

Lecciones aprendidas de experiencias con robots educativos y entornos de programación visuales en asignaturas de programación

Felipe I. Anfurrutia, Ainhoa Álvarez, Mikel Larrañaga y Juan-Miguel López-Gil

Departamento de Lenguajes y Sistemas Informáticos, Universidad del País Vasco UPV/EHU,
C/ Nieves Cano 12, 01006, Vitoria-Gasteiz, España.

{felipe.anfurrutia, ainhoa.alvarez, mikel.larranaga, juanmiguel.lopez}@ehu.eus

Resumen: Las asignaturas introductorias a la programación presentan muchos retos, tanto para el alumnado que las cursa como para el profesorado que las imparten. Diversos autores plantean la incorporación de robots educativos o entornos visuales de programación como medio para facilitar su aprendizaje. En este artículo se presentan tres experiencias realizadas en un marco pedagógico basado en el ciclo de aprendizaje de Kolb y utilizando este tipo de herramientas. Las experiencias se han realizado en las asignaturas de programación (*Programación Básica y Programación Modular y Orientación a Objetos*) del primer curso del Grado en Ingeniería Informática de Gestión y Sistemas de Información en la Escuela de Ingeniería de Vitoria-Gasteiz. A partir de estas experiencias, el artículo presenta un conjunto de lecciones aprendidas y como consecuencia una propuesta de integración de entornos para dar respuesta a la problemática identificada. Se ha realizado una primera evaluación de la propuesta que también se presenta en este artículo junto con los resultados positivos de la misma.

Palabras clave: Software educativo; experiencia educativa; robots educativos; programación

Abstract: Introduction to programming modules are challenging both for lecturers and for students. Several authors propose the inclusion of educational robots or visual programming environments as a means to facilitate learning. This paper presents three experiences carried out implementing a pedagogical frame, which relies on the Kolb's learning cycle, and using the above-mentioned kinds of tools. The experiences have been carried out in two programming modules (Programming fundamentals and Object oriented programming) of the first year of the computer studies grade at the Engineering school of Vitoria-Gasteiz. The paper presents a set of lessons learned from those experiences and, as a consequence, a proposal of tool integration in order to overcome the problems identified in the experiences. A preliminary evaluation of the proposal has been conducted and its results are also presented in this paper.

Key words: Educational Software, Educational Experience, Educational Robots, Programming

1. Introducción

Uno de los principales problemas en la docencia de las asignaturas introductorias a la programación es el elevado grado de heterogeneidad que presenta el alumnado en cuanto al conocimiento previo de las competencias abordadas en dichas asignaturas [Gomes et al. 07]. Este hecho dificulta que el profesorado pueda diseñar métodos de aprendizaje adecuados para todo el alumnado [Leonard 02].

Además, las asignaturas de programación son habitualmente impartidas utilizando lenguajes de programación de propósito general, que pueden resultar muy complejos para estudiantes noveles sin conocimientos previos de la materia [Gomes et al. 07, Hirst et al. 03]. Algunos lenguajes de programación tienen una curva de aprendizaje muy alta mientras que otros, incluso para los programas más sencillos, requieren escribir una cantidad de código que resulta complicada de entender y abordar para el alumnado

novel. En general, el alumnado tiene que hacer frente a la vez tanto a la construcción de los algoritmos como a las reglas sintácticas de los lenguajes de programación empleados.

En la bibliografía se abordan dos formas principales para intentar paliar los problemas que el alumnado novel se encuentra en las asignaturas de programación [Wu et al. 08]. Por una parte, se han empleado entornos visuales para el aprendizaje de la programación. Este tipo de entornos abstraen al alumnado de las complejidades de los lenguajes de programación empleados permitiendo que se centren en la comprensión de los conceptos fundamentales antes de comenzar a programar [Wilson et al. 10]. Por otra parte, se plantea el uso de dispositivos físicos que permiten una aproximación en la que el alumnado puede interactuar con sus programas en el mundo real [O'Sullivan et al. 04].

Los autores de este artículo han explorado ambas aproximaciones en los últimos años. Sin embargo, el mero hecho de introducir estos elementos en una asignatura no asegura mejoras [Sartatzemi et al. 08], por lo que su implantación se ha realizado en combinación con el ciclo de aprendizaje de Kolb [Kolb 84]. Estas exploraciones se han realizado durante 5 cursos académicos en las dos asignaturas de programación que se imparten en el primer curso del Grado en Ingeniería en Informática de Gestión y Sistemas de Información en la Escuela de Ingeniería de Vitoria-Gasteiz: *Programación Básica*, durante el primer cuatrimestre, y *Programación Modular y Orientación a Objetos* (PMOO) en el segundo. Además, se debe tener en cuenta que en la definición del grado se determinó que el lenguaje de programación a emplear en las diferentes asignaturas de programación sería Java. Este aspecto no se puede modificar, lo que condiciona las innovaciones educativas que se pueden llevar a cabo en las diferentes experiencias.

El resto del artículo se organiza de la siguiente manera. En primer lugar, se describe el diseño general de las experiencias, seguido de la descripción de las experiencias concretas realizadas. A continuación se presentan los resultados obtenidos a partir de los estudios realizados sobre dichas experiencias. A partir de los resultados de estas experiencias, se detallan un conjunto de aspectos

derivados que conviene considerar al emplear cualquiera de las opciones presentadas. Para dar respuesta a los problemas detectados en los estudios, se presenta una propuesta de integración de entornos y la evaluación preliminar de la misma. Finalmente, se presentan las conclusiones del trabajo.

2. Diseño general de las Experiencias

El aprendizaje simultáneo de los aspectos conceptuales de la programación y de las reglas sintácticas de los lenguajes de programación supone una dificultad añadida para el alumnado novel [Gomes et al. 07]. Para mitigar este problema, los autores han definido un marco pedagógico para las asignaturas de programación tomando como punto de partida el ciclo de aprendizaje de Kolb e incorporando robots educativos y entornos visuales.

A continuación, se describen el marco pedagógico general, la selección de herramientas de apoyo realizada y el protocolo seguido para la recopilación de los datos en las diferentes experiencias.

2.1. Marco pedagógico

Cualquier innovación docente requiere la aplicación de una teoría de aprendizaje para que su diseño sea adecuado y satisfaga las necesidades del alumnado. En este sentido, algunos autores han sugerido abordar en primer lugar los aspectos conceptuales de la programación, para posteriormente trabajar los aspectos notacionales asociados a un lenguaje de programación concreto [Zhu et al. 03]. Además, en el caso de la programación, varios autores han apuntado que las situaciones prácticas, en las que los estudiantes construyen el conocimiento de manera activa, son las más adecuadas [Lahtinen et al. 05, Wang et al. 12].

Teniendo en cuenta estas recomendaciones, los autores decidieron trabajar los contenidos desde dos perspectivas diferentes: primero desde una perspectiva conceptual y posteriormente desde una perspectiva notacional. El objetivo de la perspectiva conceptual es que el alumnado comprenda y sea capaz de aplicar los contenidos abordados en la resolución de problemas mientras que la perspectiva

notacional considera los aspectos sintácticos y semánticos de un lenguaje de programación concreto o de formalismos como los diagramas de flujo. Los autores determinaron aplicar, en cada una de las perspectivas, el ciclo de aprendizaje de Kolb [Kolb 84], que ha demostrado ser de gran utilidad para el aprendizaje de la programación [Yan 09]. Kolb plantea que para que el aprendizaje sea efectivo el alumnado debe realizar un ciclo de las cuatro etapas mostradas en la Figura 1. En primer lugar, debe realizar una actividad concreta. Después, debe reflexionar sobre la experiencia, para poder luego conceptualizar la teoría que permite explicar las observaciones realizadas. Finalmente, debe aplicar la teoría en situaciones nuevas.

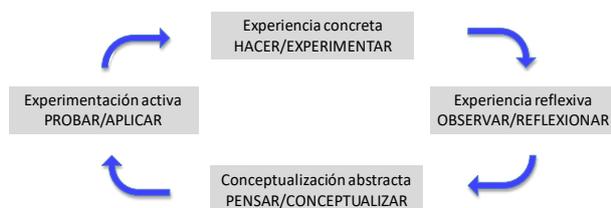


Figura 1 Ciclo de aprendizaje de Kolb

En esta propuesta se plantea abordar cada una de las etapas que conforma el ciclo de aprendizaje de Kolb con los tipos de actividades que se detallan a continuación.

- 1 **Experiencia concreta:** El objetivo principal de esta etapa es involucrar activamente al alumnado en actividades concretas relacionadas con los temas que se van a tratar. Para ello, se proporcionarán programas de ejemplo para que el alumnado pueda experimentar con ellos, tanto ejecutándolos como realizando modificaciones en ellos.
- 2 **Experiencia reflexiva:** Una vez que se ha realizado una experiencia concreta es necesario que el alumnado reflexione sobre dicha experiencia. Para ello, las actividades diseñadas para la etapa de experiencia concreta se han complementado con series de preguntas que inducen al alumnado a reflexionar sobre los conceptos abordados. Así mismo, el profesorado fomenta la discusión en grupo sobre lo observado en la etapa anterior de experiencia concreta.

- 3 **Conceptualización abstracta:** En esta etapa los contenidos abordados se trabajan de manera formal bajo la supervisión del profesorado. Para ello, el profesorado proporciona ejemplos relacionados con los temas tratados en las experiencias realizadas en las dos etapas anteriores que permiten al alumnado asimilar los temas abordados.
- 4 **Experimentación activa:** En la última etapa se le plantean al alumnado nuevos escenarios para poner en práctica y consolidar los temas tratados en las etapas anteriores.

En las experiencias que se describen en este artículo se ha aplicado este ciclo de aprendizaje en cada uno de los temas de la asignatura seleccionados para cada una de las experiencias.

2.2. Selección de herramientas

Para llevar a cabo cada una de las experiencias se realizó un estudio de las herramientas existentes para determinar aquellas que mejor se adaptaran a las características de cada asignatura, facilitarán al alumnado el aprendizaje y minimizarán los problemas identificados en la bibliografía. Para ello se analizaron herramientas de los dos principales grupos identificados en la bibliografía: robots educativos y entornos de programación visual [Wu et al. 08, Wilson et al. 10, O'Sullivan et al. 04]. Para la selección se tuvieron en cuenta comparativas publicadas [Hirst et al. 03, Georgantaki et al. 07] así como el soporte que daban para las perspectivas conceptual y notacional.

En el caso de los robots educativos, entre la diversidad de opciones existentes, se optó por utilizar robots Lego MINDSTORMS®. Una de las razones principales para esta selección es la flexibilidad que tienen a la hora de utilizar diferentes entornos para programarlos y la familiaridad que tienen los estudiantes con ellos, ya que muchos de los estudiantes han interactuado con Lego en su infancia.

En el caso de los entornos de programación visual, se seleccionaron diferentes herramientas en cada asignatura. En el caso de *Programación Básica*, el objetivo era aprender los aspectos básicos para el diseño y la implementación de programas sencillos,

por lo que se optó por Scratch (<https://scratch.mit.edu/>). Esta herramienta se diseñó para inculcar nociones de programación a una audiencia más joven, pero se ha utilizado con éxito en instituciones educativas superiores [Maloney et al. 10]. En cambio, en la asignatura de *Programación Modular y Orientación a Objetos* el objetivo era trabajar con los conceptos principales de la Orientación a Objetos. En este caso, se optó por trabajar la primera mitad del curso con BlueJ y la segunda mitad con Greenfoot. Estas dos herramientas han demostrado su valía para la enseñanza en éste área [Georgantaki et al. 07] y, además, dan soporte para la perspectiva notacional, ya que han sido desarrolladas para el aprendizaje de la Programación Orientada a Objetos en Java

2.3. Recopilación de datos

Para evaluar el efecto de las innovaciones realizadas y la percepción de los actores involucrados, se han realizado estudios cualitativos y cuantitativos en cada una de las experiencias. La información recabada se ha obtenido mediante cuestionarios en los que el alumnado valoró las herramientas utilizadas y su contribución en el aprendizaje de los diferentes conceptos relacionados con las asignaturas. Así mismo, en alguno de los casos se han realizado análisis estadísticos para determinar si el uso de las herramientas seleccionadas tienen efecto en los resultados o calificaciones.

3. Experiencias realizadas

En total se realizaron tres experiencias en dos asignaturas de programación. A continuación se describe cada una de las experiencias realizadas, reflejando cómo se aplicó el marco pedagógico basado en el ciclo de aprendizaje de Kolb y cómo se integraron las herramientas seleccionadas.

3.1. Experiencia 1

Esta experiencia se diseñó para la asignatura *Programación Básica* con el objetivo de dar respuesta a uno de los principales problemas que presenta su alumnado: la definición de algoritmos.

Dado que el objetivo de la experiencia era mejorar las habilidades del alumnado en la resolución de problemas e introducir los conceptos asociados al diseño, la aplicación del marco pedagógico se limitó a estos temas, que se abordan en las primeras clases de la asignatura.

En esta experiencia, se decidió utilizar los robots educativos Lego MINDSTORMS®. Para programarlos se optó por el entorno de desarrollo visual NXT-G, para que el alumnado comprendiera y explorara los distintos tipos de bloques y su semántica, y posteriormente abordararlos con una notación algorítmica estándar y un lenguaje de programación concreto. Estos robots se utilizaron durante tres cursos académicos con un total de 100 estudiantes [Álvarez et al. 13, Álvarez et al. 16].

Dado que el objetivo de la asignatura se centra en los diferentes aspectos relacionados con la programación y no en la construcción de robots, se utilizaron robots previamente montados. En una primera clase se presentaron al alumnado los diferentes componentes de los robots (sensores, motores, etc.). A continuación, se realizaron varios ciclos de Kolb para trabajar cada una de las perspectivas definidas en el marco pedagógico. Los siguientes párrafos detallan las actividades realizadas en cada ciclo.

3.1.1. Kolb en la perspectiva conceptual

Para aplicar Kolb en la perspectiva conceptual de la asignatura, se presentó al alumnado un ejercicio desarrollado en el entorno visual NXT-G para que experimentasen su comportamiento al trasladarlo al robot. Se les planteó también la realización de pequeñas modificaciones sobre los mismos para que experimentasen y observasen el comportamiento de su programa lo que les ayudó a reflexionar sobre la experiencia. Cuando los estudiantes terminaron con la realización de estos ejercicios, entrarían en la tercera etapa de abstracción.

En esta etapa, el profesorado ayudó a los estudiantes a conceptualizar los temas tratados en los diferentes ejercicios para continuar su aplicación en otros nuevos. Finalmente, se le planteó al alumnado que implementase un conjunto de ejercicios que trataban

los mismos temas que los trabajados en las etapas anteriores.

3.1.2. Kolb en la perspectiva notacional

En esta aplicación del ciclo de Kolb, se trabajaron los aspectos formales de notación utilizada para representar los algoritmos de los ejercicios realizados en el ciclo anterior. En este caso, para la primera etapa, el profesorado presentó un diagrama de flujo que describía el algoritmo correspondiente al ejercicio ejemplo presentado en la experiencia concreta del ciclo en la perspectiva conceptual. El alumnado observó y reflexionó sobre las relaciones entre el programa ejemplo y los elementos del algoritmo propuesto por el profesorado, y con su ayuda, conceptualizó los elementos de la notación trabajados. Finalmente, el alumnado realizó más ejercicios.

3.2. Experiencia 2

Tras los dos años de utilización de los robots educativos en *Programación Básica*, se decidió aplicar el marco pedagógico a todo el temario. Sin embargo, algunos autores han documentado experiencias negativas con el uso de robots educativos cuando se utilizan durante todo el curso. Esto se debe principalmente a que el alumnado tiene acceso limitado a ellos fuera del horario lectivo y esta limitación de acceso produce que su impacto en el aprendizaje se vea interrumpido [Fagin et al. 02].

Por esta razón se desestimó la utilización de los robots para esta segunda experiencia y se optó por el entorno visual de programación Scratch (<https://scratch.mit.edu/>). Aunque este entorno se ha utilizado principalmente para la enseñanza de la programación en centros educativos con estudiantes más jóvenes, también se ha utilizado en universidades [Maloney et al. 10]. Scratch se ha utilizado en la asignatura *Programación Básica* durante los tres últimos cursos académicos, ya que permite abordar desde la perspectiva conceptual los principales aspectos de la programación (secuencia de instrucciones, instrucciones condicionales, etc.), aislando al alumnado de los aspectos sintácticos que presentan los lenguajes de programación. En esta experiencia participaron 187 estudiantes. La

aplicación del marco pedagógico se realizó de la siguiente manera:

3.2.1. Kolb en la perspectiva conceptual

En esta experiencia, la perspectiva conceptual se abordó de la siguiente manera. En primer lugar, se presentó un ejemplo sencillo en Scratch relacionado con un tema de la asignatura (p.ej., sentencias condicionales). A continuación, el alumnado experimentó con los bloques de Scratch extendiendo y modificando el programa presentado, lo que permite al alumnado observar y reflexionar sobre los efectos de las modificaciones realizadas en el apartado anterior. Posteriormente, y con ayuda del profesorado, el alumnado conceptualizó los contenidos a partir de las observaciones realizadas. Finalmente, se plantearon ejercicios similares al anteriormente presentado para que el alumnado experimentase y practicara con los conceptos trabajados.

3.2.2. Kolb en la perspectiva notacional

Una vez asimilados los conceptos trabajados en la aplicación del ciclo en la perspectiva conceptual, se presentó la representación del ejemplo propuesto en el ciclo anterior tanto en notación algorítmica como su implementación en Java. El alumnado experimentó con el programa proporcionado para reflexionar y conceptualizar los aspectos asociados, identificando la relación de los aspectos conceptuales y los sintácticos. Finalmente, el alumnado debía diseñar e implementar un conjunto de ejercicios propuestos para la experimentación activa.

3.3. Experiencia 3

La tercera experiencia se implantó en la asignatura de *Programación Modular y Orientación a Objetos* (PMOO) [Anfurrutia et al. in press]. El objetivo principal de la experiencia era ayudarles a comprender los conceptos relacionados con la orientación a objetos.

Para esta asignatura se optó por utilizar BlueJ (<http://bluej.org>) y Greenfoot (<http://greenfoot.org>),

dos entornos visuales de desarrollo integrados y diseñados con fines educativos [Barnes et al. 12, Kölling 10]. Ambos entornos están orientados al aprendizaje de los conceptos de la Programación Orientada a Objetos y dan soporte para trabajar tanto la perspectiva conceptual como la notacional. Para ello, proporcionan herramientas visuales tales como el diagrama de clases UML para la representación de los objetos. Además, desde su perspectiva visual, permiten por un lado crear objetos e interactuar con ellos (mediante paso de mensajes) sin escribir una línea de código. Por otro lado, también permiten analizar el estado y comportamiento de los objetos.

Ambos entornos de programación visual se utilizaron de forma continuada en la asignatura de PMOO durante 5 cursos académicos con 340 estudiantes. Por sus características, de las 15 semanas que consta la asignatura, se utilizó *BlueJ* durante la primera mitad del cuatrimestre para introducir los conceptos de clase y objeto, mientras que en la segunda mitad se empleó *Greenfoot* para trabajar los mecanismos de herencia y polimorfismo. Además, la última herramienta también es utilizada en la realización de un proyecto por el alumnado fuera del horario lectivo.

Durante las 15 semanas del curso, las clases se dividen en clases magistrales y laboratorios prácticos. Al final de cada laboratorio, el alumnado debía contestar un cuestionario de tipo test relacionado con los temas trabajados en el laboratorio, lo que conlleva una mayor integración de las herramientas utilizadas en la evaluación de la asignatura. A continuación se detalla cómo se ha desarrollado esta experiencia.

3.3.1. Kolb en la perspectiva conceptual

El alumnado utilizó la herramienta seleccionada para experimentar visualmente con un proyecto ejemplo con el objetivo de trabajar uno de los temas de la asignatura. Por ejemplo, interactuaron con clases y objetos mediante sus métodos además de analizar tanto el cambio de estado de los objetos como la colaboración entre ellos. Por otro lado, el alumnado debía rellenar un formulario en el que reflejaba lo observado en la experiencia previa. Este formulario estaba compuesto de preguntas que guiaban al alumnado a realizar las observaciones adecuadas en esta fase. Teniendo en cuenta estas observaciones, y

con la ayuda del profesorado, conceptualizaron los temas trabajados en el ejercicio. Finalmente, el alumnado realizó nuevos ejercicios para poner en práctica lo aprendido.

3.3.2. Kolb en la perspectiva notacional

En esta asignatura, se aborda la implementación en Java de la interacción entre objetos experimentada en el entorno educativo en la perspectiva anterior. Para ello, el alumnado analiza y experimenta con la implementación del proyecto ejemplo. De nuevo, un formulario compuesto de preguntas orientó al alumnado a realizar observaciones que faciliten la posterior conceptualización de los contenidos. Finalmente, se plantearon ejercicios en los que el alumnado implementaba las acciones experimentadas en el ciclo anterior.

4. Resultados de los estudios sobre las experiencias

A continuación se resaltan los resultados más destacados de cada una de las experiencias realizadas [Álvarez et al. 16, Anfurrutia et al. in press].

4.1. Experiencia 1

Del análisis de los cuestionarios se deduce que tanto el interés del alumnado como su motivación se incrementaron gracias al uso de los robots. En cuanto a la conciencia del aprendizaje, fundamental en los procesos educativos, el alumnado percibió que los robots ayudaban a comprender mejor los conceptos abordados en la asignatura. Estas observaciones fueron refrendadas por el profesorado de la asignatura, que también detectaron un aumento en la motivación del alumnado, así como una mejora de la atmósfera de clase. Para el profesorado fue muy estimulante ver como algunos estudiantes grababan y difundían los videos de los robots ejecutando algunas de las tareas.

En el análisis de los resultados se han detectado diferencias en las opiniones considerando la perspectiva de género. A la hora de determinar si el uso de los robots educativos les ayudó a comprender

las instrucciones condicionales o las estructuras iterativas, las mujeres encuestadas respondieron de una manera levemente más negativa que los hombres. Sin embargo, las respuestas a la pregunta formulada para determinar si el uso de los robots les ha ayudado a ser conscientes del conocimiento adquirido no presentan esta diferencia (el 70% de los hombres y el 67% de las mujeres respondió afirmativamente a la pregunta). Así mismo, el 73% de las mujeres y el 75% de los hombres consideran que el uso de robots contribuyó positivamente a comprender mejor la utilidad de la asignatura.

A pesar de todos estos efectos positivos, la mejora en las calificaciones no fue estadísticamente significativa.

Por otro lado, en estas experiencias también se observaron algunos aspectos negativos. Por un lado, dado que el uso de los robots se realizó en las clases iniciales y que su uso no se evaluaba posteriormente hizo que, a pesar de ser conscientes de su efecto positivo, algunos estudiantes solicitaran la utilización de entornos de desarrollo de ámbito general como Eclipse desde el principio. Estos resultados concuerdan con los de [Orton-Johnson 09], que describen que el alumnado prefiere prescindir de aquellas actividades que no tienen impacto directo en la calificación. Por otro lado, también se observaron dificultades inherentes al uso de dispositivos físicos: factores relacionados al movimiento de los robots y aspectos relacionados con las condiciones lumínicas del aula. Las diferencias en la fricción de las superficies o la carga de las baterías afectaban al movimiento de los robots, mientras que las diferencias de las condiciones lumínicas, incluso dentro de la misma aula, afectaban al reconocimiento de los colores, esenciales para la realización de determinadas tareas.

4.2. Experiencia 2

Los profesores de la asignatura observaron con satisfacción que, a medida que se avanzaba en el curso, el alumnado exploraba y experimentaba con el entorno Scratch, probando alternativas sobre los ejercicios propuestos, e incluso tratando de realizar actividades adicionales. Además, en el apartado de evaluación se integraron preguntas asociadas a

Scratch (comprensión de los bloques, reconocimiento de errores o soluciones adecuadas), lo que conllevó que se integrara mejor con la asignatura y el alumnado fuera menos reticente a utilizarlo.

En los resultados de las encuestas, se observa que el alumnado opina que el uso de Scratch le ha ayudado a entender los conceptos relativos a los condicionales y a las sentencias iterativas. Del conjunto de estudiantes encuestados, entre los hombres, sólo un 24% opina que no le ha ayudado a entender las estructuras condicionales y un 31% que no le ha ayudado a entender las sentencias iterativas. En el caso de las mujeres, ninguna indica que no le haya ayudado con las estructuras condicionales y sólo un 10% indica que no le ha ayudado con las iterativas. Además, el 60% de los hombres está de acuerdo con haber utilizado Scratch. Este porcentaje alcanza el 78% en el caso de las mujeres. Es decir, en esta experiencia, la respuesta de las mujeres hacia el uso de Scratch es más positiva.

4.3. Experiencia 3

Los estudios realizados en la Experiencia 3 indican que el alumnado considera que las herramientas utilizadas son de ayuda para el aprendizaje (sólo un 14% del alumnado indica que no le ha ayudado). Es destacable que el porcentaje de estudiantes que considera que el uso de estas herramientas les ayuda es mayor entre los repetidores que entre los no repetidores. Además, el alumnado ha indicado en las encuestas realizadas que no dejarían de utilizarlos. Sin embargo, cabe destacar las diferencias observadas en función del género del alumnado en el resultado de las encuestas. Las respuestas de las mujeres son notablemente más negativas que las de los hombres en este aspecto.

Para determinar el efecto de la innovación realizada se han analizado también los resultados académicos del alumnado en la asignatura. Con la incorporación de estos nuevos entornos como apoyo al ciclo de aprendizaje de Kolb, el porcentaje de estudiantes presentados al examen final ha subido de porcentajes que rondaban un 40% a una tasa superior al 60%. Además, de este alumnado presentado, el porcentaje de aprobados ha pasado de un 45% a valores cercanos al 70%.

Sin embargo, los resultados en cuanto a la motivación, en general, no cumplen las expectativas del profesorado involucrado. De hecho, resultan peores que los alcanzados en experiencias previas con robots físicos en la asignatura de Programación Básica [Wu et al. 08, Álvarez et al. 16]. En este apartado se observó una elevada tasa de indecisos, especialmente en el caso de BlueJ, en el que alcanzó un 52%.

5. Lecciones aprendidas

Las experiencias llevadas a cabo han permitido identificar un conjunto de aspectos a considerar a la hora de implantar experiencias similares en asignaturas de programación.

5.1. Estudiantes: género y conocimiento previo

En las experiencias ha quedado claramente reflejado que existen diferencias en los resultados atendiendo a ciertas características del alumnado, su género y el conocimiento previo en programación que tienen.

Aunque los resultados obtenidos son, en general, positivos, se han detectado en algunos casos grandes diferencias en los resultados atendiendo al género del alumnado. Estas diferencias plantean la necesidad de ampliar el estudio antes de continuar con nuevas implantaciones.

Los estudios realizados se deben ampliar para analizar por ejemplo si el problema son los entornos de desarrollo seleccionados o las temáticas de los tipos de ejercicios realizados. Ampliar esta parte del estudio permitiría adaptar adecuadamente los resultados positivos que se han obtenido en las experiencias presentadas.

Un problema latente que debe ser tratado adecuadamente es la diferencia de conocimiento previo que tiene el alumnado. En las respuestas a las encuestas se han detectado diferencias notorias respecto a la motivación y a la aceptación de las herramientas utilizadas atendiendo al conocimiento previo del alumnado.

Una posibilidad para dar respuesta a esta problemática puede ser utilizar diferentes entornos de programación para las diferentes categorías de estudiantes. Incluso conforme los estudiantes van avanzando se les puede ir cambiando el entorno de programación, dado que no existe un entorno adecuado para cualquier situación [Hirst et al. 03]. Para el caso de los robots LEGO se podrían utilizar entornos visuales gráficos sencillos como Enchanting (<http://enchanting.robotclub.ab.ca>) o NXT-G con aquellos que no tienen conocimiento previo, mientras que con los estudiantes que sí tengan conocimientos previos empezar directamente con librerías especializadas como LeJOS (<http://www.lejos.org>).

5.2. Selección de herramientas de apoyo

Se han detectado ciertas dificultades inherentes a la utilización de dispositivos físicos. Sin embargo, no se deben descartar automáticamente ya que se ha observado una motivación mayor que con los entornos no físicos. Esto plantea la necesidad de que los ejercicios deben diseñarse teniendo estos factores contextuales en cuenta [Álvarez et al. 16].

Por otro lado, los entornos visuales permiten abstraerse fácilmente de los detalles reales y centrarse en la implementación de la lógica de la aplicación. Además, dado que no hay ningún entorno que sirva para todo, parece interesante combinar diferentes herramientas. En relación a la integración entre los dispositivos físicos y los entornos de desarrollo, generalmente no suele haber problemas; los primeros ofrecen librerías para reutilizar la funcionalidad, mientras que los segundos permiten ser extendidos. Sin embargo, cabe reseñar la falta de integración de la funcionalidad de despliegue desde el entorno de desarrollo del proyecto al dispositivo físico. Para ello, se requiere de compiladores del lenguaje utilizado por el programador al lenguaje ensamblado utilizado por el dispositivo. En este aspecto, librerías como LeJOS ofrecen dicha funcionalidad para Eclipse.

5.3. Integración en la asignatura

Para lograr una mayor aceptación de los entornos utilizados por parte del alumnado, éstos deben integrarse convenientemente en el proceso de

evaluación de la asignatura, o bien hay que relacionarlos con otras asignaturas de la rama de programación para dar una perspectiva global. Por ello, se plantea también que sería interesante utilizar la misma herramienta con metodología similar en *Programación Básica* y posteriormente en *Programación Modular y Orientación a Objetos*. Esto podría hacer que el alumnado tuviera una mejor visión de los mismos al estar más contextualizados en el grado.

5.4. Aspectos metodológicos

A pesar de no haber tenido una mejora en las calificaciones estadísticamente significativa, el efecto positivo del marco pedagógico establecido es palpable (aumento en la tasa de presentación, etc.). Para poder trabajar con un marco pedagógico como el presentado en este artículo y aplicar correctamente el ciclo de Kolb, es fundamental formular preguntas al final de los ejercicios para que el alumnado observe, reflexione y encuentre respuestas a los problemas planteados. Los entornos de desarrollo visual facilitan la realización de dichas tareas y dan un soporte adecuado a los aspectos metodológicos.

6. Integración de entornos: propuesta y experiencia piloto

Las lecciones aprendidas durante las experiencias presentadas en los apartados anteriores indican que sería de gran interés realizar en la asignatura una integración efectiva de entornos visuales y físicos que contemplasen tanto las diferencias de género como el conocimiento previo del alumnado.

En este apartado se presenta una nueva propuesta para dar respuesta a la problemática presentada anteriormente. Primero se presenta una propuesta de integración de estos tipos de entornos y, posteriormente, se describe cómo se pueden utilizar en un marco pedagógico como el presentado en este artículo. Finalmente se presenta la evaluación de una experiencia piloto realizada.

6.1. Propuesta de integración de entornos

En las lecciones aprendidas se han identificado aspectos positivos y negativos tanto en los robots educativos como en los entornos educativos visuales. Los robots físicos han generado una influencia positiva mayor en la motivación pero se observan problemas relacionados con el entorno físico en el que se utilizan y sus limitaciones de uso fuera del aula. Los entornos visuales no han presentado una influencia tan positiva en la motivación del alumnado pero adolecen de los otros problemas detectados en la experiencia con robots físicos. Teniendo estos aspectos en cuenta, a continuación se propone una solución integradora que combine las ventajas de los entornos visuales y los robots educativos para proporcionar un soporte tecnológico adecuado al marco pedagógico propuesto en este artículo.

El modelo de integración de entornos propuesto conlleva dos experiencias, una virtual y otra física (ver Figura 2). Para la experiencia virtual, se utilizan mundos virtuales basados en robots en el entorno visual. El objetivo principal es mantener la motivación del alumnado, y evitar los problemas surgidos por las limitaciones de uso fuera del horario lectivo, así como los derivados del uso de robots en entornos reales. La experiencia virtual consta de 3 fases:

1. **Desarrollo:** Se plantea que los estudiantes utilicen un entorno de desarrollo visual (p.ej, entornos de programación por bloques de instrucciones). Este entorno debería dar soporte a los dos ciclos (conceptual y notacional) identificados en el marco pedagógico.
2. **Simulación:** El entorno integrado muestra una simulación visual en pantalla del comportamiento de los robots ejecutando el programa desarrollado en la primera fase. La herramienta que se utilice debería permitir hacer pausas así como observar y analizar el estado del escenario y de los robots. Esta fase posibilita la observación y reflexión de lo ocurrido, dando soporte a las fases correspondientes del ciclo de aprendizaje de Kolb.
3. **Validación:** Se centra en la validación automática del programa desarrollado mediante pruebas unitarias. Si en la fase de validación automática se detectan errores, el

alumnado recibirá un informe que le permitirá no sólo subsanar los errores cometidos, sino que también le ayudará a interpretar lo eventos ocurrido y contrastar lo realizado con lo aprendido. Los informes generados en esta fase proporcionarán información de ayuda para la etapa de conceptualización del ciclo de Kolb.

En el caso de que la validación fuera satisfactoria y como premio al esfuerzo realizado, se realizará la experiencia física desplegando el código del programa en el robot y ejecutándolo en un entorno físico real.



Figura 2. Modelo de integración de entornos

Esta propuesta permite dar respuesta a los problemas identificados previamente. El uso de un entorno virtual que muestra la simulación alivia la necesidad de utilizar robots físicos, permitiendo que el alumnado pueda realizar tareas en este entorno fuera de las horas lectivas en escenarios que les resultan más motivantes. Además, la simulación y la validación permiten una identificación temprana de errores y soluciones incorrectas, permitiendo hacer un uso más eficiente del tiempo. La posibilidad de proporcionar retroalimentación como resultado de la validación permitirá realizar una evaluación formativa [Edwards 04] y facilita la labor del profesorado. Así mismo, la obtención de puntuación en la fase de validación permitirá una mayor integración del entorno en la evaluación de la asignatura. Por último, la posibilidad de probar los programas desarrollados en un robot físico puede ser un aspecto que incremente la motivación del alumnado.

6.2. Aspectos metodológicos

Dado que el marco pedagógico empleado en las experiencias educativas descritas anteriormente ha tenido efectos positivos, se plantea su aplicación con el soporte tecnológico descrito en el apartado anterior. De nuevo, se continuarán utilizando primero desde una perspectiva conceptual y a continuación desde una perspectiva notacional tal y como se ha realizado en las experiencias previas.

Para que la aplicación sea satisfactoria, es de vital importancia definir adecuadamente los tipos de ejercicios que se plantean al alumnado. En este aspecto, se plantea la generación de una lista de ejercicios adaptable a las preferencias de temática del alumnado lo que permitirá dar respuesta, entre otros aspectos, a las diferencias detectadas en base al género.

En este caso, la forma de aplicar el marco pedagógico sería el siguiente: En la fase de desarrollo, cada estudiante recibirá un prototipo de programa que se adecúe a sus preferencias y, siguiendo un guión, realizará pequeñas modificaciones en el programa. En la fase de simulación podrá observar los efectos de las modificaciones realizadas. Estas observaciones pueden ser complementadas con la retroalimentación proporcionada por la validación y ser de utilidad para la fase de conceptualización del ciclo de aprendizaje de Kolb. Posteriormente, el alumnado puede experimentar desarrollando nuevos programas, donde además podrá observar cómo se comportan tanto los robots virtuales como los físicos.

Por otro lado, dado que las herramientas que se plantean deben soportar tanto la edición basada en bloques como la edición utilizando un lenguaje de programación concreto, se podrían llevar a cabo la perspectiva conceptual y la notacional en el mismo entorno. Además, esto permitiría que aquellos estudiantes con conocimientos previos aborden, por ejemplo, sólo la perspectiva notacional.

6.3. Experiencia piloto realizada

La implantación de la propuesta integradora completa puede presentar dificultades considerables y conllevar un coste económico importante. Teniendo en cuenta el trabajo y los resultados presentados por [Wu et al.

08], antes de proceder a un proceso de tal envergadura, se decidió realizar una prueba piloto a menor escala. En este caso se optó por realizar una prueba inicial para evaluar el impacto y la aceptación de la innovación propuesta considerando solamente la experiencia virtual. La experiencia se realizó en la asignatura de *Programación Modular y Orientación a Objetos*, y participaron un total de 28 estudiantes.

En esta experiencia piloto, se utilizó un entorno educativo visual para trabajar con micro-mundos basados en robots y dar soporte a las 3 primeras fases de la propuesta integradora descrita en el apartado anterior. A continuación se detalla el proceso seguido en cada una de las fases (ver Figura 3.):

1 **Desarrollo:** El desarrollo de los programas propuestos se ha realizado mediante el entorno educativo Greenfoot. Éste presenta dos formas de trabajar: (1) la tradicional utilizada hasta ahora basada en escenarios gráficos y el editor de Java

y (2) un editor basado en bloques denominado Stride. Este último editor permite al alumnado experimentar y trabajar con aspectos de programación orientada a objetos minimizando los problemas sintácticos. Por otro lado, aquellos estudiantes con más conocimientos pueden utilizar el editor tradicional y trabajar con el lenguaje de programación Java. Esto permite que el alumnado pueda trabajar en diferente grado con ambos de acuerdo a sus conocimientos previos.

2 **Simulación:** La simulación de los programas implementados se ha desarrollado sobre los escenarios educativos de Karel Robot que permiten visualizar y verificar el funcionamiento en un entorno simulado. Para esta fase se analizaron distintas alternativas (p.ej., LeJOS y simuladores basados en ésta) pero se descartó su uso por incompatibilidades con la herramienta utilizada en la fase de validación.



Figura 3. Experiencia piloto realizada

- Validación:** En esta fase se definieron pruebas unitarias descritas mediante JUnit que fueron ejecutadas en un servidor con el software Web-CAT (<http://web-cat.org/>) empleado para validar automáticamente el código desarrollado por el alumnado y proporcionarles a todos una retroalimentación adecuada para mejorar tanto el código como su aprendizaje.

Para evaluar esta propuesta, se realizó una encuesta al alumnado. En las encuestas, los estudiantes determinaron claramente que los ejercicios en los que se utilizaban los robots como temática central, su motivación se incrementaba. Sólo un 11% de los estudiantes indicó que este no era el caso. De nuevo, existen diferencias atendiendo al género. En este caso, sólo el 9% de los hombres indicó que no le motivaba más el entorno de los robots que otros mientras que en el caso de las mujeres esta cantidad se incrementa hasta un 17%.

También se preguntó al alumnado si preferían seguir utilizando el entorno de los robots o preferían otro tipo de entorno. En este caso, el 64% del alumnado indicó que quería seguir utilizando los robots y un 7% prefería utilizar juegos. Además, un 10% indicó que le era indiferente el entorno utilizado. En este caso, la aceptación de las mujeres fue ligeramente superior que la de los hombres.

En cuanto a la fase de validación, sólo un 32% del alumnado indicó que la evaluación automática no le motiva más para terminar los ejercicios propuestos. Además, un 54% del alumnado está totalmente de acuerdo con que la validación automática le ha ayudado a corregir los errores y sólo un 25% indicó que este no era el caso.

7. Conclusiones y líneas futuras

En este artículo se ha presentado un conjunto de experiencias de innovación educativa, basadas en el ciclo de aprendizaje de Kolb y realizadas en asignaturas de introducción a la programación.

El análisis de los resultados de las experiencias realizadas sugiere que una adecuada combinación del uso de entornos visuales junto con robots físicos

podría mejorar la motivación del alumnado en *Programación Modular y Orientación a Objetos*. Sin embargo, el alto grado de heterogeneidad en los conocimientos previos del alumnado de primer curso afecta a la aplicación de mejoras como las descritas, sobre todo en *Programación Básica*. Por lo tanto, el uso de este tipo de herramientas requiere ajustar el tipo de entornos de aprendizaje de la programación empleados a los conocimientos previos del alumnado, así como gestionar adecuadamente el número de entornos a utilizar en una misma asignatura o cuatrimestre, sobre todo si son nuevos para el alumnado.

Por otro lado, una adecuada integración de nuevas herramientas en la asignatura requiere que estas formen parte del sistema de evaluación. De esta manera se puede conseguir una mayor aceptación y motivación por parte del alumnado. El hecho de que la evaluación sea continuada resulta también positivo en este sentido.

Una vez detectadas estas consideraciones, en este artículo se ha realizado una propuesta de innovación docente, que mantiene los aspectos positivos de los experimentos y a su vez intenta dar respuesta a los aspectos negativos detectados en las mismas.

Actualmente estamos ampliando la experiencia piloto descrita en el apartado anterior. En esta ampliación se contempla la combinación completa de los entornos visuales y físicos. El alumnado diseñará sus programas en un entorno visual, en el que podrá ver visualmente la ejecución y a continuación lo transferirán al dispositivo físico. También se plantea como línea futura de trabajo la implementación de un compilador para portar el código Java del escenario Karel Robot basado en Greenfoot al robot físico LEGO Mindstorms.

Agradecimientos

Este trabajo está subvencionado parcialmente por la Universidad del País Vasco UPV/EHU (GIU16/15, GIU16/20 y EHUA16/22) y el Vicerrectorado de Estudios de Grado e Innovación de la Universidad del País Vasco (UPV/EHU) quien a través del

SAE/HELAZ ha subvencionado el proyecto PIE 6819 en el bienio 2014-2016.

Referencias

- [Álvarez et al. 13] A. Álvarez and M. Larrañaga. "Using LEGO Mindstorms to Engage Students on Algorithm Design". In *Frontiers in Education (FIE)*, 1346–1351. 2013.
- [Álvarez et al. 16] A. Álvarez and M. Larrañaga. "Experiences Incorporating Lego Mindstorms Robots in the Basic Programming Syllabus: Lessons Learned". *Journal of Intelligent & Robotic Systems*, V 81 N 1, p.117–129. 2016.
- [Anfurrutia et al. 17] F.I. Anfurrutia, A. Álvarez, M. Larrañaga and J.-M. López-Gil. "Entornos de Programación Visual para Programación Orientada a Objetos: Aceptación y Efectos en la Motivación de los Estudiantes". *VAEP-Revista Iberoamericana de Tecnologías de Aprendizaje*, V 5 N 1, pp. 11-18, 2017.
- [Barnes et al. 12] D.J. Barnes and M. Kölling. "Objects first with Java: a practical introduction using BlueJ". Boston: Pearson. 2012.
- [Edwards 04] S.H. Edwards. "Using software testing to move students from trial-and-error to reflection-in-action". *ACM SIGCSE Bulletin*, V 36 N 1, p.26–30. 2004.
- [Fagin et al. 02] B.S. Fagin and L. Merkle. "Quantitative Analysis of the Effects of Robots on Introductory Computer Science Education". *Journal on Educational Resources in Computing (JERIC)*, V 2 N 4.
- [Georgantaki et al. 07] S. Georgantaki and S. Retalis. "Using educational tools for teaching object oriented design and programming". *Journal of Information Technology Impact*, V 7 N 2, p.111–130. 2007.
- [Gomes et al. 07] A. Gomes and A.J. Mendes. "Learning to program-difficulties and solutions". In *International Conference on Engineering Education-ICEE*, Coimbra, Portugal.
- [Hirst et al. 03] A.J. Hirst, J. Johnson, M. Petre, B.A. Price and M. Richards. "What is the best programming environment/language for teaching robotics using Lego Mindstorms?". *Artificial Life and Robotics*, V 7 N 3, p.124–131. 2003.
- [Kolb 84] D.A. Kolb. "Experiential learning: experience as the source of learning and development". Prentice-Hall. 1984.
- [Kölling 10] M. Kölling. "The Greenfoot Programming Environment". *ACM Transactions on Computing Education*, V 10 N 4, p.1–21. 2010.
- [Lahtinen et al. 05] E. Lahtinen, K. Ala-Mutka and H.-M. Järvinen. "A study of the difficulties of novice programmers". *ACM SIGCSE Bulletin*, V 37 N 3, p.14–18. 2005.
- [Leonard 02] D.C. Leonard. "Learning theories, A to Z". Westport, Conn.: Oryx Press. 2002.
- [Maloney et al. 10] J. Maloney, M. Resnick, N. Rusk, B. Silverman and E. Eastmond. "The Scratch Programming Language and Environment". *ACM Transactions on Computing Education*, V 10 N 4, p.1–15. 2010.
- [Orton-Johnson 09] K. Orton-Johnson. "I've stuck to the path I'm afraid': exploring student non-use of blended learning". *British Journal of Educational Technology*, V 40 N 5, p.837–847. 2009.
- [O'Sullivan et al. 04] D. O'Sullivan and T. Igoe. "Physical Computing: Sensing and Controlling the Physical World with Computers" 1st edition. Boston: Thomson. 2004.
- [Sartzemi et al. 08] M. Sartzemi, V. Dagdilelis and K. Kagani. "Teaching Introductory Programming Concepts with Lego MindStorms in Greek High Schools: A Two-Year Experience". In *Service Robot Applications*, InTech 2008.
- [Wang et al. 12] C. Wang, L. Dong, C. Li, W. Zhang and J. He. "The Reform of Programming Teaching Based on Constructivism". In W. Hu (ed) *Advances in Electric and Electronics. Lecture Notes in Electrical Engineering*, 425–431. Springer Berlin Heidelberg 2012.
- [Wilson et al. 10] A. Wilson and D.C. Moffat. "Evaluating Scratch to introduce younger schoolchildren to programming". *Proceedings of the 22nd Annual Psychology of Programming Interest Group (Universidad Carlos III de Madrid, Leganés, Spain)*.

- [Wu et al. 08] C.-C. Wu, I.-C. Tseng and S.-L. Huang. "Visualization of Program Behaviors: Physical Robots Versus Robot Simulators". In R. T. Mittermeir and M. M. Sysło (eds) *Informatics Education - Supporting Computational Thinking*. Lecture Notes in Computer Science, 53–62. Springer Berlin Heidelberg.
- [Yan 09] L. Yan. "Teaching Object-Oriented Programming with Games". In *Sixth International Conference on Information Technology: New Generations*, 2009. ITNG '09, 969–974. 2009.
- [Zhu et al. 03] H. Zhu and M. Zhou. "Methodology First and Language Second: A Way to Teach Object-oriented Programming". In *Companion of the 18th Annual ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages, and Applications*. OOPSLA '03, 140–147. New York, NY, USA: ACM.