

El software de animación como una estrategia innovadora para el aprendizaje de lenguajes de programación

Pedro A. WILLGING^{1,2}, Gustavo J. ASTUDILLO¹, Silvia BAST¹

¹ Departamento de Matemática, Facultad de Ciencias Exactas y Naturales, UNLPam

² CONICET

pedro@exactas.unlpam.edu.ar, astudillo@exactas.unlpam.edu.ar, bast@exactas.unlpam.edu.ar

Resumen

El dominio de un lenguaje de programación es un desafío tanto para los docentes que lo enseñan como para los estudiantes que lo quieren aprender. Desde que se han comenzado a utilizar las computadoras de manera masiva, los docentes de informática han buscado maneras innovadoras de facilitar la enseñanza y el aprendizaje de la programación. Este trabajo expone las experiencias desarrolladas en la Facultad de Exactas y Naturales (UNLPam), durante tres años, en un curso de apoyo a ingresantes. En el mismo se utilizó Scratch y la metodología propuesta por George Polya para la resolución de problemas, buscando introducir a los estudiantes en las nociones básicas de programación.

Contexto

Una de las temáticas abordadas en los primeros cursos de las carreras Profesorado en Computación, Profesorado en Matemática y Licenciatura en Matemática¹ es la programación estructurada, tópico éste, sobre el que los estudiantes no han recibido formación en la escuela media. Asimismo, los docentes de la asignatura Introducción a la Computación, han percibido que los estudiantes que ingresan presentan serias dificultades para:

- Interpretar los enunciados de los problemas que se les proponen.

¹ Facultad de Ciencias Exactas y Naturales (UNLPam)

- Adquirir los conceptos de programación relacionados con las sentencias de control de un lenguaje de programación.

Es por lo anterior que desde 2010 se viene desarrollando un trabajo conjunto entre la cátedra de *Introducción a la Computación* y el proyecto *Sistemas de Gestión de Aprendizajes de Código Abierto*. Este último trabaja, principalmente, sobre tres líneas de investigación: (a) red de cliente delgado con software libre, (b) objetos de aprendizaje y (c) compatibilidad de software de código abierto. En el marco de esta última se desarrolla la experiencia del presente trabajo. El proyecto cuenta con la colaboración de investigadores de la Universidad de San Luis y de la Universidad de Illinois. Se ha podido poner a punto una plataforma para educación virtual, y se ha experimentado con varias aplicaciones para producción de contenidos multimediales digitales; destinados a cursos en línea.

Palabras clave: software de animación, Scratch, lenguajes de programación, ingresantes

Introducción

Una de las líneas de investigación del proyecto aborda el análisis de diferentes aplicaciones de código abierto y sus posibilidades para su utilización en un aula (virtual o física). El software de animación Scratch es fruto de la búsqueda y evaluación de recursos disponibles en la Web y del análisis de trabajos que abordan esta temática. Se pudo observar que se trata de una aplicación de fácil instalación y

multiplataforma. Además, cuenta con un apropiado y amplio soporte a través de manuales y foros de ayuda (en español), y gran cantidad de materiales (ejemplos) disponibles en línea para ser descargados, utilizados y modificados libremente. A partir de las pruebas de campo realizadas con varios grupos de alumnos/as en situación de escolarización primaria y secundaria surge que se trata de una aplicación con un bajo nivel de dificultad en el aprendizaje de la interface y en el diseño de los programas.

Debido al potencial de la aplicación y dado que desde la Cátedra Introducción a la Computación se deseaba implementar acciones tendientes a mejorar el desempeño de los estudiantes que cursan por primera vez la asignatura, se resolvió coordinar actividades en conjunto con el equipo de trabajo del proyecto de investigación Sistemas de Gestión de Aprendizajes de Código Abierto. Es así que se decidió implementar un Curso de Resolución de Problemas (CRP), el cual tuvo (y tiene) como objetivo mejorar y facilitar el aprendizaje de los estudiantes, que se inician en la programación de computadoras.

Curso de Resolución de Problemas

Desde 2010 la Facultad de Exactas y Naturales (UNLPam) desarrolla un conjunto de actividades para sus ingresantes enmarcadas en lo que se denomina Jornadas de Ambientación Universitaria. En este contexto y previo al inicio de las cursadas 2010, 2011 y 2012, se implementó un “Curso de Resolución de Problemas” (CRP) con los siguientes objetivos:

- Identificar los datos e incógnitas de un problema dentro de un enunciado.
- Utilizar estrategias adecuadas para la resolución de un problema.
- Incorporar la noción de algoritmo.
- Identificar primitivas de entrada, salida, selección, y repetición y utilizarlas apropiadamente para el diseño del algoritmo que resuelva un problema

El curso es de carácter optativo y se dicta en la modalidad *b-learning*. Las actividades propuestas a los estudiantes, que se desarrollan durante los primeros meses del año (febrero-marzo), incluyen la resolución de problemas planteados en un foro sobre Moodle, la lectura del marco teórico (método de Polya) y la resolución de problemas basadas en soluciones algorítmicas. Para el diseño de los programas se utiliza el software de animación Scratch (Willging, Astudillo & Bast, 2010). En CRP se combinan la utilización de recursos tradicionales con los que ofrece la Web 2.0, encuentros presenciales y virtuales –sobre Moodle– y la utilización de una herramienta interactiva y visual que permite programar con el mouse sin preocuparse por la sintaxis.

Durante las dos primeras semanas se desarrollaron actividades focalizadas en que los estudiantes aprendieran a analizar enunciados y se familiarizaran con técnicas de resolución de problemas. Durante las semanas restantes, se trabajó en la aplicación de las mismas en la resolución de problemas de solución algorítmica con la aplicación Scratch.

Participaron del curso, en 2010, 2011 y 2012, estudiantes que cuentan con un alto grado de acceso a Internet –desde sus casas y/o desde los cybers– y utilizan entre 5 y 10 horas por día la computadora. En general, no han tenido experiencia en el uso de lenguajes de programación.

Software Scratch

En los cursos mencionados se diseñaron las actividades para ser resueltas utilizando Scratch como herramienta de software. Esta elección está basada en la filosofía del lenguaje que lleva a los estudiantes a incorporar las nociones básicas de programación de manera muy sencilla, para posteriormente, realizar una transición a lenguajes de programación tradicionales

Scratch es un lenguaje de programación con el cual se pueden crear historias interactivas, animaciones, juegos, música y arte; los cuales pueden compartirse alojándolos en el sitio

web <http://scratch.mit.edu>. Los usuarios de Scratch, (niños, adolescentes, jóvenes y adultos) van incorporando conceptos de programación mientras aplican distintas estrategias de resolución de problemas a través de la creación de proyectos. Scratch esta desarrollado por un grupo de investigadores del Lifelong Kindergarten en el Laboratorio de Medios de MIT y puede descargarse de forma gratuita.

Los creadores de Scratch se basaron en las experiencias del lenguaje Logo, que fuera presentado por Seymour Papert en su libro *Mindstorms* (1980), y otros proyectos relacionados a las animaciones y software de programación como Alice (Kelleher & Pausch, 2007).

La gramática de Scratch se basa en un conjunto de “bloques gráficos de programación” que se combinan para crear programas. Estos bloques tienen la apariencia de piezas de rompecabezas y encastran entre sí sólo si son sintácticamente correctos, por lo que liberan al estudiante de la complejidad sintáctica que implica el aprendizaje de un lenguaje de programación y les permite concentrarse sólo en seleccionar y combinar los bloques apropiados para la resolución del problema. Cada una de las sentencias de control, como así también las de entrada salida, está representada por uno o más bloques. La forma de los mismos indica al usuario cuáles son las que pueden anidar otras sentencias, dado que tienen forma de llave o C, tal es el caso de las condicionales y repetitivas. En cuanto a los operadores hay claras diferencias en la forma de los booleanos y los numéricos

La interfaz de Scratch (Figura 1) imita un escritorio físico donde los usuarios pueden ir dejando bloques que creen que serán necesarios para el proyecto actual o descartando aquellos que ya no usen. En este ambiente es aceptable ser un poco desordenado y propiciar la experimentación. A partir de la práctica los usuarios comienzan a crear de un modo más planificado y ordenado. Existen ya, varias experiencias de utilización de Scratch como recurso

pedagógico (Crook, 2009; Pepler & Kafai, 2007) y no es difícil anticipar que será utilizado en muchas más experiencias de docencia e investigación.



Figura 1. Ambiente de trabajo de Scratch

Líneas de investigación y desarrollo

Las investigaciones se centran en la identificación y evaluación de aplicaciones educativas y el impacto que éstas tienen en el diseño de materiales y/o en los procesos de enseñanza y aprendizaje.

Se han definido líneas de acción para testear y evaluar las posibilidades pedagógicas de varias aplicaciones educativas libres y/o de código abierto, entre las que se encuentra Scratch.

Se han implementado actividades de capacitación que permiten difundir, en nuestra comunidad, los resultados de esta línea de investigación. Estas acciones propician, además, la obtención de información y la recopilación de experiencias que retroalimentan la investigación.

Se continuará con la recolección de datos, a partir del contacto con docentes (de diferentes niveles educativos), que permitan analizar el impacto que tienen este tipo de recursos sobre los aprendizajes y motivación de los estudiantes y docentes.

Resultados obtenidos/esperados

Para la evaluación de la propuesta, se utilizaron las herramientas que ofrece la plataforma de aprendizaje Moodle: informe de participación de los estudiantes, las encuestas y los cuestionarios.

Se realizaron dos encuestas a los estudiantes. Una que permite evaluar su participación en el curso, su experiencia con Scratch, y con las estrategias de resolución de problemas y la otra, ofrece un perfil del estudiante respecto al acceso y uso de la tecnología. También se realiza una auto-evaluación a través de una encuesta al grupo de docentes/investigadores involucrados.

El reporte de actividades provisto por Moodle, el material publicado por los estudiantes en foros y tareas y el feedback que recibieron de los tutores, ofrecen datos valiosos que permiten efectuar el seguimiento de los estudiantes y obtener información relevante para la mejora iterativa del curso.

Del análisis de los datos de 2010 se pudo observar que del total de los estudiantes que participaron del CRP la mayoría se inscribieron para cursar Introducción a la Computación. De ellos la mitad participaron de todas las instancias de evaluación de la asignatura y el resto no completaron la cursada (abandonaron). Del grupo de estudiantes que realizaron el curso y rindieron los parciales, sólo la mitad aprobaron el primer parcial (1P).

Al analizar el trabajo de los estudiantes durante las actividades virtuales del CRP se observó que hubo una baja participación en los foros. Por su parte sólo la mitad de los participantes accedieron a los recursos y al material de lectura.

En la etapa presencial, si bien la asistencia fue alta, fue escasa la publicación de los ejercicios resueltos en la plataforma. Pero se pudo observar un marcado interés de los estudiantes en descargar y utilizar Scratch para programar.

Un dato alentador fue el resultado del cuestionario donde se indaga a los estudiantes sobre temas relacionados a programación estructurada (condicionales, ciclos, variables, entre otros). Más de la mitad de los participantes completó el cuestionario con resultados muy aceptables.

Luego del análisis de los resultados se hicieron algunos ajustes en el CRP para 2011.

En el mencionado año, la mayoría de los que participaron del CRP se inscribieron y cursaron Introducción a la Computación. De los que aprobaron el 1P casi el total habían realizado el curso.

Al analizar el trabajo de los estudiantes durante las semanas que se desarrolló el CRP, en forma virtual, se puede observar que hubo una aceptable participación tanto en los foros (proponiendo solución a diferentes problemas), como en las actividades de análisis de enunciados.

En la fase presencial, hubo también una alta participación en las clases y esto fue acompañado una importante intervención en las actividades propuestas en la plataforma.

Para el trabajo con Scratch, luego del encuentro presencial, si bien la mitad de los estudiantes accedieron a los trabajos propuestos sólo un tercio de ellos publicaron las soluciones y una proporción similar visitó la página de descarga del programa.

Respecto del cuestionario, los resultados fueron similares –aunque algo por debajo– a los obtenidos en 2010.

Con los datos de 2010 y 2011 se hicieron algunas modificaciones al CRP 2012 entre las cuales se pueden destacar el aumento en la cantidad de clases presenciales y una mayor difusión. Esto último está ligado al análisis de una de las encuestas donde los estudiantes manifiestan no haber participado del curso por no estar informados.

Finalmente, del análisis de las encuestas del cuerpo de docentes y tutores del CRP se pudo advertir que, el hecho de que Scratch trabaje con el encastrado de bloques para armar los programas, permite que los estudiantes, que dan sus primeros pasos en programación, puedan concentrarse en la resolución del problema. Además, la sintaxis va siendo incorporada por los estudiantes en tanto arman los algoritmos en Scratch, esto impacta positivamente en el correcto uso de las sentencias y la sintaxis en el lenguaje Pascal.

En este momento, se están analizando los datos correspondientes a los ingresantes

2012. Una primera evaluación, hace presumir que se mantienen (o incluso mejoran) los indicadores de los ciclos anteriores.

Formación de recursos humanos

En este proyecto, trabajan actualmente dos investigadores formados (director y codirectora) y seis investigadores en formación. Los investigadores pertenecen a tres universidades: dos universidades nacionales y una extranjera.

Tecnología en Educativa & Educación en Tecnología (TE&ET), 167-176.

Willging, P., Astudillo, G., & Bast, S. (2011). Impacto del software de animación en los aprendizajes de los estudiantes y en el diseño de materiales educativos. *Memorias del XIII Workshop de Investigadores en Ciencias de la Computación* (WICC 2011), 815-818.

Bibliografía

Crook, S. (2009). Embedding Scratch in the classroom.

<http://scratch.redware.com/scratch-in-the-classroom.html>

Kelleher, C. & Pausch, R. (2007). Using storytelling to motivate programming. *Communications of the ACM* 50(7), 58–64.

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books.

Peppler, K. & Kafai, Y. B. (2007). Collaboration, Computation, and Creativity: Media Arts Practices in Urban Youth Culture. In C. Hmelo-Silver & A. O'Donnell (Eds.), *Proceedings of the Conference on Computer Supported Collaborative Learning*, New Brunswick, NJ.

Resnick, M. (2007). All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten. *Proceedings of the SIGCHI Conference on Creativity and Cognition*, Washington, D.C.

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

Willging, P., Astudillo, G., & Bast, S. (2010). Aprender a programar (¿y a pensar?) jugando. *Memorias del congreso de*