

Descubriendo efectos inesperados en la programación en parejas

Discovering unexpected effects on pair programming

Noelia Sánchez-Marcano, Beatriz Pérez-Sánchez

noelia.sanchez@udc.es, beatriz.perezs@udc.es

Departamento de Computación
Universidade da Coruña
A Coruña, España

Resumen- Aprender a programar supone un esfuerzo importante e incluso un reto para muchos estudiantes dando lugar a elevadas tasas de fracaso. La programación en parejas (“pair programming”) es una actividad de aprendizaje colaborativa considerada en el entorno académico como una estrategia propicia para la enseñanza de programación. Distintos estudios demuestran los beneficios que genera la aplicación de esta técnica en un primer curso de programación. Por ello se aplica en la asignatura de Programación II, del primer curso del Grado en Ingeniería Informática de la Universidad da Coruña (España). El estudio realizado muestra que, a pesar de las motivaciones iniciales por parte del profesorado, los alumnos que conforman una misma pareja, de manera general, no se desenvuelven igual y exhiben notables diferencias de rendimiento académico.

Palabras clave: evaluación de prácticas, rendimiento académico, aprendizaje colaborativo, programación por pares

Abstract- Learning to program is a major effort and even a challenge for many students resulting in high rates of failure. Pair programming is a collaborative learning activity considered in the academic environment as a propitious strategy for teaching programming. Several studies show the benefits of applying this technique in a first programming course. Therefore, it is applied in the subject of Programming II at first course of the Degree in Computer Engineering of the University of A Coruña (Spain). The study carried out in this paper shows that, despite the initial motivations of the teaching staff, the students that form the same pair, in general, exhibit remarkable differences of academic performance.

Keywords: practical evaluation, academic performance, collaborative learning, pair programming

1. INTRODUCCIÓN

Las dificultades tanto en la enseñanza como en el aprendizaje de un curso introductorio de programación son ampliamente reconocidas dentro de la comunidad educativa de ciencias de la computación. Los estudiantes encuentran importantes inconvenientes a la hora de aprender a programar debido principalmente, a la necesaria capacidad de abstracción y resolución de problemas para solventar incluso el más trivial de todos ellos (Viekers, 2009). Aprender a programar supone un esfuerzo integrado e incremental debido a que cada tema de un curso introductorio de programación se basa en la información facilitada en los temas previos. Este hecho hace que muchos estudiantes se queden descolgados cuando no han afianzado suficientemente los conceptos base. La suma de

todos estos factores conlleva un descenso importante de la autoconfianza del estudiante a la hora de afrontar y completar las distintas tareas, que se traduce de manera generalizada en elevadas tasas de fracaso (Kinnunen y Simon, 2012), (Scott y Ghinea, 2013).

El desarrollo de software es un proceso que requiere esfuerzos coordinados de los miembros de uno o más equipos de trabajo. En consecuencia, es importante que los cursos de programación proporcionen a los estudiantes no solo conocimientos técnicos sino también, las habilidades necesarias para afrontar los diferentes proyectos a los que se tendrán que enfrentar en su vida laboral (Figl, 2010). Un problema que puede incrementar las dificultades de los estudiantes con la programación es la falta de un ambiente formalizado para el aprendizaje colaborativo entre compañeros. En la industria, todos los proyectos software son el resultado de un importante esfuerzo colaborativo. Los programadores profesionales recurren frecuentemente a la experiencia de sus colegas para resolver problemas. La capacidad de comunicación y trabajo en equipo resultan imprescindibles.

La programación en parejas (“pair programming”) es una actividad de aprendizaje colaborativa considerada en el entorno académico como una estrategia propicia para la enseñanza de programación. “Pair programming” es el término empleado para describir el proceso en el que dos programadores trabajan codo con codo, en el diseño y codificación de una tarea concreta. Se establecen dos roles principales, el “driver” que tiene el control del ratón y el teclado y el “navigator” que observa el trabajo del “driver”, a la vez que aporta sugerencias y correcciones. Ambos programadores deben intercambiar los roles y colaborar en el proceso completo desde el diseño a la implementación y posterior revisión. La extensa literatura referente a la aplicación de la estrategia de “pair programming” en educación de ciencias de la computación revela que, de forma generaliza, su implantación conlleva: (a) una mejora en la comprensión de los estudiantes, (b) un aumento de la autoconfianza a la hora de programar, (c) un incremento del rendimiento en las evaluaciones, (d) un mayor intercambio de conocimiento, y (e) la motivación necesaria para realizar un trabajo mejor (Salleh, Mendes y Grundy, 2011), (Hanks, Fitzgerald, McCauley, Murphy y Zander, 2011).

A mayores, vale la pena mencionar que los estudios realizados a lo largo de los años sobre la aplicación de “pair programming” en cursos de programación no solo persiguen analizar los beneficios directos, tales como la mejora del aprendizaje y los resultados académicos, sino también recoger y analizar las opiniones de los estudiantes sobre su experiencia con esta técnica. A modo de resumen se puede decir que en la mayoría de las investigaciones constatan el siguiente sentir general por parte de los estudiantes

- Actitud positiva hacia la colaboración y comunicación (Howard, 2007).
- Disfrute del trabajo en equipo (Zacharis, 2011).
- Aumento de la autoconfianza en el trabajo desarrollado (Braught, Wahls y Eby, 2011).
- Reducción de la frustración de los estudiantes (Braught, Wahls y Eby, 2011).
- Percepción de un mayor aprendizaje al trabajar con un compañero (Edwards, Stewart y Ferati, 2010).

Por todos estos motivos, en la asignatura de Programación II (primer curso del Grado en Ingeniería en Informática) las prácticas se desarrollan en parejas. En este trabajo analizaremos las calificaciones obtenidas durante dos cursos para comprobar si el rendimiento académico de los miembros de la misma pareja es similar. A pesar de que el peso concedido en la evaluación a las prácticas en laboratorio es muy importante (es indispensable aprobarlas para superar la materia), la mayor parte de la nota (80%) corresponde a una prueba objetiva que se realiza al final de curso. En esta prueba más de la mitad de los ejercicios están relacionados directamente con las prácticas de laboratorio, por lo que un buen trabajo en laboratorio debería suponer una buena nota en la materia. En este trabajo también comprobaremos si se cumple esta premisa analizando la relación existente entre las notas de la prueba objetiva y las notas de prácticas.

2. CONTEXTO

La asignatura Programación II se encuadra en el segundo cuatrimestre de primer curso de Ingeniería Informática de la Universidade da Coruña y, como su nombre indica, tiene una fuerte dependencia de la asignatura Programación I. Esta dependencia ocasiona que numerosos alumnos abandonen la materia al inicio de curso (Pérez-Sánchez y Sánchez-Marzoño, 2013) ya que tras no haber superado Programación I no cuentan con los conocimientos básicos para afrontarla.

El contenido de esta materia se centra principalmente en los Tipos Abstractos de Datos incluyendo: 1) Listas, 2) Colas, 3) Pilas y 4) Árboles. Así, las clases de prácticas obligan a la realización de prácticas de programación sobre estructuras de datos en un lenguaje de alto nivel. Durante el curso se realizan dos prácticas, la primera de ellas versa sobre listas (véase un ejemplo en la Tabla 1), la segunda es una extensión de la primera y suele incluir el tema de colas. La evaluación de las prácticas supone el 20% de la nota de la materia y ambas prácticas tienen el mismo valor. El 80% restante se consigue a través de una prueba objetiva a final de curso, aunque se puede conseguir un 10% adicional gracias a los trabajos de grupo reducido. En la prueba objetiva, 5,5 puntos (sobre 10) están estrechamente relacionados con la práctica (véase el ejemplo en la Tabla 2 que tenía un valor de 3 puntos).

Tabla 1. Ejemplo de enunciado de la práctica 1

“A lo largo de este curso implementaremos una aplicación que permite almacenar imágenes a usuarios registrados. Para su implementación necesitamos diseñar una estructura de datos que permita almacenar toda la información de los usuarios y su galería de imágenes asociada [...] Con tal objetivo emplearemos una **lista NO ordenada**. [...] poner en práctica el concepto de independencia de la implementación en los Tipos Abstractos de Datos (TADs). Por ello se pide crear dos implementaciones de una lista no ordenada, que deberán funcionar de manera intercambiable: una **estática** y otra **dinámica**”

Tabla 2. Ejemplo de ejercicio de examen

“Para gestionar la lista de imágenes se utilizaba el TAD List [...]. El primer cambio consiste en implementar esta lista como una **lista dinámica doblemente enlazada**. [...] realizar la implementación de findItem y deleteAtPosition”

Teniendo en cuenta las ventajas de la programación por parejas comentadas en la sección anterior, además del elevado número de alumnos (del orden de 400), la realización de las prácticas en esta materia es por parejas. Los alumnos tienen libertad para escoger compañero de prácticas dentro de su mismo grupo de prácticas. Además, para aquellos alumnos que no encuentren pareja en su grupo, se facilita un foro a través del entorno virtual de aprendizaje para que busquen en otros grupos de prácticas. En situaciones debidamente justificadas (por ejemplo, por motivos laborales), los alumnos pueden realizar las prácticas de modo individual. La situación que se espera es que aquellas parejas que realizan un buen trabajo de prácticas no deberían tener grandes dificultades para superar la prueba objetiva y por tanto, aprobar la materia. Como siempre las situaciones ideales se complican en la práctica, el estudio experimental realizado nos permitirá conocer las deficiencias de la estrategia docente planteada.

3. DESCRIPCIÓN

Los autores de este trabajo llevan impartiendo esta materia durante más de 10 años. Durante este periodo esta materia ha sufrido diversos cambios con el objetivo de mejorar continuadamente y lograr el mejor aprendizaje del alumno. Por ello, a pesar de disponer de datos de numerosos cursos académicos, se dispone de muy pocos que siguen el planteamiento actual de la materia. Así, en este trabajo usaremos los datos de los cursos 2014-2015 y 2015-2016. Con ellos, analizaremos las siguientes situaciones:

1. Correlación entre notas de examen y prácticas.
2. Correlación entre el ejercicio práctico del examen y prácticas.
3. Diferencias en la nota del examen entre miembros de la misma pareja de prácticas.

Para los dos primeros puntos se han usado los datos de todos los alumnos matriculados (398 en el curso 2014-2015 y 429 en 2015-2016). No obstante, para estudiar la correlación solo se ha considerado la nota del examen (o del ejercicio del examen) del alumno con mejor nota de la misma pareja de prácticas. Es

decir, los resultados que mostraremos en la siguiente sección se corresponden con el caso más favorable, donde se espera que una buena nota de prácticas conlleve una buena nota en el examen, o al menos, en el ejercicio relacionado con las prácticas. Para el tercer punto es decir, diferencias entre notas, se han considerado únicamente las parejas de prácticas, por ello, inicialmente se han descartado todos aquellos alumnos que realizaron la práctica de manera individual y, posteriormente, se eliminaron aquellos alumnos que, a pesar de haber formado una pareja de prácticas, no llegaron a presentarse al examen, así el número de parejas final es 82 y 91, respectivamente, para cada curso.

4. RESULTADOS

En esta sección mostraremos los resultados alcanzados en los dos cursos académicos considerados. Los resultados de las dos primeras subsecciones serán referentes a la correlación entre notas de prácticas y del examen o el ejercicio del examen, respectivamente. Se mostrarán de acuerdo a las categorías clásicas de Suspensos, Aprobados, Notables, Sobresalientes y Matrícula de Honor (MH) que se alcanzan igualando o superando la nota que se muestra en la Tabla 3. La categoría “NP” significa “No presentado”. Hay que señalar que las prácticas contemplan una categoría especial denominada “No Apto” que indica una práctica con fallos muy graves, en la mayoría de los casos imposible de corregir. Un “NP” o un “No Apto” en prácticas supone que la materia no está aprobada, independientemente de la nota alcanzada en el examen. Las notas de prácticas, cumpliendo con la normativa de la Universidad da Coruña, están disponibles 3 días antes del inicio del periodo de exámenes. Por tanto, los alumnos con un “No apto” en prácticas normalmente ya no se presentan al examen y, en aquellos casos que lo hacen, es para ver la dificultad de la prueba y suelen entregar en blanco (obsérvense las filas con “No apto” en prácticas de las Tablas 4 y 5).

Tabla 3. Nota para alcanzar cada categoría (entre paréntesis el valor máximo de cada prueba)

	Examen (10)	Ejercicio (2,5)	Prácticas (2)
Aprob	4,5	1,25	1
Not	7	1,75	1,34
Sobr	9	2,1	1,6
MH	9,5	2,4	1,9

La subsección C aborda la diferencia entre notas, para ella se emplearán las diferencias numéricas en la nota del examen de los dos miembros de una pareja de prácticas, teniendo en cuenta que un “No presentado” se ha valorado con un -0,1 para diferenciarlo del caso del 0. Es importante recordar que aquellos casos en los que ambos miembros de la pareja optan por no presentarse al examen se han descartado.

A. Correlación entre notas de examen y prácticas

Las Tablas 4 y 5 muestran las notas alcanzadas en el examen en contraposición a las notas de prácticas considerando solo aquellos alumnos que: a) hicieron la práctica individualmente o b) tienen mejor nota en el examen con respecto a su compañero de prácticas. La tabla no es cuadrada (hay más filas que columnas) debido a la categoría

especial de prácticas “No apto”. No obstante, excluyendo la fila de “No apto” en la nota de prácticas, se han marcado en negrita los valores de la diagonal principal. Una correlación perfecta debería reflejar valores positivos en dicha diagonal y 0 en los demás. Alcanzar estos niveles de correlación es casi imposible, sin embargo, en la Tabla 4 se puede apreciar que los mayores valores están en la diagonal principal y en sus proximidades (fila inferior y superior). Algo similar ocurre en la Tabla 5. Teniendo en cuenta que el examen tiene 5,5 puntos (de 10) estrechamente relacionados con las prácticas, hay valores de las Tablas 4 y 5 que llaman la atención y que nos han hecho plantearnos las siguientes preguntas:

- ¿Cómo alumnos con un Sobresaliente o MH en prácticas es decir, prácticas casi perfectas, obtienen un Suspensos en el examen (7 casos en la Tabla 4, y 5 en la Tabla 5)?
- Por el contrario, ¿Cómo alumnos con un suspensos en prácticas son capaces de alcanzar buena nota en el examen (1 notable en la Tabla 4 y 4 en la Tabla 5)?

Con el fin de aclarar estas dudas, hemos hecho el estudio de la pregunta del examen que se presenta a continuación.

Tabla 4. Correlación entre las notas del examen y las notas de prácticas. Curso 2014-2015

Nota prácticas	Nota examen					
	NP	Susp	Aprob	Not	Sobr	MH
NP	113	1	0	0	0	0
No Apto	55	3	0	0	0	0
Susp	1	21	15	1	0	0
Aprob	0	9	17	2	0	0
Not	0	0	0	0	0	0
Sobr	0	5	7	3	0	1
MH	0	2	1	1	0	0

Tabla 5. Correlación entre las notas del examen y las notas de prácticas. Curso 2015-2016

Nota prácticas	Nota examen					
	NP	Susp	Aprob	Not	Sobr	MH
NP	87	1	0	0	0	0
No Apto	73	2	0	0	0	0
Susp	3	11	14	4	0	0
Aprob	1	6	9	9	0	0
Not	0	4	4	1	0	0
Sobr	0	3	9	4	0	0
MH	0	2	4	8	1	0

B. Correlación entre notas en el ejercicio del examen y prácticas

La prueba objetiva de Programación II supone el 80% de la nota de esta asignatura, por el contrario, el trabajo práctico solo computa un 20%. El profesorado de la asignatura ha

discutido en numerosas ocasiones este porcentaje, pero por cuestiones que salen del ámbito de este artículo (plagio, autores fantasma, etc.), no se ha aumentado el valor del trabajo práctico. En contrapartida, el examen contiene una pregunta claramente práctica que se valora con 5,5 puntos sobre 10. Esta pregunta está dividida en dos apartados con un valor de 3 y 2,5 puntos, respectivamente. El primer apartado está relacionado con escribir el código de alguna de las operaciones más comunes de alguna estructura (por ejemplo, buscar un elemento o borrar una posición de una lista tal y como se ve en la Tabla 2), mientras que el otro implica el manejo de dichas estructuras, ignorando como están desarrolladas. Este último apartado está siempre relacionado con las prácticas realizadas durante el curso y suele consistir en añadir una nueva funcionalidad. Por ejemplo, siguiendo con la práctica comentada en la Tabla 1, los alumnos debían realizar varias funcionalidades como cargar o eliminar una imagen, denunciar una imagen por su contenido, etc. –cabe mencionar que en la práctica no se trabajaba con imágenes, solo con texto que representaba el nombre de un fichero que, hipotéticamente, contenía una imagen–. En la prueba objetiva, la función añadida suponía realizar la retirada de denuncias. Con esta similitud entre el contenido del trabajo práctico y el examen, es de esperar que aquellos alumnos con una buena evaluación en prácticas tengan una buena nota en el ejercicio del examen y, si la práctica no ha sido buena, la nota también debería ser acorde. Las Tablas 6 y 7 analizan este caso, recordemos que para su elaboración se ha tomado la mejor nota en el examen de los alumnos que conforman pareja de prácticas. De nuevo, se han marcado en negrita los valores de la diagonal principal (obviando la fila “No apto”) y se aprecia que los valores más altos se encuentran en dicha diagonal. No obstante, hay 19 alumnos en el curso 2014-2015 y 14 en el curso 2015-16 que, con una práctica suspensa o simplemente aprobada, realizan el ejercicio de manera brillante. También se encuentran situaciones en el extremo contrario, alumnos con prácticas con buenas calificaciones que, sin embargo, no realizan este ejercicio correctamente en el examen. Si bien los valores no parecen elevados, si prescindimos de los alumnos no presentados y con “No apto” en prácticas, tenemos que el 22% y el 15% de los alumnos en cada curso, respectivamente, realizan mucho mejor el ejercicio práctico que las prácticas y, un porcentaje similar en el curso 2015-16, aunque algo inferior en el 2014-15 se encuentran en el caso inverso, realizan mucho mejor las prácticas que el ejercicio del examen. Por tanto, la correlación inicialmente esperada parece difuminarse en vista de los resultados obtenidos.

C. Diferencias entre miembros de la misma pareja

Durante el desarrollo de las prácticas en laboratorio se realizan una serie de controles en el aula para, entre otras cosas, intentar asegurar que ambos miembros de una pareja están trabajando en la misma. Además, como hemos comentado previamente, en la materia que nos ocupa las prácticas están claramente ligadas al contenido teórico y, consecuentemente, a la prueba objetiva. Con todo esto, sería de esperar que las notas del examen de miembros de una misma pareja fueran relativamente similares. Las Figuras 1 y 2 demuestran que este hecho se cumple en muchos casos, véanse las columnas [0-1] y [1-2], pero también que hay notables diferencias ya que hay alumnos que alcanzan un Sobresaliente, mientras que su compañero ronda el 0 o ni se presenta. La diferencia media en las notas de examen de los miembros de la misma pareja, junto con la desviación típica, es $2,79 \pm 1,96$ y

$2,80 \pm 2,20$ para los cursos 2014-2015 y 2015-2016, respectivamente. Esta diferencia podría deberse al resto de ejercicios del examen, por ejemplo, preguntas relacionadas con el tema de las estructuras de árboles no consideradas en el desarrollo de la práctica. Por ello, hemos analizado las diferencias existentes en la pregunta estrictamente práctica, las Figuras 3 y 4 muestran los resultados obtenidos. La Figura 3 muestra abrumadoras diferencias, con un 64,63% de los alumnos con diferencias iguales o superiores a 1 punto (en un ejercicio con un valor máximo de 2,5 puntos). De modo más atenuado, la Figura 4 también ilustra estas diferencias, el 51,65% de las parejas tienen 1 o más puntos de diferencia entre la nota de sus integrantes en este ejercicio.

Tabla 6. Correlación entre las notas del ejercicio práctico del examen y las notas de prácticas. Curso 2014-2015

Nota prácticas	Nota ejercicio práctico examen					
	NP	Susp	Aprob	Not	Sobr	MH
NP	113	1	0	0	0	0
No Apto	55	2	0	0	0	0
Susp	1	15	8	4	0	10
Aprob	0	7	3	9	1	8
Not	0	0	0	0	0	0
Sobr	0	4	1	9	0	2
MH	0	2	0	1	0	1

Tabla 7. Correlación entre las notas del ejercicio práctico del examen y las notas de prácticas. Curso 2015-2016

Nota prácticas	Nota ejercicio práctico examen					
	NP	Susp	Aprob	Not	Sobr	MH
NP	87	1	0	0	0	0
No Apto	73	1	0	0	1	0
Susp	3	11	3	9	3	3
Aprob	1	6	7	3	3	5
Not	0	3	5	0	1	0
Sobr	0	6	3	5	1	1
MH	0	2	3	4	1	5

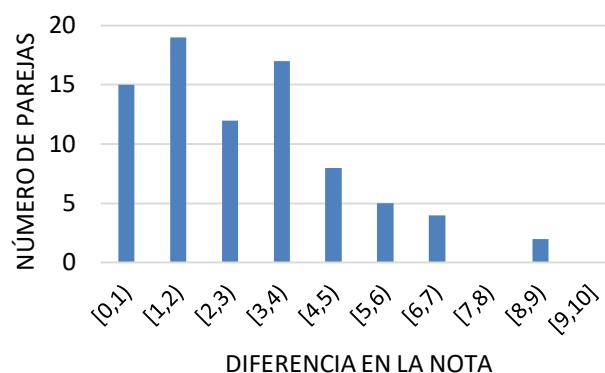


Figura 1. Diferencias en las notas del examen entre miembros de la misma pareja. Curso 2014-2015

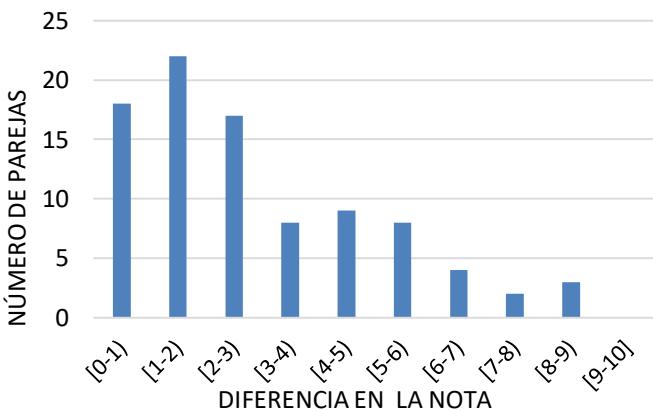


Figura 2. Diferencias en las notas del examen entre miembros de la misma pareja de prácticas. Curso 2015-2016

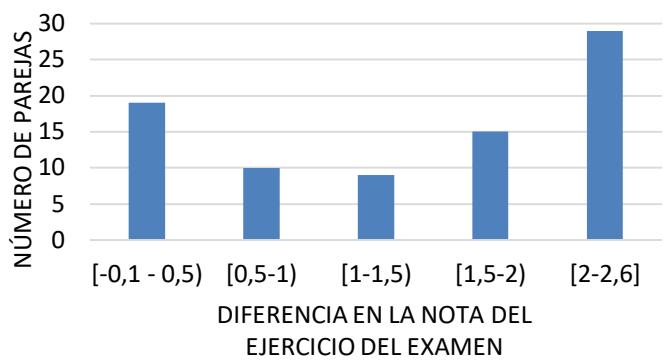


Figura 3. Diferencias en las notas del ejercicio práctico del examen entre miembros de la misma pareja de prácticas. Curso 2014-2015

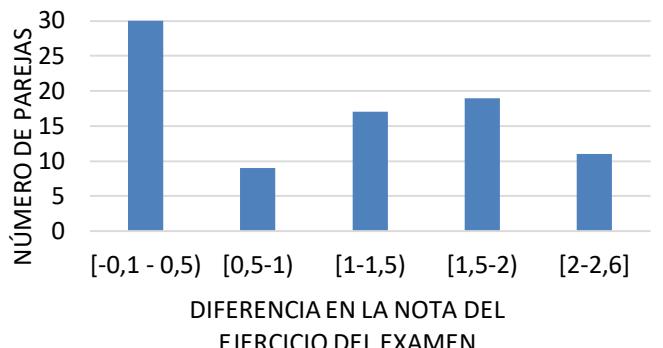


Figura 4. Diferencias en las notas del ejercicio práctico del examen entre miembros de la misma pareja de prácticas. Curso 2015-2016

5. CONCLUSIONES

A pesar de que diferentes estudios han demostrado que la técnica de “pair programming” contribuye positivamente al rendimiento y a la confianza de los estudiantes, el trabajo presentado muestra que también conlleva desventajas importantes. Las más destacables son: a) moderada correlación entre las notas de prácticas y las alcanzadas en la prueba objetiva y b) notables diferencias en la nota del examen

en los miembros de la pareja. Teniendo en cuenta los resultados obtenidos y con el objeto de establecer un plan de mejora continua en la docencia de la materia se extraen ciertas consideraciones que deben ser abordadas. Así, como líneas de trabajo futuro, sería interesante que el profesorado implemente medidas para asegurar que los miembros de la pareja contribuyen igualmente a la actividad del grupo. Además, resulta de importancia imponer un cambio de roles para motivar a los estudiantes. Otra cuestión a valorar es la rotación de parejas para evitar el vínculo entre los miembros del equipo, también conocido como “pair jellying” en “pair programming”.

REFERENCIAS

- Braught, G., Wahls, T., y Eby, L. M. (2011). The Case for Pair Programming in the Computer Science Classroom. *ACM Transactions on Computing Education*, 11(1), Article 2, 1-21.
- Edwards, R. L., Stewart, J. K., y Ferati, M. (2010). Assessing the effectiveness of distributed pair programming for an Online informatics curriculum. *ACM Inroads*, 1(1), 48-54.
- Figl, K. (2010). A systematic review of developing team competencies in information systems education. *Journal of Information Systems Education*, 21(3), 323-337.
- Hanks, B., Fitzgerald, S., McCauley, R., Murphy, L., y Zander, C. (2011). Pair programming in education: a literature review. *Computer Science Education*, 21(2), 153-173.
- Howard, E. V. (2007). Attitudes on using pair-programming. *Journal of Educational Technology Systems*, 35(1), 89-103.
- Kinnunen, P., y Simon, B. (2012). My program is ok – am I?. Computing freshmen’s experiences of doing programming assignments. *Computer Science Education*, 22(1), 1-28.
- Pérez-Sánchez, B., y Sánchez-Marcano, N. (2013). Disminuyendo la tasa de abandono y absentismo debida a la fuerte interrelación entre materias. *X Foro Internacional sobre la evaluación de la calidad de la investigación y de la educación superior (FECIES)*, Libro de Capítulos, pp. 1121-1125.
- Salleh, N., Mendes, E., y Grundy, J. C. (2011). Empirical studies of pair programming for CS/SE teaching in higher education: A systematic literature review. *IEEE Transactions on Software Engineering*, 37 (4), 509-525.
- Scott, M., y Ghinea, G. (2013). Educating programmers: A reflection on barriers to deliberate practice. *In Proceedings of 2nd HEA Conference on Learning and Teaching in STEM Disciplines*, 85-90.
- Vickers, P. (2009). *How to think like a programmer*. London, UK: Cengage Learning.
- Zacharis, N. Z. (2011). Measuring the effects of virtual pair programming in an introductory programming Java course. *IEEE Transactions on Education*, 54(1), 169-170.