



IMPACTOS DE LA INNOVACIÓN EN LA DOCENCIA Y EL APRENDIZAJE

Trabajo Autónomo y Videojuegos como refuerzo del aprendizaje lúdico en Programación

Una experiencia en el Grado de Ingeniería Informática

Campaña Gómez, Jesús Roque

Universidad de Granada

Departamento de Ciencias de la Computación e Inteligencia Artificial

C/ Daniel Saucedo Aranda, s/n 18071 Granada (España)

jesuscg@decsai.ugr.es

Pegalajar Cuéllar, Manuel

Universidad de Granada

Departamento de Ciencias de la Computación e Inteligencia Artificial

C/ Daniel Saucedo Aranda, s/n 18071 Granada (España)

manupc@decsai.ugr.es

Marín Jiménez, Ana Eugenia

Universidad de Granada

Departamento de Estadística e Investigación Operativa

Avenida de la Fuente Nueva S/N C.P. 18071 Granada (España)

anamarin@ugr.es

Ros Izquierdo, María

Universidad de Granada

Departamento de Ciencias de la Computación e Inteligencia Artificial

C/ Daniel Saucedo Aranda, s/n 18071 Granada (España)

Marosiz@decsai.ugr.es

Sánchez Fernández, Daniel

Universidad de Granada

Departamento de Ciencias de la Computación e Inteligencia Artificial

C/ Daniel Saucedo Aranda, s/n 18071 Granada (España)

daniel@decsai.ugr.es



IMPACTOS DE LA INNOVACIÓN EN LA DOCENCIA Y EL APRENDIZAJE

Medina Rodríguez, Juan Miguel

Universidad de Granada

Departamento de Ciencias de la Computación e Inteligencia Artificial

C/ Daniel Saucedo Aranda, s/n 18071 Granada (España)

Medina@decsai.ugr.es

Vila Miranda, María Amparo

Universidad de Granada

Departamento de Ciencias de la Computación e Inteligencia Artificial

C/ Daniel Saucedo Aranda, s/n 18071 Granada (España)

Vila@decsai.ugr.es

Ruiz Jiménez, María Dolores

Universidad de Granada

Departamento de Ciencias de la Computación e Inteligencia Artificial

C/ Daniel Saucedo Aranda, s/n 18071 Granada (España)

Mdruiz@decsai.ugr.es

Martín Bautista, María José

Universidad de Granada

Departamento de Ciencias de la Computación e Inteligencia Artificial

C/ Daniel Saucedo Aranda, s/n 18071 Granada (España)

mbautis@decsai.ugr.es

1. **RESUMEN:** Iniciarse en la programación requiere desarrollar y ejercitar diferentes habilidades. Dada la dificultad que supone al alumnado, hemos propuesto actividades y materiales de trabajo autónomo, entre los que se incluyen videotutoriales, y desarrollo de videojuegos para asignaturas de iniciación a la programación. El objetivo es mejorar la motivación y favorecer el aprendizaje. Describimos y analizamos los resultados de la experiencia realizada con estudiantes del Grado de Ingeniería Informática.
2. **ABSTRACT:** Programming is a complex task that requires the development and training of different skill sets.
Given the difficulty posed to the students, we propose diverse activities and provide autonomous learning materials that include video-tutorials and video game development activities, for courses of introduction to programming. With this, we



IMPACTOS DE LA INNOVACIÓN EN LA DOCENCIA Y EL APRENDIZAJE

intend to improve motivation and enhance learning. We describe an experience with students of the Computer Engineering Degree and analyze the results.

3. PALABRAS CLAVE: Informática, Programación, Trabajo Autónomo, Desarrollo de Videojuegos

KEYWORDS: Computer Science, Programming, Autonomous Learning, Game Development

4. DESARROLLO:

Aprender a programar es una tarea difícil que requiere la consecución de varios objetivos (Jenkins, 2002). El primer y principal objetivo es aprender las estructuras y sintaxis del lenguaje de programación a utilizar. Después se ha de comprender cómo esas estructuras del lenguaje permiten realizar operaciones. Si los estudiantes son capaces de asimilar por completo estos dos aspectos del aprendizaje de un lenguaje, ya tienen mucho terreno ganado, pero no podríamos decir que “saben programar”. La programación va más allá del simple hecho de aprender un lenguaje de programación, en realidad está más relacionada con la capacidad de resolver problemas. En concreto, podríamos decir que es la capacidad de resolver problemas utilizando un lenguaje de programación. Un buen programador es aquel que domina a la perfección los tres aspectos comentados, conoce la sintaxis del lenguaje, comprende el uso de las estructuras y es capaz de emplearlo para resolver un problema.

Es habitual que el alumnado manifieste en las primeras experiencias con la programación cierta frustración, ya que los primeros programas que se suelen hacer son bastante sencillos, utilizan cálculos matemáticos y tienen una presentación 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.

Dado que las estructuras empleadas en programación suelen ser las mismas para todos los lenguajes, y sólo varían algunos aspectos sintácticos, proponemos desacoplar el aprendizaje del lenguaje del aprendizaje de las estructuras, y aprender éstas últimas a través de videojuegos.

En 1962 Perlis, manifestó su opinión acerca de que todo el mundo debería aprender a programar como parte de su educación. Esta idea partía de la comprensión de que en el futuro la mayor parte de las tareas estarían ejecutadas en forma de procesos por máquinas. Esta idea visionaria está cada vez más cerca de la realidad gracias a un movimiento educativo que intenta impulsar el aprendizaje del Pensamiento Computacional (Computational Thinking - CT) desde los niveles más básicos del sistema educativo. Jeannette M. Wing en 2006 definió el pensamiento computacional como un acercamiento a la resolución de problemas, diseño de



IMPACTOS DE LA INNOVACIÓN EN LA DOCENCIA Y EL APRENDIZAJE

sistemas y comprensión del comportamiento humano que usa los conceptos fundamentales de la computación. En los últimos años se han desarrollado grandes cantidades de herramientas de apoyo al pensamiento computacional (Guzdial, 2008). Google y el Instituto Tecnológico de Massachusetts (MIT) han apostado por la idea de enseñar los fundamentos básicos de la computación desde el jardín de infancia hasta la universidad. A través de cursos de formación (<https://computationalthinkingcourse.withgoogle.com/>) y herramientas de programación para jóvenes como Scratch (Resnick et al., 2009 - <https://scratch.mit.edu/>), se ha abierto la puerta a multitud de iniciativas que no dejan de aportar nuevas herramientas que usar en la práctica docente.

El uso de entornos visuales en programación facilita el proceso de aprendizaje, y lo dota de un carácter más lúdico. Habitualmente en los estudios universitarios el profesorado se centra en exceso en el contenido, sin tener en cuenta la necesidad de captar la atención del alumnado y sin preocuparse en exceso en aumentar la motivación. Sin embargo, trabajando estos y otros aspectos podemos llegar a lograr un aprendizaje más eficaz (Moore et al., 2012).

Con el objetivo de motivar al alumnado y hacer ameno el aprendizaje, nos hemos propuesto integrar en la metodología docente de las asignaturas de programación el desarrollo de videojuegos (Overmars, 2004).

Metodología

La metodología empleada está pensada para ser aplicada en los cursos de iniciación a la programación del Grado en Ingeniería Informática, para las asignaturas Fundamentos de Programación y Metodología de la Programación.

Para aplicar la metodología se seleccionó un grupo de primer curso en el que se encontraban matriculados 25 estudiantes en Fundamentos de Programación y 27 en Metodología de la Programación. La mayor parte del alumnado era de nuevo ingreso, por lo que estaban matriculados en ambas asignaturas, contabilizando un total de 32 alumnos únicos entre ambas.

La asignatura de Fundamentos de Programación se imparte en el primer semestre, por lo que es habitual que los estudiantes no tengan ningún conocimiento previo sobre programación.

Para iniciarlos en la programación, comenzaremos con el aprendizaje de las estructuras comunes a los lenguajes de programación. Para llevarlo a cabo, emplearemos algunas de las muchas herramientas desarrolladas a partir del auge de la enseñanza del pensamiento computacional. A través de sencillos ejercicios como los presentados en la Hora de Código (<https://code.org/>) se aprenden los elementos básicos comunes a todo lenguaje, los cuales están representados como piezas de un puzzle que encajan entre sí para formar programas de forma visual. Una vez los estudiantes están familiarizados con esta forma de crear programas,



IMPACTOS DE LA INNOVACIÓN EN LA DOCENCIA Y EL APRENDIZAJE

se puede pasar a trabajar la resolución de problemas con otros recursos como por ejemplo Blockly Games (Fraser, 2013 - <https://blockly-games.appspot.com/>).

Estas herramientas son apropiadas para comprender las estructuras de un lenguaje de programación y para ejercitar la capacidad de resolución de problemas abstrayéndonos del lenguaje concreto en que se programa.

Aunque este es un buen comienzo, no colma todas las necesidades de un alumno de programación de Ingeniería Informática. En algún momento se ha de pasar a la codificación de programas empleando código. Para evitar que la transición sea muy brusca, a lo largo del curso los estudiantes reciben tareas y ejercicios que deben de resolver de forma autónoma tanto utilizando estas herramientas visuales como empleando código.

A través de los materiales de trabajo autónomo, permitimos al alumnado trabajar a su propio ritmo y que puedan mantenerse al día con los contenidos de la asignatura. A través de los guiones de trabajo autónomo, el alumnado aprende a resolver problemas usando el lenguaje de programación C++. Algunos de los problemas planteados en los guiones, consisten en la elaboración de sencillos videojuegos clásicos como las 3 en raya, el ahorcado, etc. Durante el desarrollo de la asignatura las interfaces de salida de los programas elaborados en C++ serán en modo texto, puesto que para desarrollar un entorno gráfico necesitan más conocimientos. En esta primera aproximación al desarrollo de videojuegos, se hará más énfasis en la creación y gestión de la lógica del juego, que en el aspecto visual.

El sistema de evaluación de esta primera asignatura sigue el esquema habitual en el que hay una parte teórica que se evalúa a través de exámenes, y una práctica, que se evalúa con la entrega de los ejercicios propuestos en los guiones de trabajo autónomo.

La asignatura de Metodología de la Programación se desarrolla en el segundo semestre del curso académico. El alumnado que haya superado la asignatura de Fundamentos de Programación debería de conocer los rudimentos básicos de la programación y ser capaz de resolver pequeños problemas. En esta asignatura se introducen conceptos más complejos de programación y el tamaño de los problemas aumenta, hasta el punto de plantearse como pequeños proyectos de programación. Además, se les introducirá en la creación de interfaces gráficas básicas, a través del uso de librerías.

Como mecanismo de recuperación para aquellos alumnos que no hubiesen superado la asignatura de programación en el primer semestre, se han impartido seminarios en los que se volvía a incidir sobre los conceptos básicos previamente estudiados, pero esta vez con aplicación a la robótica usando los robots Lego Mindstorms (<http://mindstorms.lego.com/>) y el software NXT que se emplea para programarlos.



IMPACTOS DE LA INNOVACIÓN EN LA DOCENCIA Y EL APRENDIZAJE

Para la evaluación de la parte teórica de esta asignatura introducimos una variación, dado que el alumnado posee conocimientos previos de programación, se propone la realización de guiones de trabajo autónomo, que en caso de ser evaluados de forma satisfactoria, conformarán la calificación de teoría y por tanto eximirá a los estudiantes que los hayan completado de realizar el examen correspondiente a la parte teórica. Los guiones de trabajo autónomo se complementan con videotutoriales donde se explica cómo se debe de resolver un problema y con ejercicios resueltos de convocatorias anteriores.

Para la evaluación de la parte práctica de la asignatura se ha de elaborar un proyecto consistente en el desarrollo de videojuegos. El proyecto se divide en dos partes, el desarrollo guiado de un videojuego, y el desarrollo autónomo de otro videojuego.

En la primera parte se entregará a los estudiantes una librería gráfica de Software Libre para desarrollo de videojuegos, y guiones que explican como se debe de desarrollar el proyecto, así como código que debe completar el alumnado. Esta primera etapa trata de instruir en el uso de la librería gráfica y muestra la forma correcta de estructurar los datos a los que tendrá acceso el videojuego. Siguiendo las instrucciones de los guiones el alumnado podrá construir de forma guiada un videojuego desde cero, desarrollando por el camino las habilidades necesarias para poder hacerlo de forma autónoma.

La segunda parte del proyecto, consiste en el desarrollo de un videojuego que se abordará desde el inicio, poniendo en práctica todo lo aprendido en los guiones de la primera parte.

En nuestro caso como juego de desarrollo guiado elegimos Candy Crush (<http://candycrushsaga.com/>), un juego muy conocido por todos, por lo que es mucho más fácil el desarrollo, ya que la mecánica de juego es conocida. Además se les proporcionaron todos los elementos gráficos necesarios (legales y gratuitos, aprovechando para realizar un breve seminario acerca de los derechos de copyright, copyleft y creative commons) para darle un aspecto profesional al juego como puede verse en la Figura 1.

Para el proyecto de videojuego autónomo los estudiantes debían realizar el juego de las damas, siendo en esta ocasión los responsables de buscar los elementos gráficos y de organizar e implementar el programa.

En el Grado de Ingeniería Informática consideramos esencial que desde los primeros cursos, el alumnado piense en su futuro profesional e identifique posibles caminos a emprender. Para reforzar esta actitud, periódicamente se llevan al aula ofertas de trabajos reales, para que el alumnado pueda conocer los perfiles solicitados. Junto a la realización del proyecto guiado, les suministramos una oferta de trabajo real de King, la empresa creadora de Candy Crush, en la que buscaban desarrolladores para sus videojuegos que programasen en C++, el mismo lenguaje usado en el proyecto.



IMPACTOS DE LA INNOVACIÓN EN LA DOCENCIA Y EL APRENDIZAJE

Todas las actividades y materiales que se han desarrollado están orientados a mejorar la confianza del estudiantado y actuar como motivación, fomentando la implicación con la asignatura y la programación en general. Consideramos que esta actitud positiva, junto a la competencias alcanzadas, debe trasladarse a los resultados académicos.

Evaluación y Seguimiento

Para evaluar la efectividad de los materiales realizados y de nuestra propuesta, hemos planteado una evaluación basada en encuestas de opinión del alumnado, utilizando preguntas con respuestas de tipo Likert de 5 puntos (Ary et al., 2013). En la encuesta se presentaba la afirmación “Este tipo de material / actividad me ha resultado muy útil”, para valorar los distintos tipos de materiales suministrados y las actividades realizadas, y se ofrecían las siguientes posibilidades de respuesta “1 - Totalmente en desacuerdo, 2 - Algo en desacuerdo, 3 - Indiferente, 4 - Algo de acuerdo, 5 - Totalmente de acuerdo”.

Los materiales presentados a evaluación son aquellos que se han desarrollado para ambas asignaturas (ejercicios resueltos y videotutoriales, guiones de trabajo autónomo, proyectos prácticos basados en videojuegos y seminarios de recuperación usando NXT y robots Lego), y los usados tradicionalmente (material de clase para teoría, bibliografía recomendada y exámenes de años anteriores).

Para el análisis se ha empleado la moda como medida de tendencia central (Boone y Boone, 2012). Para poder realizar el seguimiento de la evolución de las opiniones a lo largo del tiempo y las distintas asignaturas manteniendo el anonimato, se realizaron preguntas de control para verificar la identidad (grupos de música favoritos, ciudades a las que les gustaría viajar y el motivo, etc).

En la Figura 2, se muestra la moda de las respuestas del alumnado para cada uno de los items sobre los que se realizó la consulta.

Atendiendo a estos resultados, podemos comprobar que los materiales mejor valorados son los ejercicios resueltos y los videotutoriales de apoyo. Han sido también valorados de forma positiva, los ejercicios de trabajo autónomo entregados como parte de la evaluación de la teoría de la asignatura, y los seminarios de recuperación empleando el software NXT para programar los robots de Lego. El resto de materiales no han sido considerados de especial utilidad por el alumnado. Podemos observar que los nuevos materiales y actividades, con la excepción de los proyectos prácticos, han recibido una mejor valoración que los usados tradicionalmente.

Merece la pena detenerse a analizar con más detalle el caso de los proyectos prácticos. Como ya se ha indicado, estos consistían en la elaboración de videojuegos con interfaz gráfico, un tema por el que muchos se habían interesado a principio de curso. Sin embargo, los resultados



IMPACTOS DE LA INNOVACIÓN EN LA DOCENCIA Y EL APRENDIZAJE

de las opiniones analizadas no reflejan ese interés. Aunque el valor de la moda es 3, la mayoría de las valoraciones se reparten entre los valores 3 y 4, por lo que consideramos que la valoración es moderadamente positiva, aunque no lo es tanto como se esperaba a la hora de plantear la actividad. La posible causa de esta valoración, puede venir dada por la complejidad que alcanza la asignatura y que se refleja en la realización del proyecto. Esa complejidad puede ser demasiado alta para aquellos alumnos que no superaron la primera asignatura de programación. Esta explicación también encajaría con la popularidad alcanzada en las encuestas por los seminarios de recuperación usando NXT.

Durante el análisis de los datos no se observaron diferencias en la opinión del alumnado sobre los materiales al realizar contrastes por sexo y experiencia previa con programación.

En cuanto a los resultados académicos, hemos comparado el grupo en que hemos aplicado la nueva metodología con el grupo del año académico anterior en el que no se aplicó ninguna de las nuevas actividades. La tasa de éxito (número de alumnos aprobados de entre los presentados) subió ligeramente desde un 38% a un 47% al aplicar la metodología en la asignatura de Fundamentos de Programación, y de un 44% a 75% en Metodología de la Programación. Sin embargo, el número de alumnos no presentados sigue siendo muy alto, lo que hace que las tasas de rendimiento (número de alumnos aprobados del total de matriculados) sean pequeñas. Para Fundamentos de Programación la tasa de rendimiento subió de un 21% a un 28%, y para Metodología de la Programación lo hizo de un 21% a un 22%.

Aunque la mejora en el rendimiento académico es mínima, confiamos en que la tendencia siga al alza en cursos siguientes.

La valoración global del desarrollo de las asignaturas ha sido calificada como positiva por parte del alumnado. Si bien es cierto que han destacado aspectos a mejorar, principalmente relacionados con la realización de los proyectos prácticos. El alumnado se queja de la cantidad de tiempo necesario que se ha de invertir en el desarrollo de estas actividades prácticas. Si bien es cierto que realizar los proyectos requiere tiempo y dedicación, se les ha dado la posibilidad de entregar al final de la asignatura cualquier actividad que no hubiese sido entregada a tiempo, lo cual debería de eliminar el problema. En las sesiones de tutoría tanto grupales como individuales, los estudiantes han manifestado su opinión acerca de la complejidad de las proyectos prácticos, los cuales creen abarcan mayor complejidad que los conocimientos adquiridos.



IMPACTOS DE LA INNOVACIÓN EN LA DOCENCIA Y EL APRENDIZAJE

Conclusiones y Trabajo Futuro

La metodología propuesta ha sido acogida con entusiasmo por el alumnado, que ha manifestado en sesiones de tutoría su motivación con la asignatura. Esto también se ha visto reflejado en las sesiones de clase prácticas, donde el ambiente de trabajo y la implicación ha sido mayor que en años anteriores.

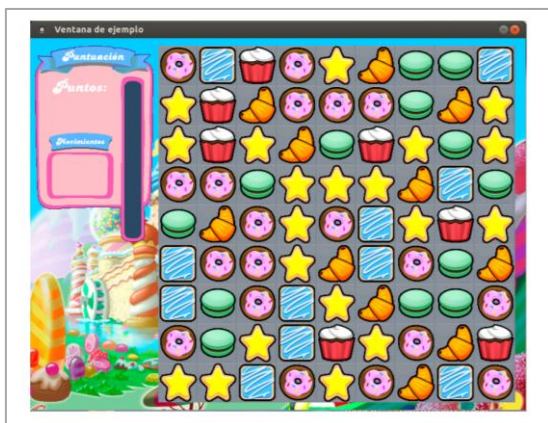
El alumnado ha respondido positivamente a algunas de las actividades propuestas y materiales desarrollados, tales como los guiones de trabajo autónomo, y los seminarios de recuperación basados en la programación de robots con el software NXT, y ha valorado especialmente la realización de videotutoriales y los ejercicios resueltos.

Aunque la mejora a nivel académico no ha sido especialmente reseñable, los resultados obtenidos nos animan a seguir en esta línea de trabajo, aunque para próximos cursos intentaremos potenciar las actividades que han tenido más aceptación y trataremos de hacerles ver la importancia que tiene la programación no sólo en su futuro laboral, sino también en su presente académico.

Agradecimientos

Este trabajo ha sido desarrollado en el marco del Proyecto de Innovación Docente del Vicerrectorado 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.

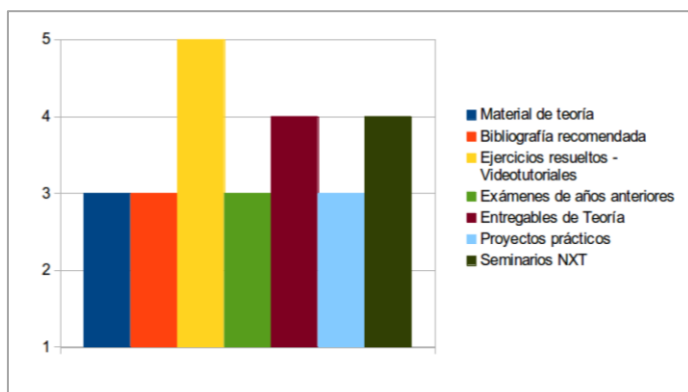
4.1. FIGURA O IMAGEN 1





IMPACTOS DE LA INNOVACIÓN EN LA DOCENCIA Y EL APRENDIZAJE

4.2. FIGURA O IMAGEN 2



5. REFERENCIAS BIBLIOGRÁFICAS

Ary, D., Jacobs, L., Sorensen, C., y Walker, D. (2013). Introduction to research in education (Ninth Edition). Wadsworth Cengage Learning.

Boone, H. N., y Boone, D. A. (2012). Analyzing Likert Data, Journal of extension, 50(2), 1-5.

Fraser, N. (2013). Blockly: A visual programming editor.

<https://developers.google.com/blockly/> (Último acceso: 15/05/2016)

Guzdial, M. (2008). Education Paving the way for computational thinking. Communications of the ACM, 51(8), 25-27.

Jenkins, T., (2002). On the difficulty of learning to program. In Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences. pp. 53 -58.

Moore, S., Walsh, G. y Rísquez, A. (2012). Estrategias eficaces para enseñar en la universidad. Narcea Editores.

Overmars, M. (2004). Teaching computer science through game design. Computer, 37(4), 81-83.

Perlis, A. (1962) The computer in the university. In M. Greenberger, Ed., Computers and the World of the Future, MIT Press, Cambridge, MA, pp. 180 -219.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver J., Solveman, B. y Kafai, Y. (2009). Scratch: programming for all. Communications of the ACM, 52(11), 60-67.



IMPACTOS DE LA INNOVACIÓN EN LA DOCENCIA Y EL APRENDIZAJE

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.

Recursos en la red:

<https://computationalthinkingcourse.withgoogle.com/> (Último acceso: 15/05/2016)

<https://scratch.mit.edu/> (Último acceso: 15/05/2016)

<https://code.org/> (Último acceso: 15/05/2016)

<https://blockly-games.appspot.com/> (Último acceso: 15/05/2016)

<http://mindstorms.lego.com/> (Último acceso: 15/05/2016)

<http://candycrushsaga.com/> (Último acceso: 15/05/2016)