Actas de las Jenui, vol. 1. 2016.
Páginas: 241-248

Metodologías activas y gamificación en las asignaturas de iniciación a la programación

Jesús R. Campaña[1], Ana E. Marín[2], María Ros[1], Daniel Sánchez[1], Juan M. Medina[1], M. Amparo Vila[1], M. Dolores Ruiz[1], Manuel P. Cuéllar[1], Maria J. Martin-Bautista[1]

[1] Dpto. de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada

[2] Dpto. de Estadística e Investigación Operativa, Universidad de Granada

jesuscg@decsai.ugr.es, anamarin@ugr.es, marosiz@decsai.ugr.es

[daniel|medina|vila|mdruiz|manupc|mbautis]@decsai.ugr.es

Resumen

Aprender a programar es una tarea compleja que requiere del desarrollo de diversas habilidades. Los estudiantes que se inician en la programación se encuentran con serias dificultadas en el aprendizaje de esta materia. Actualmente en los nuevos Grados se han introducido asignaturas relacionadas con las tecnologías de la información, en donde se realiza un primer contacto con la programación. En esta comunicación presentamos el trabajo realizado para el desarrollo de una metodología docente para la enseñanza de programación, basada en el uso de metodologías activas y el empleo de gamificación para incentivar la participación del alumnado y aumentar su motivación. Comentamos la implantación de esta metodología en un curso de iniciación a la programación del Grado en Ingeniería Informática y exponemos los resultados obtenidos.

Abstract

Learning programming is a complex task that requires the development of various skills. Students who are new to programming encounter serious difficulties in learning the subject. Nowadays degrees in the new study system have introduced subjects related to information technology, where students make their first contact with programming. In this paper we present the work done to develop a methodology for teaching programming based on the use of active methodologies and gamification, to encourage student participation and increase motivation. We discuss the implementation of this methodology in an introductory course to programming in the Degree in Computer Science and present some results.

Palabras clave

Programación, Metodologías Activas, Gamificación

1. Introducción

La implantación de los nuevos planes de estudio ha introducido asignaturas relacionadas con las tecnologías de la información en diversos estudios de Grado, en los que previamente no existían. Como parte de estas nuevas asignaturas se exigen competencias relacionadas con el aprendizaje de algún lenguaje de programación.

Aprender a programar es un proceso bastante dificultoso, máxime si tenemos en cuenta que supone la organización y sistematización de los procesos de pensamiento, e implica además un alto grado de creatividad a la hora de resolver un problema. El alumnado que se inicia en la programación, suele encontrar bastantes dificultades para aprender los principales conceptos y el uso del razonamiento algorítmico. A todo esto se añade un nivel de dificultad adicional cuando es necesario resolver problemas empleando un lenguaje de programación.

Estas dificultades son comunes a todas las asignaturas de iniciación a la programación de los distintos Grados, pero pueden presentar un grave inconveniente en los estudios del Grado de Ingeniería Informática. Mientras que en otros Grados el aprendizaje de un lenguaje de programación es algo útil, pero complementario, en los estudios de informática es fundamental, ya que si el alumnado no alcanza las competencias exigidas en las asignaturas de iniciación, su rendimiento a lo largo de los diferentes cursos puede verse seriamente afectado. Es por esto, por lo que hemos decidido enfocar el desarrollo de la experiencia en la asignatura de *Fundamentos de Programación* del *Grado en Ingeniería Informática*.

Los nuevos planes de estudios conllevan nuevos re-

querimientos docentes tales como trabajo autónomo del alumnado, evaluación mediante competencias, etc. Los materiales y las metodologías basadas únicamente en clases magistrales y la realización de ejercicios de prácticas, no bastan para que el alumnado alcance las competencias necesarias. Sin embargo es éste mayoritariamente el material disponible para programación.

La dificultad del aprendizaje de la programación no es un problema exclusivo del alumnado en España, sino que es un problema a nivel mundial sobre el que se ha colocado el foco y se está trabajando para encontrar una solución. Si a la dificultad inherente de la tarea, añadimos las características particulares de nuestro centro, en el que el alumnado llega con ciertas carencias formativas y poca motivación, el problema se acrecienta. La alta tasa de fracaso a la hora de afrontar las asignaturas de programación, acaba por lastrar el éxito del alumnado en la finalización de sus estudios de Grado, y produce una alta tasa de abandono inicial.

Para solventar en la medida de lo posible este problema, hemos puesto en marcha una iniciativa para añadir metodologías activas al aprendizaje de la programación, y añadir un componente lúdico que permita motivar al alumnado mediante la gamificación de la evaluación continua de la asignatura. Tratamos de aumentar la motivación e implicación del alumnado con el objetivo de aumentar el rendimiento académico y reducir la tasa de abandono inicial.

La evaluación de los resultados obtenidos empleando esta metodología se valora atendiendo a resultados objetivos de rendimiento académico y reducción en la tasa de abandono inicial, así como subjetivos medidos a través de encuestas de satisfacción, que además servirán para valorar el interés mostrado por el alumnado en cada actividad.

Este artículo está organizado en las siguientes secciones: La Sección 2 presenta una contextualización del problema y las distintas actividades a realizar, presentando algunos conceptos relevantes a nuestras propuesta, tales como gamificación y pensamiento computacional. En la Sección 3, se presenta la metodología empleada y se detallan los criterios utilizados en la gamificación del aprendizaje, mientras en la Sección 4 se evalúa la propuesta. Finalmente la Sección 5 presenta las conclusiones y trabajo futuro.

2. Actividades Propuestas

El profesorado ha podido comprobar como el número de alumnos que superan las asignaturas de programación es bajo con respecto al total de matriculados. Investigando sobre este fenómeno en la literatura y en la red, hemos podido comprobar que no es exclusivo de nuestra universidad, ni siquiera de nuestro país. Aprender a programar, es una tarea de gran dificultad y es clave para lograr el éxito en los estudios de informática. Para algunos estudiantes el tener una mala experiencia con las asignaturas de programación puede llevarles incluso a abandonar los estudios. Es por esto que nos proponemos como objetivo general a alcanzar, la mejora de la tasa de éxito de alumnos en asignaturas de iniciación a la programación, así como la reducción de la tasa de abandono inicial.

Programar conlleva lograr dos objetivos diferenciados. Por una parte tenemos el aprendizaje de un lenguaje de programación. Este objetivo consiste en comprender y aprender, el uso de las diferentes instrucciones y aplicarlas para el desarrollo de un programa. El segundo objetivo es bastante más complejo y está estrechamente ligado con el primero, ya que consistiría en la resolución de un problema, creando para ello un programa en el lenguaje que se está estudiando. Si bien ambos objetivos están estrechamente ligados, la consecución de uno de ellos, no implica directamente la del otro. Si bien el alumnado puede llegar a aprender y comprender el uso del lenguaje de programación, eso no garantiza que sepa usarlo para resolver problemas. Del mismo modo, es posible que el alumnado llegue a determinar un procedimiento para resolver un problema y no sea capaz de expresar esa solución con el lenguaje de programación.

La Universidad de San Diego es pionera en la enseñanza de cursos de iniciación a la programación utilizando nuevas metodologías activas que han conducido a mejorar la tasa de éxito de sus estudiantes. Una combinación de Programación en Parejas (*Pair Programming*), Instrucción entre Pares (*Peer Instruction*) y Computación Multimedia (*Media Computation*), ha permitido a los estudiantes comprender mejor los conceptos asociados a la programación [9].

2.1. Programación en Parejas

La programación en parejas consiste en asignar la resolución de un problema a una pareja de estudiantes. La actividad se inicia con ambas partes acordando una solución, mientras un miembro de la pareja codifica la solución, el otro revisa el código creado. Pasado un intervalo de tiempo a decidir por el profesor, los roles se intercambian. El proceso se repetirá varias veces, hasta que se termine el tiempo asignado, o se haya encontrado con la solución al problema.

Esta actividad consigue aumentar la confianza y capacidad de resolver problemas de los programadores noveles. Esta técnica permite mejorar la retención, especialmente en el sector femenino del alumnado, tal y como se indica en los trabajos de Werner, McDowell et al. [4, 7].

2.2. Instrucción entre Pares

La instrucción entre pares es una actividad centrada en el alumnado, que fomenta la transferencia de información entre los propios estudiantes. La actividad comienza con la entrega de material o lecturas por parte del profesor, las cuales deben trabajarse de forma autónoma. Posteriormente, en clase se plantea un problema y los estudiantes deben intercambiar ideas para llegar a una solución de consenso. El profesor comenta las ideas que se han ido exponiendo, y los estudiantes discuten sobre ello y vuelven a escoger una solución de consenso. El proceso se repite hasta que los conceptos han sido asimilados.

2.3. Computación Multimedia

La computación multimedia, consiste en la resolución de problemas en los que se explica como se manipulan elementos multimedia (vídeo, audio, imágenes,...). Los estudiantes responden mejor si tienen un contexto de aplicación con el que trabajar. Este tipo de actividad es especialmente eficaz en grupos pertenecientes a minorías como se indica en los trabajos de Forte [1]. El principal beneficio observado es el aumento de la motivación.

Es habitual que el alumnado manifieste en las primeras experiencias con la programación cierta frustración, ya que los primeros programas suelen ser sobre cálculos matemáticos y en modo texto. Pasar a la realización de programas visuales, requiere de la adquisición de unas competencias previas que aún no se han podido desarrollar. Sin embargo, podemos utilizar contenidos multimedia que permitan observar las modificaciones que se hace de ellos mediante un programa. Los ejemplos más claros son lo de la manipulación de imágenes o de sonidos. El alumnado puede observar cómo sus programas afectan a una imagen o un sonido, haciendo palpable la relación causa efecto de una forma atractiva, y sin necesidad de utilizar complejos entornos de programación visual, dado que los archivos se abren con programas externos.

2.4. Aprendizaje Basado en Proyectos

Como parte de la evaluación de la asignatura el alumnado debe realizar pequeños proyectos de programación. Estos proyectos se basan en el desarrollo de programas breves a partir de un enunciado y unas directrices básicas. Para dotar a la actividad de un carácter motivador, los problemas propuestos consisten en la realización de videojuegos con sencillas interfaces de texto, como pueden ser *3 en raya*, *ahorcado*, etc. o problemas básicos de computación multimedia para manipulación de imágenes.

2.5. Problemas Resueltos

Habitualmente se suele decir que "La programación se aprende programando", pero esta afirmación no es del todo acertada. En 1985 Sweller y Cooper [6], publicaron un artículo en el que a través de un experimento con problemas de álgebra, se mostraba que el alumnado adquiría mejores competencias para resolver problemas, viendo problemas resueltos, en lugar de resolviendo problemas. Algunas experiencias han demostrado que ocurre lo mismo en el ámbito de la programación.

A la hora de afrontar el trabajo autónomo es muy importante tener acceso a una amplia batería de problemas resueltos. Los problemas se deben adaptar y documentar paso a paso para que puedan ser comprendidos por los estudiantes de forma autónoma. El disponer de una amplia variedad de problemas resueltos les ayudará a tener una mejor perspectiva sobre diferentes formas de resolver problemas.

El material para trabajo autónomo ha sido desarrollado por el profesorado, y también se han seleccionado y adaptado problemas provenientes de universidades y centros de investigación de reconocido prestigio que están disponibles a través de la web o cursos OpenCourseWare (Harvard University OCW `http://cs50.tv`, MIT OCW `http://ocw.mit.edu`). Adicionalmente, los propios alumnos pueden contribuir a la elaboración de estas baterías de problemas, aportando sus soluciones a los problemas planteados en las relaciones de ejercicios.

2.6. Ejercicios de Trazas de Código

Habitualmente en los primeros cursos de programación se hace mucho énfasis en el desarrollo de programas, y se deja un poco de lado la lectura y traza del código. Una de las partes más importantes y que consumen bastante en el desarrollo de un programa, es la depuración del mismo para localizar errores. Los ejercicios de trazas de código permiten ejercitar la habilidad de detectar fallos de programación en un código. Además, las personas que decidan dedicar su carrera profesional al desarrollo de programas, con total seguridad acabarán trabajando integradas en equipos, por lo que se hace imprescindible el desarrollo de competencias relacionadas con la lectura y comprensión de programas desarrollados por otras personas.

Para tratar de evaluar de la forma más adecuada las competencias obtenidas en materia de lectura y trazas de código, empleamos una prueba objetiva, basada en la propuesta de Lister et al [3]. Esto nos permitirá realizar una evaluación de los resultados, comparándolos con los reflejados en el artículo.

2.7. Pensamiento Computacional y Programación Visual

El Pensamiento Computacional (*Computational Thinking*) [8] consiste en el desarrollo de un conjunto de habilidades para la resolución de problemas, diseño de sistemas, y comprensión del comportamiento humano, empleando los conceptos fundamentales utilizados en ciencias de computación.

Del mismo modo que en generaciones recientes la alfabetización resultaba absolutamente necesaria para establecer una comunicación entre pares. En la era de Internet donde el papel ha sido sustituido por la pantalla, para comunicarnos no sólo basta con saber leer y escribir, debemos alfabetizarnos en este nuevo medio, lo que implica generar contenidos multimedia y establecer interacciones de dichos contenidos con los usuarios.

En este escenario los conocimientos de programación se hacen indispensables para poder llevar a cabo una comunicación efectiva, y poder expresarnos usando este nuevo medio sin límites. Es por esto que en los últimos años ha tomado especial relevancia la idea del Pensamiento Computacional y el aprendizaje de la programación como un motor de cambio y desarrollo. Si bien puede parecer que la complejidad inherente a la tarea de la programación limitaría el aprendizaje de estas nuevas habilidades a niveles educativos medios o superiores, existe un gran interés por acercar este nuevo paradigma a los primeros niveles escolares. Con este objetivo en mente, se han desarrollado multitud de herramientas que facilitan la programación y que están pensadas específicamente para el aprendizaje en la escuela.

Scratch [5] es un entorno de programación para niños desarrollado en el Instituto Tecnológico de Massachusetts (MIT), en el que establece un paradigma de programación visual, donde las diferentes estructuras de control del lenguaje se representan de forma gráfica. Cada estructura de control es similar a una pieza de puzzle, donde las distintas estructuras pueden encajarse para formar programas. Este intuitivo sistema ha sido también empleado por Google para desarrollar un lenguaje de programación visual denominado Blockly (`https://code.google.com/p/blockly`).

Entender y comprender el funcionamiento de las estructuras de control de un lenguaje de programación es una de las tareas primordiales en los primeros pasos con la programación. Entender estos conceptos y como se relacionan con la sintaxis del lenguaje de programación que se emplee, puede ser complejo. Gracias a los entornos de programación visual podemos desacoplar el aprendizaje de las estructuras de control comunes a los lenguajes de programación, de un lenguaje de programación concreto, y por tanto centrarnos en aplicarlas en la resolución de problemas desde un punto de vista más abstracto.

Existen en la red multitud de tutoriales que permiten conocer los rudimentos de las estructuras de control en muy poco tiempo, así como diversos ejercicios y ejemplos que permiten practicar la resolución de problemas. Como parte de nuestro cambio en la metodología, incluimos actividades de resolución de problemas empleando ejercicios de programación visual. En muchos casos existe un componente lúdico en estos problemas, que los convierten en ideales para el trabajo autónomo, ya que son bastante motivadores.

2.8. Gamificación

La gamificación consiste en el uso de mecanismos de juego tales como la obtención de puntos o de insignias de progreso, para motivar y captar la atención, fomentando un determinado comportamiento deseado. El principal atractivo de la gamificación es que toma la capacidad motivadora de los juegos y la aplica a la vida real. De un tiempo a esta parte se está trabajando en la aplicación de los principios básicos de la gamificación a la educación, y en particular al aprendizaje de la programación [2].

A través de un sistema de puntos y recompensas se puede mejorar la motivación del alumnado con el objetivo de facilitar el aprendizaje de la programación. El sistema de puntos permite ver un progreso a lo largo de la asignatura y ofrece una sensación de avance en el aprendizaje. Al mismo tiempo se establece una cierta competitividad entre el alumnado dentro del ambiente de juego, lo cual les motiva a esforzarse más para mejorar las puntuaciones obtenidas.

3. Metodología Empleada

La idea principal con la que queremos impregnar la metodología empleada es la motivación. Planteando actividades diferentes y motivadoras tratamos de compensar la dificultad inherente a la tarea del aprendizaje de la programación.

Nuestro objetivo es alternar todas las actividades mencionadas anteriormente y englobarlo todo en una actividad continua gamificada, donde el alumnado va recibiendo puntos y completando actividades.

La experiencia de implantación se ha llevado a cabo en la titulación de Grado en Ingeniería Informática para la asignatura *Fundamentos de Programación*. La primera promoción de los nuevos estudios de Grado fue la del curso 2013/2014. El profesorado de la titulación había observado como muchos de los estudiantes de cursos superiores no podían graduarse por tener pendientes asignaturas de programación y que seguían teniendo serias dificultades para programar. Con el objetivo de intentar mejorar la situación se planteó es-

ta experiencia en la que hemos tratado de mejorar el aprendizaje y motivar al alumnado hacia las asignaturas de programación.

En esta primera aproximación optamos por un modelo mixto en el que decidimos gamificar la parte de evaluación continua de la asignatura, integrando las nuevas actividades con otras tradicionales como la realización de proyectos y los exámenes de evaluación. La calificación de evaluación continua se dividía en 100 puntos y cada una de las actividades realizadas indicaba claramente la cantidad de puntos a conseguir una vez completada.

Es muy importante antes de iniciar las actividades que el alumno tenga una idea absolutamente precisa de las actividades a realizar y de las recompensas a recibir una vez terminadas esas actividades. Una buena práctica es la de consensuar con el alumnado el valor que se va a otorgar a cada una de las actividades, así lo hacemos partícipe del proceso de evaluación, pero siempre guiado por el criterio del docente. Una vez establecido el número máximo de puntos a conseguir (puede variar de una asignatura a otra o incluso se puede aplicar únicamente a partes de la asignatura), y los puntos a recibir por cada actividad, se comienza el proceso.

Como parte de esta experiencia se han desarrollado diversas actividades y material docente de apoyo a las metodologías activas introducidas en el aula, así como se han creado instrumentos de evaluación de la satisfacción de los alumnos. Concretamente se han desarrollado las siguientes actividades y materiales:

Para trabajo autónomo:

- Baterías de ejercicios resueltos paso a paso. Problemas resueltos de programación, por cada uno de los bloques de la asignatura.
- Resolución de problemas utilizando programación visual.

Para clases teóricas:

- Conjuntos de actividades de instrucción entre pares. Instrucciones para el profesor con las preguntas a realizar, instrucciones para los alumnos y sugerencias sobre la dinámica a establecer.

Para clases prácticas:

- Conjuntos de ejercicios para realizar con programación en parejas. Junto a los ejercicios se desarrolla un manual para el profesorado, con instrucciones acerca de cómo organizar y gestionar los grupos y la evaluación.
- Conjuntos de ejercicios sobre computación multimedia. Se han realizado ejercicios de manipulación de contenidos multimedia que permitan dar contexto a los programas realizados por los estudiantes.

Para evaluación de la percepción de la programación y de las actividades realizadas:

- Se ha creado un modelo de encuesta para determinar y observar la actitud ante la programación, así como la valoración de las actividades propuestas. La encuesta se ha pasado a lo largo del curso, en Diciembre y Febrero, para ver la evolución de las opiniones a lo largo del tiempo.

Cada una de las actividades planteadas va acompañada de unas instrucciones precisas sobre cómo se debe desarrollar la actividad, y los puntos y recompensas a recibir. Para motivar al alumnado y favorecer la competitividad desde el punto de vista lúdico, las actividades a realizar en tiempo limitado, como por ejemplo resolución de problemas en parejas o individuales, pueden contener puntos adicionales para aquellos que terminen la tarea en menor tiempo. Esto que en principio puede parecer intrascendente, en la práctica hemos comprobado que activa bastante la dinámica en clase y fomenta el trabajo. En cursos previos a la utilización de esta metodología, las clases de prácticas eran desaprovechadas sistemáticamente por el alumnado y no finalizaban las tareas asignadas. Con la introducción de la gamificación y el sistema de puntuación, el ambiente está mucho más enfocado al trabajo y el alumnado se centra completamente en la tarea a realizar. El cambio de actitud ante las sesiones presenciales prácticas es completo, y se aprovecha mejor el tiempo.

Como complemento a las actividades, semanalmente se publica en la web de la asignatura un gráfico donde el alumnado puede seguir su progreso, ver los puntos acumulados y ver cual es su situación respecto al grupo. Nos hemos encontrado en ocasiones a estudiantes que pensaban que dedicaban el tiempo suficiente a la asignatura y que las actividades realizadas serían suficientes para superarla, sin ser del todo conscientes de que su dedicación estaba muy por debajo de lo esperado. Gracias a la gráfica de seguimiento, cada estudiante visualiza el trabajo que ha realizado, y es capaz de contextualizarlo con respecto al del resto de sus compañeros, de modo que puede tomar medidas antes de que sea demasiado tarde.

Con objetivo de validar las actividades y la metodología empleada, hemos realizado las actividades de evaluación y seguimiento que se detallan a continuación.

4. Evaluación y Seguimiento

Para poder seguir el proceso de implantación de la nueva metodología y corregir posibles deficiencias, hemos establecido un modelo de evaluación que nos ha servido de guía.

Se ha desarrollado una encuesta siguiendo la escala de tipo Likert de 5 puntos para observar cómo evolu-

1 – Totalmente en desacuerdo *2 – Algo en desacuerdo* *3 – Indiferente* *4 – Algo de acuerdo* *5 – Totalmente de acuerdo*

¿Que piensas sobre la programación?

		1	2	3	4	5
En mis estudios me resultará útil saber programar	P1					
Terminar mis estudios será más fácil si se programar	P2					
Saber programar me ayudará a encontrar trabajo	P3					
Saber programar será útil en mi trabajo	P4					
Me resultará fácil aprender a programar	P5					
Conseguir que el programa haga lo que quiero será fácil	P6					
Programar será fácil	P7					
Pienso utilizar la programación en mis estudios	P8					
En mis estudios escribiré programas que me ayuden	P9					
La programación estará presente en mi profesión	P10					

Figura 1: Preguntas sobre percepción del alumnado sobre programación.

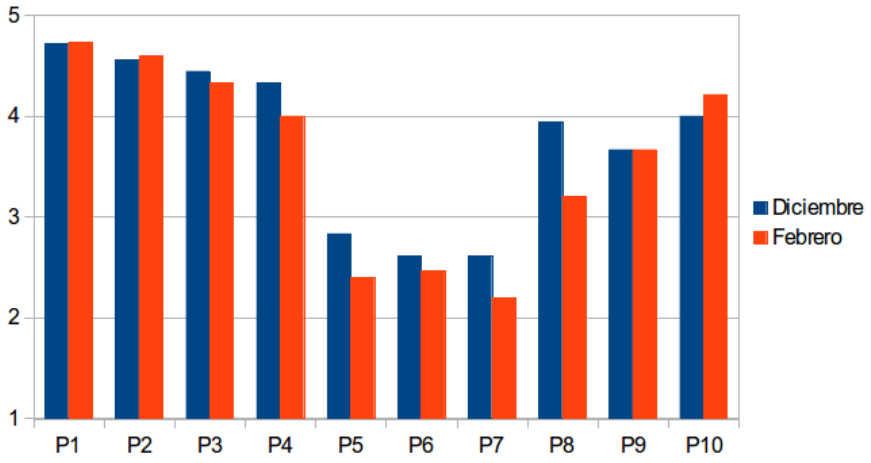

Figura 2: Percepción del alumnado sobre programación.

ciona la percepción de la programación por parte del alumnado a lo largo del curso, y sondear la opinión sobre las nuevas actividades planteadas. Las preguntas sobre aceptación de la tecnología y de actitud hacia la programación usadas se muestran en la Figura 1, y los resultados se detallan en la Figura 2.

Estas mismas preguntas ya fueron empleadas en las encuestas realizadas en el marco del proyecto de innovación docente *"PID-11-73: Desarrollo de herramientas para el trabajo autónomo del estudiante de Fundamentos de Informática en Ciencias e Ingeniería."*. En dicha ocasión, se utilizaron con cursos del Grado en Ingeniería Química y del Grado en Biología. El uso de las mismas cuestiones nos permite estudiar la actitud de diferentes tipologías de alumnado ante la programación.

La encuesta se pasó en dos ocasiones, la primera en Diciembre, tras un examen parcial, y la segunda en Febrero, tras el examen final. De esta forma se pudo tomar la opinión del 88 % del alumnado en distintos momentos, habiendo respondido un 52 % a ambas encuestas. Por los resultados obtenidos, podemos observar que los valores otorgados a las preguntas P1, P2, P9 y P10 que tratan acerca de la utilidad de la programación en general y el rol que desempeñará esta en el futuro profesional del alumnado, aumentan ligeramente. En las preguntas P1 y P2 cuanto más avanza el curso más consideran el alumnado que les será útil programar para sus estudios. Esto es importante ya que les ayuda a valorar la utilidad de la asignatura en el contexto de la titulación global.

Sin embargo, los valores para las preguntas P3 y P4, relacionadas con la utilidad de la programación para su futuro profesional, la mayoría está de acuerdo en la utilidad de la programación, pero sin embargo el valor desciende de una a otra encuesta. Donde encontramos

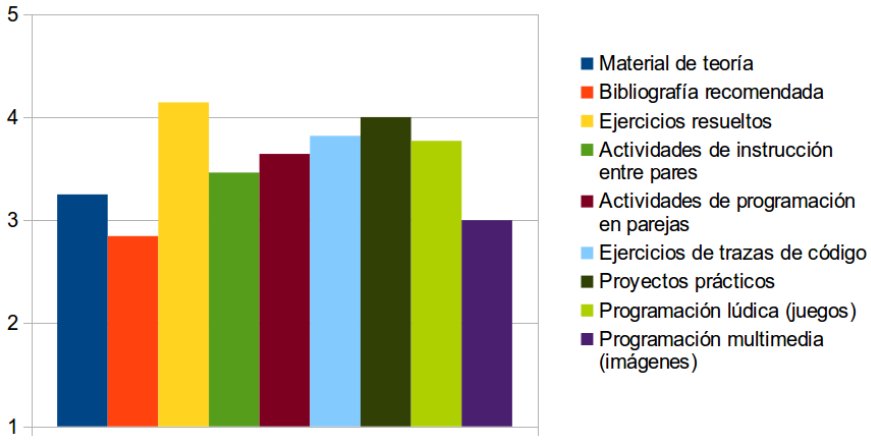

Figura 3: Valoración de las actividades por parte del alumnado.

mayor diferencia, es los valores para las preguntas P5 a P8, en estas preguntas se trata de averiguar la confianza del alumnado respecto a sus capacidades para la programación. Esto nos indica que el alumnado es consciente de la dificultad inherente a la programación, pero resulta relevante, que cuanto más tiempo llevan programando, menos confían en sus capacidades para poder conseguir los objetivos propuestos en la asignatura, esto puede deberse a que la complejidad de los contenidos aumenta a lo largo del curso.

Es de especial relevancia la respuesta a la pregunta P8, en la que existe una gran diferencia entre la primera valoración y la segunda, esta pregunta trata acerca de la percepción del uso que se hará de la programación en sus estudios, por lo que el alumnado cree que no va a usar programación tanto como creía a mitad de la asignatura. En esta apreciación puede radicar la clave de la situación actual, los estudiantes afrontan la programación como una asignatura más y no son conscientes de que es un pilar fundamental tanto para el desarrollo de su carrera profesional, como para el desarrollo de otras asignaturas del Grado.

El seguimiento de la metodología no sólo se ha realizado a través de las encuestas del alumnado, también se han realizado sesiones de tutoría individual y grupal, donde los alumnos de forma anónima pueden expresar su opinión siguiendo una práctica similar a la de *one-minute paper*, estableciendo posteriormente un diálogo y debatiendo posibles mejoras y cambios. La percepción obtenida a partir de estas sesiones de tutorías es que el alumnado estaba altamente motivado, sin embargo, esa motivación no se estaba traduciendo en una mayor implicación con la asignatura. Con el cambio de mecánica en las sesiones prácticas conseguimos aprovecharlas y exprimirlas al máximo, pero los estudiantes seguían sin dedicar tiempo suficiente al trabajo autónomo en casa.

En las encuestas, también se pidió al alumnado que expresasen su opinión sobre los distintos tipos de actividades y materiales que se habían utilizado a lo largo del curso. En la Figura 3 podemos ver las valoraciones del alumnado sobre las actividades realizadas siguiendo la escala de tipo Likert de 5 puntos.

Como se puede apreciar en la gráfica, las actividades mejor valoradas son los ejercicios resueltos, la programación en parejas, las trazas de código, la programación lúdica (videojuegos) y los proyectos prácticos. Si bien las actividades de instrucción entre pares no han sido mal valoradas, en un análisis pormenorizado podemos observar que el interés por parte del alumnado en la actividad ha estado más polarizado. La programación multimedia, realizada mediante ejercicios de manipulación de imágenes también ha dividido las opiniones del alumnado encontrándose la mayoría de valores repartidos entre las valoraciones 1, 3 y 5.

No se observaron diferencias en la opinión del alumnado sobre los materiales al realizar contrastes por sexo y experiencia previa con programación.

En cuanto al rendimiento académico, éste fue muy similar al del año anterior en el que no se aplicó la metodología, de una tasa de éxito del 38 % se ha pasado a un 46 %, lo cual en números absolutos no representa una gran cantidad debido al número reducido de alumnos. Es destacable que la retroalimentación por parte de alumnos repetidores mostraba una actitud más entusiasta hacia la asignatura, pero esto no llega a reflejarse en forma de éxito académico. Para el cálculo de la tasa de abandono inicial aún no hay resultados estadísticos oficiales proporcionados por la universidad, por lo que hemos realizado una estimación utilizando las actas oficiales y la matriculación del siguiente curso académico. Según el cálculo realizado la tasa de aban-

dono inicial ha descendido de un 46 % a un 36 %. Como ocurría anteriormente, no se ha obtenido una mejora especialmente significativa, a pesar de la evolución positiva.

Las encuestas realizadas por la propia universidad, muestran una buena valoración del profesorado y la asignatura, por encima de 4 puntos. Es difícil de entender que si alumnado valora positivamente las actividades, la asignatura, el profesorado, y se encuentra motivado, los resultados académicos no mejoren de forma más visible.

5. Conclusiones y Trabajo Futuro

En esta comunicación hemos presentado la experiencia llevada a cabo en el marco de un proyecto de innovación docente para la creación de actividades y la implantación de una metodología, que permita mejorar la comprensión de la materia de programación.

El objetivo es el de ayudar a los alumnos a superar las asignaturas de iniciación a la programación, tratando de corregir las dificultades inherentes a la tarea, y mejorando el aprendizaje, así como la percepción que el alumno tienen del proceso de aprendizaje.

La metodología se ha implantado en la titulación de Grado en Ingeniería Informática. En la titulación concreta en la que hemos trabajado, actualmente hay un alto número de estudiantes que a pesar de estar en cursos superiores siguen sin adquirir las competencias necesarias en programación. Esto conlleva el fracaso en todas aquellas asignaturas cuyos requisitos previos incluyen habilidades relacionadas con la programación. Esta situación produce cierta frustración a los estudiantes, que tienen la sensación de no avanzar en su trayectoria académica, al acumular asignaturas pendientes de cursos anteriores.

Como parte de esta experiencia se han desarrollado actividades y material docente de apoyo a las metodologías activas introducidas en el aula, así como instrumentos de evaluación de la satisfacción de los alumnos.

A pesar de que los resultados académicos derivados de la metodología empleada son similares a los de años anteriores, sí que se ha visto una mejora clara en la motivación del alumnado, en la dinámica de clase, e incluso en la relación entre los estudiantes. En los siguientes años se tratará de identificar los principales problemas que lastran el rendimiento académico, visto que trabajar la motivación no ha dado los resultados esperados.

La metodologías y los materiales desarrollados son potencialmente aplicables a otras asignaturas de introducción a la programación de los distintos grados (Grado de Ingeniería de Tecnologías de Telecomunicación, Grado en Ingeniería Civil, etc...).

Agradecimientos

Este trabajo ha sido desarrollado en el marco del Proyecto de Innovación Docente del Vicerrectora de Ordenación Académica y Profesorado de la Universidad de Granada, con código PID-14-89, y parcialmente financiado por el proyecto P11-TIC-7460 de la Junta de Andalucía.

Referencias

[1] Andrea Forte. Programming for communication: overcoming motivational barriers to computation for all. En *IEEE Symposium on Human Centric Computing Languages and Environments, 2003. Proceedings.*, 285–286. IEEE, 2003.

[2] Balraj Kumar y Parul Khurana. Gamification in education-learn computer programming with fun. *International Journal of Computers and Distributed Systems*, 2(1):46–53, 2012.

[3] Raymond Lister, Elizabeth S. Adams, Sue Fitzgerald, William Fone, John Hamer, Morten Lindholm, Robert McCartney, Jan Erik Moström, Kate Sanders, Otto Seppälä, Beth Simon y Linda Thomas. A multi-national study of reading and tracing skills in novice programmers. En *ACM SIGCSE Bulletin*, volume 36, 119–150. ACM, 2004.

[4] Charlie McDowell, Linda Werner, Heather E. Bullock, y Julian Fernald. Pair programming improves student retention, confidence, and program quality. *Communications of the ACM*, 49(8):90–95, 2006.

[5] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman y Yasmin Kafai. Scratch: programming for all. *Communications of the ACM*, 52(11):60–67, 2009.

[6] John Sweller y Graham A. Cooper. The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction*, 2(1):59–89, 1985.

[7] Linda Werner, Brian Hanks, Charlie McDowell, Heather Bullock, y Julian Fernald. Want to increase retention of your female students. *Computing Research News*, 17(2):2, 2005.

[8] Jeannette M. Wing. Computational thinking. *Communications of the ACM*, 49(3):33–35, 2006.

[9] Leo Porter and Beth Simon. Retaining nearly one-third more majors with a trio of instructional best practices in cs1. In *Proceeding of the 44th ACM technical symposium on Computer science education*, pages 165–170. ACM, 2013.