las entregas fueran más frecuentes. El tipo de problemas incluidos seguía siendo exactamente el mismo. Simplemente, se reorganizaron los mismos ejercicios en tres bloques con múltiples entregas. La PEC1 se dividió en 4 (las nuevas PEC1 a PEC4), la PEC2 en 4 (las nuevas PEC5 a PEC8) y la PEC3 se partió en 2 (las nuevas PEC9 y PEC10).

Con esta nueva planificación, las entregas pasan a ser semanales, recibiendo el estudiante retroalimentación cada semana con la publicación de la solución y comentarios generales para todo el grupo. La evaluación individualizada y la calificación por parte del profesor se ha mantenido en tres momentos del semestre, después de cada bloque, es decir, justo después de la cuarta, la octava y la décima actividad (PEC4, PEC8 y PEC10), respectivamente.

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

siguiente:

- PEC1: Ejercicio para trabajar tipos básicos de datos.
- PEC2: Ejercicio de expresiones con tipos básicos.
- PEC3: Ejercicio para practicar con la estructura alternativa.
- PEC4: Ejercicio de uso de la estructura iterativa.
- PEC5: Ejercicio que requiere el uso de tuplas.
- PEC6: Ejercicio de uso de acciones y funciones (modularidad).
- PEC7: Ejercicio que requiere del paso de parámetros.
- PEC8: Ejercicio que requiere el uso de tablas.
- PR1: Práctica de integración de todo lo aprendido en un proyecto de programación en C de más envergadura.
- PEC9: Ejercicio que introduce un tipo abstracto de datos sencillo: pila, cola o lista.
- PEC10: Ejercicio para practicar operaciones con pilas, colas y listas.
- PR2: Añadir al proyecto de la PR1 una estructura de un tipo de datos abstracto.

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

Las pruebas o actividades semanales están alineadas con las prácticas y forman parte del mismo contexto (caso práctico), de tal modo que los ejercicios de programación semanales acaban siendo utilizados después en parte de las prácticas y así se simplifica la comprensión del contexto de las prácticas y se facilita su resolución. Aunque la evaluación continua es optativa, estas prácticas son de realización obligatoria y es también indispensable superarlas para aprobar la asignatura. El cuadro I resume el modelo de evaluación.

Cuadro I
SISTEMA DE EVALUACIÓN

	Número	Obligatoria	Presencial/Virtual
PEC	3 (antes) / 10 (después)	NO	Virtual
Prácticas	2	SI	Virtual
Examen	1	SI	Presencial

En la nota de evaluación continua, las PEC tienen un peso de 40% y las PR del 60%. Esta nota se combina con la del examen según el siguiente porcentaje: 70% nota de evaluación continua y 30% nota del examen. Si no se hacen las PEC, la nota final se calcula de acuerdo con la siguiente fórmula: 40% de la nota de las prácticas y 60% de la nota del examen.

V. RESULTADOS

La intervención presenta una mejora clara en el porcentaje de personas que entregan la primera práctica. El porcentaje de estudiantes que entregan PR1 ha aumentado en los semestres posteriores al cambio, pasando del 48.13% al 60.70%.

La intervención realizada muestra también un claro aumento del porcentaje de estudiantes que supera la asignatura y un acusado descenso del porcentaje de no presentados en la evaluación continua.

En las figuras 1 y 2 se puede ver cómo evolucionan las calificaciones de los estudiantes en las distintas entregas de evaluación continua hasta el momento de la presentación de la práctica PR1 antes y después de la intervención. Los dos colores indican si el estudiante presentó o no la práctica 1 (en verde y marrón, respectivamente). Se observa cómo el número de abandonos (no presentados, N) va aumentando a medida que avanza el semestre. Con la planificación original de actividades, hay un grupo grande de estudiantes que no entrega la primera PEC y ya no vuelve a presentar ninguna actividad posterior. En el caso de la nueva planificación, ese número es menor y, además, se observa que algunas personas que obtienen bajas calificaciones en las primeras entregas consiguen ponerse al día y aprobar el resto de PEC y presentar la práctica.

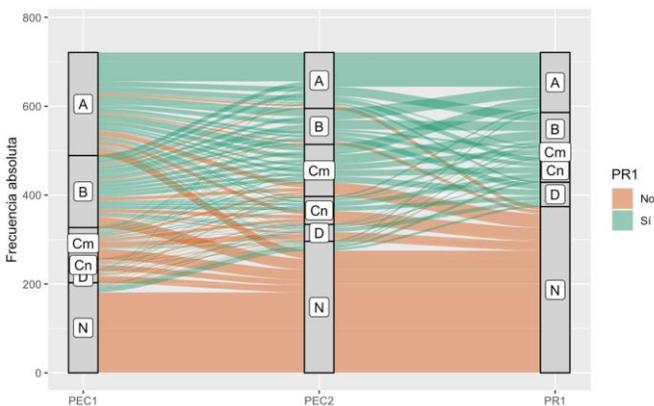


Figura 1. Evolución de las calificaciones de PEC en semestres previos según entreguen o no PR1.

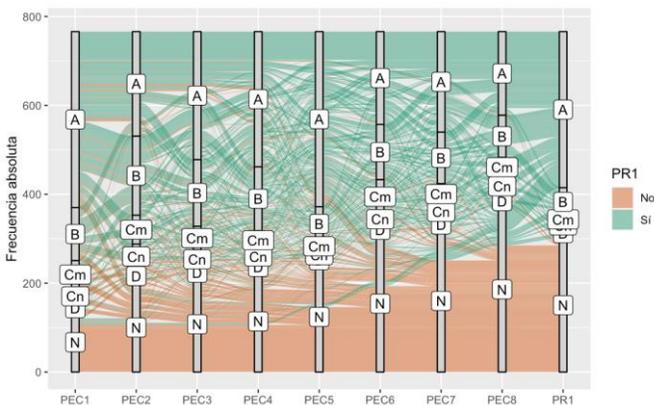


Figura 2. Evolución de las calificaciones de PEC en semestres posteriores según entreguen o no PR1.

En ambos gráficos se observa también la tendencia general a mantener las calificaciones en niveles similares de una actividad a la siguiente: los estudiantes que empiezan con calificaciones de A o B se suelen mantener en el mismo rango. Sin embargo, las personas que empiezan con malos resultados abandonan en un alto porcentaje.

Se puede observar que el número de estudiantes que no entregan la PEC1 antes del cambio duplica al número de quienes no entregan la PEC4 tras la intervención. Estas entregas se corresponden con la misma semana del curso.

Sigue habiendo diferencia, aunque más pequeña, al comparar la entrega de la PEC2 previa con la PEC8 de los semestres

Cuadro II
DISTRIBUCIÓN DE NOTAS PRIMERA PRÁCTICA (PR1)

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

posteriores.

El cuadro II muestra el desglose de notas obtenidas por los estudiantes antes y después de la intervención. Se puede observar un incremento de los aprobados, especialmente de las calificaciones Notable (B) y Excelente (A). Y, sobre todo, hay un importante descenso de no presentados (N). El cambio en la planificación parece haber mejorado también las calificaciones finales de las prácticas, incrementado el número de aprobados.

En las figuras 3 y 4 se muestran dos mapas de calor, obtenidos a partir de los datos previos y posteriores a la intervención, respectivamente, que relacionan la calificación de las prácticas y de la evaluación continua si se ha entregado la primera práctica. Se puede observar la alta correlación que hay en la superación o no de ambos bloques en ambos casos.

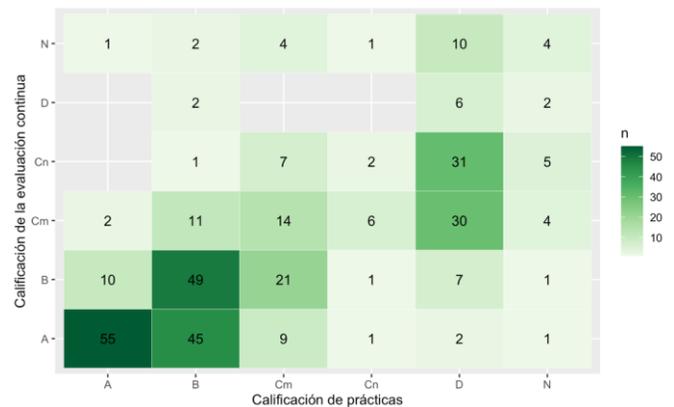


Figura 3. Relación entre las calificaciones de prácticas y evaluación continua si se presenta PR1 en los semestres previos a la intervención.

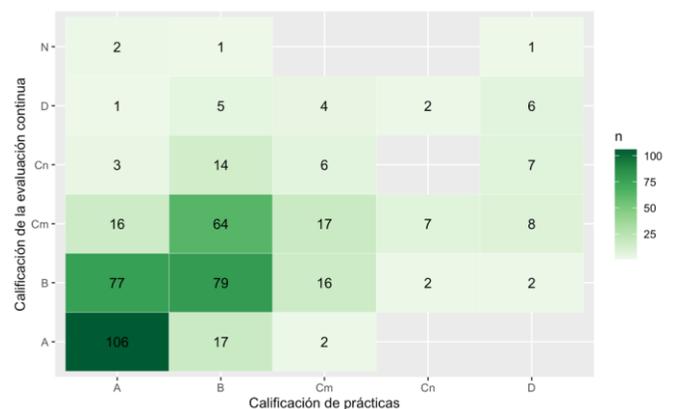
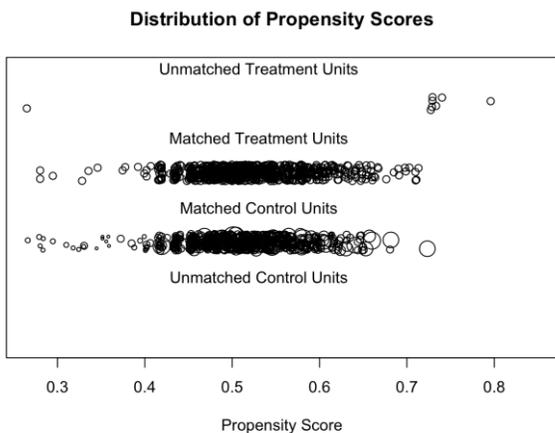


Figura 4. Relación entre las calificaciones de prácticas y evaluación continua si se presenta PR1 en los semestres posteriores a la intervención.

Sin embargo, en los semestres previos a los cambios en la planificación hay un grupo importante de personas que

aprueban la evaluación continua con calificaciones bajas (C+, C-) que no superan las prácticas (D). En



	2016.2	2017.1	2017.2	2018.1
Rendimiento	34.3 %	23.5 %	52.8 %	48.72 %
Seguimiento de la EC	60.2 %	61.9 %	86.2 %	87.0 %

Se puede pensar en las implicaciones que este análisis puede tener para el futuro de la asignatura. Por una parte, parece razonable pensar que el incremento de actividades suaviza la curva de aprendizaje inicial y consigue un ritmo de trabajo más continuado, de tal modo que el estudiante participa en la asignatura desde el principio. Estos dos factores incrementan la posibilidad de superar la asignatura y también redundan en mejores resultados. Como muestra el cuadro III, el rendimiento de la asignatura mejoró significativamente los semestres posteriores a la intervención.

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

No obstante, a pesar de los buenos resultados, hay que tener en cuenta que la intervención ha implicado también un incremento importante de trabajo para el profesorado, especialmente porque necesitan más tiempo para las correcciones y por la complicación que conlleva preparar actividades de evaluación continua vinculadas entre sí y

relacionadas todas con el mismo contexto que las prácticas. Además, para algunos estudiantes es complicado seguir un ritmo de actividades semanal, especialmente si cursan simultáneamente otras asignaturas con un planteamiento similar. Y, aunque los estudiantes no vieron incrementada la carga de trabajo, sino que esta se repartió de forma más gradual a lo largo del semestre, sería interesante conocer el impacto que tiene esta intervención sobre las otras asignaturas que se cursan simultáneamente. Desde el curso 2021-22, la carga de evaluación se ha suavizado, automatizando la corrección de algunas actividades.

A. Validación del análisis con PSM

Para que estos indicadores previos y posteriores a la intervención sean realmente comparables, debemos asegurarnos de que las poblaciones de ambos periodos son similares.

Se ha utilizado un análisis basado en PSM para emparejar los datos previos y posteriores a la intervención. Tras el emparejamiento de los datos se ha comprobado si se ha descartado o no una cantidad significativa de datos, para determinar si se mantiene un subconjunto suficientemente representativo de los datos originales. Además, se ha verificado que la distribución de las variables consideradas permanecía equilibrada en ambos conjuntos.

Se han realizado pruebas empleando diferentes algoritmos de emparejamiento (*Exact Matching*, *Subclassification*, *Nearest Neighbor Matching* con calibre y sin él y *Full Matching*), aunque los resultados no son válidos en algunos de ellos. Por ejemplo, con el algoritmo *Exact Matching* se consigue un equilibrio final absoluto en todas las variables, pero se pierden más de la mitad de las observaciones, al emparejar solo las observaciones de cada conjunto que tienen valores idénticos en todas las variables. En otros algoritmos de emparejamiento se descartan estudiantes de grupos que son poco numerosos, como las mujeres o los estudiantes de programas de estudios minoritarios con el método *Nearest Neighbor Matching* sin calibre. En estos casos, el conjunto de datos emparejados deja de estar equilibrado y pierde también generalidad.

Los resultados han sido adecuados, con pocas pérdidas de datos (solo 7 casos descartados) y variables finales equilibradas, en el caso de los algoritmos *Nearest Neighbor Matching* con calibres 0.1 y 0.2 (los valores que más se suelen utilizar), y con el algoritmo *Full Matching*, que proporciona los mejores resultados. Estos son los algoritmos que más suelen utilizarse y aparecen descritos en la literatura sobre PSM [21].

El algoritmo de emparejamiento *Full Matching* es el que ha devuelto un conjunto de datos emparejados mayor y sin desequilibrios en las covariables. En un análisis detallado de los resultados se aprecia que solo se descartan 7 observaciones y el equilibrio es adecuado en todas las covariables. En la figura 5 se observa cómo la distribución de los PS

emparejados en los conjuntos de datos previo y posterior a la intervención con el algoritmo *Full Matching* es similar. Las observaciones no emparejadas (en la parte superior de la figura) se corresponden con casos cuyos índices de propensión están fuera del rango común.

En la figura 6 se pueden comparar los histogramas de los índices de propensión. Las distribuciones son muy similares antes y después del emparejamiento (se han descartado pocos elementos) y también en el conjunto de control y el posterior a la intervención. Por tanto, el proceso de emparejamiento parece adecuado porque se mantienen casi todos los datos originales y, además, no se aprecian diferencias en las características de la población de estudiantes previa a la intervención y de la población posterior a la intervención, como demuestra el análisis del equilibrio de las covariables analizadas.

todas las covariables y sus interacciones se mantienen equilibradas.

Con resultados bastante equilibrados y con un bajo número de observaciones descartadas en varios algoritmos de emparejamiento, se puede deducir que los perfiles de los estudiantes en los conjuntos de datos de los semestres previos y posteriores a la intervención eran bastante similares. Por ello, el análisis realizado en el anterior apartado de resultados sobre rendimiento y seguimiento resulta adecuado. Con estos datos, en [33] se presentan y analizan varios modelos de regresión logística para cuantificar la influencia de la intervención en la mejora de la tasa de entrega de la primera práctica.

VI. CONCLUSIONES

En resumen, lo que resulta más relevante para la realización de la actividad obligatoria (y en consecuencia para la superación de la asignatura) es que el estudiante se comprometa con la asignatura 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 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 ofreciéndoles el apoyo que necesiten para ponerse en marcha cuanto antes.

Esta mejora está alineada con los resultados que se presentan en la literatura de investigación en educación en informática. El aumento de la retroalimentación mejora los resultados y disminuye el abandono de los estudiantes. Con las entregas de actividades más frecuentes se recibe *feedback* grupal más a menudo a través de los comentarios generales del profesor y de la solución publicada de cada actividad. De este modo, el estudiante que va entregando las actividades a tiempo, tiene información cada semana sobre su progreso y puede pedir ayuda si lo necesita. Con el cambio en el calendario de actividades, además, se consigue que el estudiantado mantenga un ritmo de trabajo más continuo, constante y progresivo, que suaviza la curva de aprendizaje y mejora el rendimiento final. Como contrapartida, exige gestionar mejor el tiempo de dedicación a la asignatura porque hay entregas cada semana.

Partiendo de los resultados de esta investigación, el trabajo futuro pasa por el diseño, implementación y evaluación de otra intervención que mejore el retorno y acompañamiento más individualizado a los estudiantes en las primeras actividades, con el propósito de recuperar cuanto antes a los estudiantes que están en situación de riesgo de abandonar la asignatura o de no superarla.

Por último, a nivel metodológico, es importante usar técnicas cuasiexperimentales tipo PSM para corregir los posibles sesgos derivados de la comparación de poblaciones potencialmente distintas.

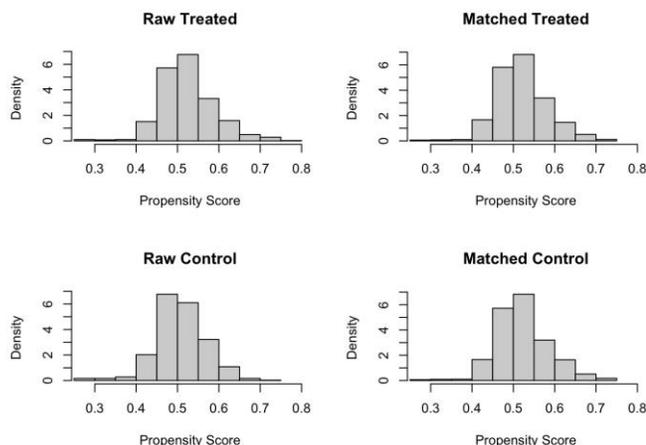


Figura 6. Histograma de PS con algoritmo *Full Matching*.

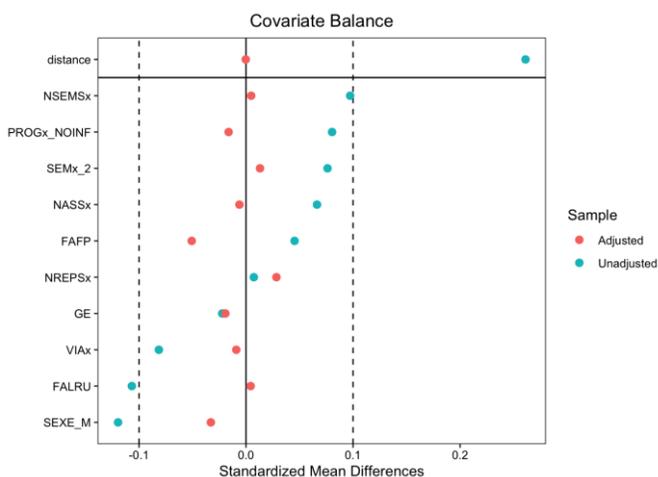


Figura 7. Equilibrio de covariables con algoritmo *Full Matching*.

Todas las covariables consideradas están equilibradas tras el emparejamiento, como puede apreciarse en la figura 7, donde se compara el equilibrio antes y después del emparejamiento de todas las covariables del perfil sociodemográfico y de estudios. En un análisis detallado de los datos se verifica que

AGRADECIMIENTOS

Los autores quieren agradecer la invitación del Comité Editorial de CINAIC 2021 para la publicación del artículo en la revista VAEP-RITA. Esta investigación ha sido parcialmente financiada por el proyecto 2021SGR01412 STEAM University Learning Research Group de la Generalitat de Catalunya.

REFERENCIAS

- [1] A. Luxton-Reilly *et al.*, «Introductory programming: a systematic literature review», en *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 2018.
- [2] C. Watson y F. W. B. Li, «Failure Rates in Introductory Programming Revisited», en *Proceedings of the 2014 Conference on Innovation in Computer Science Education*, en ITiCSE '14, New York, NY, USA: Association for Computing Machinery, 2014, pp. 39-44. doi: 10.1145/2591708.2591749.
- [3] A. S. Carter, C. D. Hundhausen, y O. Adesope, «Blending measures of programming and social behavior into predictive models of student achievement in early computing courses», *ACM Trans. Comput. Educ. TOCE*, vol. 17, n.º 3, pp. 1-20, 2017.
- [4] J. Sorva, *Visual program simulation in introductory programming education*. Aalto University, 2012.
- [5] R. Hoda y P. 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*, 2014, pp. 159-162.
- [6] C. Ott, A. Robins, y K. Shephard, «Translating principles of effective feedback for students into the CS1 context», *ACM Trans. Comput. Educ. TOCE*, vol. 16, n.º 1, pp. 1-27, 2016.
- [7] 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.
- [8] G. Kanaparan, R. Cullen, y D. D. Mason, «Self-Efficacy and Engagement as Predictors of Student Programming Performance.», en *PACIS*, 2013, p. 282.
- [9] D. Bueno, *Neurociencia para educadores*, 3.ª ed. Octaedro, 2017.
- [10] J. Biggs y C. Tang, *Teaching for Quality Learning at University*, 4th ed. SRHE and Open University Press, 2003.
- [11] B. E. Vaessen, A. van den Beemt, G. van de Watering, L. W. van Meeuwen, L. Lemmens, y P. den Brok, «Students' perception of frequent assessments and its relation to motivation and grades in a statistics course: a pilot study», *Assess. Eval. High. Educ.*, vol. 42, n.º 6, pp. 872-886, 2017, doi: 10.1080/02602938.2016.1204532.
- [12] A. Sangra, «Decálogo para la mejora de la docencia online: propuestas para educar en contextos presenciales discontinuos», *Decál. Para Mejora Docencia Online*, pp. 1-215, 2020.
- [13] 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*, en ICER '14, New York, NY, USA: ACM, 2014, pp. 19-26. doi: 10.1145/2632320.2632349.
- [14] 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)*, 2021, pp. 677-681.
- [15] G. Cousin, *Researching Learning in Higher Education: An Introduction to Contemporary Methods and Approaches*. en SEDA Series. Taylor & Francis, 2009. [En línea]. Disponible en: <https://books.google.es/books?id=I3WQAgAAQBAJ>
- [16] A. J. Ko y S. A. Fincher, «A Study Design Process», en *The Cambridge Handbook of Computing Education Research*, Cambridge University Press, 2019, pp. 81-101. doi: 10.1017/9781108654555.005.
- [17] M. M. Arias, «Lectura crítica en pequeñas dosis Índices de propensión. El deseo de parecerse al ensayo clínico», *Rev Pediatr Aten Primaria*, vol. 17, pp. 87-90, 2015.
- [18] C. Ramirez Ovalle, «Sobre la Técnica de Puntajes de Propensión (Propensity Score Matching) y sus usos en la investigación en Educación», *Educ. Cienc.*, vol. 4, n.º 43, pp. 81-89, 2015.
- [19] H. D. Harris, «Propensity score matching in higher education assessment», *Masters Theses*, may 2015, [En línea]. Disponible en: <https://commons.lib.jmu.edu/master201019/55>
- [20] S. Guo, M. Fraser, y Q. Chen, «Propensity score analysis: Recent debate and discussion», *J. Soc. Soc. Work Res.*, vol. 11, n.º 3, pp. 463-482, sep. 2020, doi: 10.1086/711393/ASSET/IMAGES/LARGE/FG1.JPEG.
- [21] H. Harris y S. J. Horst, «A brief guide to decisions at each step of the propensity score matching process», *Pract. Assess. Res. Eval.*, vol. 21, n.º 4, 2016.
- [22] L. A. Lim *et al.*, «What changes, and for whom? A study of the impact of learning analytics-based process feedback in a large course», *Learn. Instr.*, 2019, doi: 10.1016/j.learninstruc.2019.04.003.
- [23] D. E. Ho, K. Imai, G. King, y E. A. Stuart, «MatchIt: Nonparametric Preprocessing for Parametric Causal Inference», *J. Stat. Softw.*, vol. 42, n.º 8, pp. 1-28, 2011.
- [24] J. S. Sekhon, «Multivariate and Propensity Score Matching The Matching package for R», *J. Stat. Softw.*, vol. 10, n.º 2, 2008.
- [25] N. Greifer, *cobalt: Covariate Balance Tables and Plots*. 2020. [En línea]. Disponible en: <https://cran.r-project.org/package=cobalt>
- [26] J. E. Helmreich y R. M. Pruzek, «PSAgraphics: An R package to support propensity score analysis», *J. Stat. Softw.*, vol. 29, n.º 6, pp. 1-23, 2009, doi: 10.18637/jss.v029.i06.
- [27] M. J. Marco Galindo y J. Prieto Blázquez, «Necesidades específicas para la docencia de programación en un entorno virtual», *Actas JENUI*, pp. 5-12, 2002.
- [28] 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.
- [29] A. V. Robins, «Novice Programmers and Introductory Programming», en *The Cambridge Handbook of Computing Education Research*, S. A. Fincher y A. V. Robins, Eds., en Cambridge Handbooks in Psychology. Cambridge University Press, 2019, pp. 327-376. doi: 10.1017/9781108654555.013.
- [30] A. Luxton-Reilly, «Learning to Program is Easy», en *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, New York, NY, USA: Association for Computing Machinery, 2016, pp. 284-289. doi: 10.1145/2899415.2899432.
- [31] G. Bain y I. Barnes, «Why Is programming so hard to learn?», en *ITiCSE 2014 - Proceedings of the 2014 Innovation and Technology in Computer Science Education Conference*, Association for Computing Machinery, 2014, p. 356. doi: 10.1145/2591708.2602675.
- [32] T. Sancho-Vinuesa, R. Masià, M. Fuertes-Alpiste, y N. Molas-Castells, «Exploring the effectiveness of continuous activity with automatic feedback in online calculus», *Comput. Appl. Eng. Educ.*, vol. 26, n.º 1, pp. 62-74, 2018.
- [33] A. Gómez, M. J. Marco-Galindo, J. Minguillón, y J. Escayola-Mansilla, «Análisis de la mejora de los resultados de una asignatura inicial de programación tras cambiar la planificación de actividades», en *Actas de las Jornadas sobre Enseñanza Universitaria de la Informática (JENUI)*, 2021, pp. 83-90.

Alberto Gómez Licenciado en informática por la Universitat Politècnica de Catalunya (UPC). Máster Universitario en Ciencia de Datos por la Universitat Oberta de Catalunya (UOC). Profesor de la Universidad de Extremadura, imparte clases en materias de programación, principalmente. Ha participado y dirigido diversos proyectos de innovación educativa en temas relacionados con nuevas metodologías, analíticas de aprendizaje y enseñanza de programación. Es miembro de AENUI (Asociación de Enseñantes Universitarios de Informática), donde ha recibido el Premio a la Calidad e Innovación Docente de 2019.

Maria-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 EduSTEAM (STEAM University

Como citar este artículo: A. Gómez, M. -J. Marco-Galindo and J. Minguillón, "Evaluation of an Intervention on Activity Planning in CS1," in IEEE Revista Iberoamericana de Tecnologías del Aprendizaje, vol. 18, no. 3, pp. 287-294, Aug. 2023, doi: 10.1109/RITA.2023.3302174. 10

Learning Research Group), en temas relacionados con la docencia de la programación y de las competencias transversales en entornos de virtuales. Es miembro de AENUI, donde ha recibido el Premio a la Calidad e Innovación Docente de 2023.

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 EduSTEAM, 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.