Andorra La Vella, del 8 al 10 de julio 2015
ISBN: 978-99920-70-10-9

Uso de Scratch y Lego Mindstorms como Apoyo a la Docencia en Fundamentos de Programación

Roberto Muñoz[1,3], Thiago S. Barcelos[2], Rodolfo Villarroel[3], Marta Barría[1], Carlos Becerra[1], Rene Noel[1], Ismar Frango Silveira[4]
[1] Universidad de Valparaíso, Chile
[2] Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, Brasil
[3] Pontificia Universidad Católica de Valparaíso, Chile
[4] Universidade Cruzeriro du Sul, Brasil
roberto.munoz.s@uv.cl, tsbarcelos@ifsp.edu.br,
rodolfo.villarroel@ucv.cl, marta.barria@uv.cl, rene.noel@uv.cl,
carlos.becerra@uv.cl, ismar.silveira@cruzeirodosul.edu.br

Resumen

Los cursos de pregrado relacionados con fundamentos de programación de computadores tienen frecuentemente altas tasas de reprobación. Por otro lado, el uso de material didáctico tal como Scratch y Robots Lego, tienen un alto potencial para desarrollar una aplicación más tangible y concreta de las habilidades requeridas para la programación. En este artículo presentamos la experiencia del curso de estos elementos en el curso de Fundamentos de Programación, el cual está dirigido a estudiantes de Ingeniería en Informática de una universidad chilena. Los resultados muestran un incremento considerable en la tasa de aprobación de los estudiantes que cursan la asignatura por primera oportunidad.

Abstract

High attrition rates are frequently found in undergraduate courses related to programming fundamentals. On the other hand, didactic activities based on robot kits like Lego Robots or visual programming software tools such as Scratch present a high potential of helping students develop programming skills in a more tangible way. In this article we describe the application of such activities in a Programming Fundamentals course aimed at students of the Informatics Engineering career in a chilean university. The results indicate a considerable increase in the approval rates of students that were enrolled in the course for the first time.

Palabras clave

Taller de Programación de Juegos, Fundamentos de Programación, Scratch, Lego Mindstorms.

1. Motivación

En el último tiempo, la tasa de retención de estudiantes de las universidades ha comenzado a ser considerado un indicador para la medición de su calidad. De acuerdo a las estadísticas chilenas, la tasa de deserción estudiantil en las carreras de educación superior en este país es de un 53%, considerando 231 instituciones educativas [1]. En cuanto a las tasas de deserción de tres de los programas más demandados (leyes, medicina e ingeniería), se estima que un 22% de los estudiantes se demora el número mínimo de años en graduarse, mientras que un 23% toma un año más de lo que dura su carrera. Un 28% de los estudiantes de estas carreras tiene un retraso significativo en la culminación de su carrera, mientras que un 27% abandona definitivamente sus estudios [1].

La carrera de Ingeniería en Informática es ofrecida desde el año 2005 por la Escuela de Ingeniería Informática de la Universidad de Valparaíso. Esta carrera sufre la misma situación de deserción que la señalada a nivel nacional. De acuerdo a estadísticas de la propia escuela, la tasa más alta de abandono se produce en el cuarto semestre de la carrera, siendo las asignaturas con mayor reprobación Cálculo, Álgebra, Física y Fundamentos de Programación (FP), todas del primer año de la carrera [2].

El currículum estándar para cursos de pregrado de computación, de acuerdo a la ACM y la IEEE, define que el dominio de los fundamentos de programación de computadores es una habilidad esencial a ser desarrollada por los estudiantes [3]. Los estudiantes de Ingeniería Informática de la Universidad de Valparaíso reciben su formación inicial tres cursos de programación: Fundamentos de Programación, que

Andorra La Vella, del 8 al 10 de julio 2015
ISBN: 978-99920-70-10-9

aborda tipos de datos, estructuras de control, y la definición y uso de algoritmos para la resolución de problemas. Posteriormente, y previa aprobación de Fundamentos de Programación, los alumnos cursan Programación I (P1), en el que se estudia el paradigma imperativo de programación, usando el lenguaje C. A Programación II, donde se estudia el paradigma de orientación a objetos, llega un número considerablemente menor de estudiantes que los que inician Fundamentos de Programación (cerca del 30%), esto debido a la alta tasa de deserción.

Por otra parte, la introducción a los lenguajes de programación es un curso de suma importancia para cualquier carrera asociada a las Ciencias de la Computación. Sin embargo, en el primer año de estos programas, estudiantes y por lo tanto sus profesores se ven envueltos en graves problemas. La aplicación de conceptos básicos o el diseño de algoritmos que son relativamente simples para los docentes, parece ser algo difícil para el estudiante [4]. La mayor parte de esos problemas son originados por la complejidad de los conceptos tales como variables, estructuras repetitivas, arreglos, funciones de los lenguajes de programación [5].

Estas dificultades se manifiestan independientes del paradigma y/o lenguaje utilizado. Esto se puede deber a diversos factores, tales como motivación, sintaxis, estilos de aprendizajes diferentes, experiencia previa, entre otros [6], [7], [8]. Se han desarrollado diversas herramientas que implementan estrategias que permiten mitigar estas dificultades. La utilización de entornos interactivos del estilo *drag and drop* como *Scratch* [9] permite eliminar errores de sintaxis e introducir nuevos conceptos de manera atractiva. Por otra parte, la programación con *Robots Lego Mindstorms* permite desarrollar una aplicación más tangible y concreta de las habilidades de programación. A continuación se reporta el desarrollo de un taller que conjuga ambas estrategias de enseñanza, estructurado en 2 secciones, la primera contiene una asociada a la programación de Juegos bajo el ambiente de *Scratch* y la segunda asociada a la programación con Robots *Lego Mindstorms NXT*, basándose en los principios del aprendizaje activo y constructivismo para su elaboración.

2. Trabajos relacionados

Durante el último tiempo se han logrado visualizar diferentes iniciativas asociadas al uso de programación con *Scratch* o con Lego en primeros años de carreras asociadas a la informática. Por ejemplo en [10], se utiliza *Scratch* en un curso de Ciencias de la Computación introductorio con el fin de mejorar la retención de estudiantes de alto riesgo, resultando en un aumento sustancial en la matrícula con respecto a años anteriores. El objetivo del trabajo presentado en [11] es resumir la experiencia en la enseñanza en un curso que puede ser considerado como introductorio a las Ciencias de la Computación transversal (denominados CS0 / IT0). Con los fundamentos pedagógicos en materia de aprendizaje construccionista y educación informática contextualizada, los autores presentan la motivación y los detalles de un curso que utiliza el lenguaje de programación *Scratch*, *App Inventor*, y la robótica con *Lego Mindstorms*.

González y Jiménez [12] presentan un experimento con kits de robótica de bajo costo para estimular en los estudiantes la exploración de relaciones entre la matemática, la física, la electrónica y la programación de computadores. Monsalves [13] señala que un ambiente de aprendizaje adecuado para actividades didácticas con robots puede ayudar al aprendizaje de contenidos teóricos. El estudio de la robótica puede conducir a conexiones multidisciplinarias y actividades prácticas, tal como ha sido aplicado en carreras de ingeniería para apoyar la enseñanza de programación [14], sistemas de tiempo real [15], electrónica [16], entre otras materias.

3. Diseño de las actividades

3.1. Sección de programación de juegos con Scratch

Las actividades del taller fueron distribuidas a lo largo de 12 sesiones, en la que cada encuentro duró una hora y media. En cada encuentro el profesor propuso la construcción de uno o más mecanismos de interacción relacionados con la construcción de un juego digital. Para esto, los estudiantes fueron invitados a explorar conceptos relacionados con desarrollo de juegos (animación de *sprites*, colisión, controles por teclado y mouse) y de Fundamentos de Programación (variables, mensajes, estructuras condicionales, estructuras repetitivas). A partir de los tópicos a ser cubiertos en la asignatura de FP, se definieron cuatro directrices:

1. **La construcción de juegos debe motivar el desarrollo de todas las actividades del taller.** Esta directriz se basa en el principio de fluidez de juegos puesto por Peppler y Kafai [17], que argumentan que el proceso de diseño y construcción de artefactos en el contexto cultural de los juegos digitales puede promover un proceso de reflexión y aprendizaje para los estudiantes.
2. **Las actividades deben progresivamente llevar a la construcción de la mecánica de un juego completo.** En trabajos anteriores que discuten el desarrollo y conducción de actividades prácticas con el objetivo de fomentar el desarrollo del Pen-

Andorra La Vella, del 8 al 10 de julio 2015
ISBN: 978-99920-70-10-9

samiento Computacional. Lee et al. [18] proponen el esquema Usar – Modificar – Crear, en el cual este principio se basa. Inicialmente a los estudiantes se les presentan programas listos para interactuar y comprender su funcionamiento, a partir de la comprensión obtenida en esa primera etapa, los estudiantes son invitados a introducir modificaciones en su funcionamiento y su apariencia. Finalmente, a medida que los estudiantes adquieren mayor confianza, comienzan a crear sus propios juegos, aplicando los conocimientos adquiridos en etapas anteriores. Cabe destacar que esta estrategia no es aplicada de forma lineal. Un estudiante puede actuar como creador de una etapa del proceso de aprendizaje y en una etapa siguiente, volver a actuar como "usuario" para comprender un nuevo concepto. Sin embargo, la estructura de las actividades y los conocimientos y habilidades a ser movilizados inducen al estudiante a actuar cada vez como "creador" a lo largo del taller. La estrategia todavía tiene impacto en la mantención de un nivel adecuado de desafío.

3. **Las actividades deben progresivamente demandar que nuevos conceptos sean explorados por los estudiantes, al mismo tiempo, solicitar que el estudiante utilice nuevamente conceptos explorados anteriormente.** La introducción de nuevos conceptos de forma paulatina en las actividades (a través del esquema Usar - Modificar - Crear) tiene como objetivo introducir continuamente nuevos desafíos que induzcan al estudiante a buscar nuevos conocimientos. El proponer nuevos desafíos al mismo tiempo en que conceptos ya explorados son requeridos nuevamente, tiene como objetivo que los estudiantes estén en un estado de flujo en lo que se refiere a su trabajo autónomo en la construcción de juegos [19]. De la misma forma, se espera que el profesor pueda actuar como facilitador en los momentos en que los estudiantes presenten dificultades puntuales relacionadas, por ejemplo, a un concepto específico. En estos casos, se espera que la secuencia planeada de las actividades mantenga a los estudiantes en la zona de desarrollo proximal [20], permitiendo así su avance en las actividades.
4. **La mecánica de los juegos, a pesar de ser simples, debe traer referencia al universo de los juegos "reales" para que sean significativas para los estudiantes.** Alineados con una propuesta constructivista, en que la construcción de artefactos digitales por los estudiantes demanda postular de una forma razonablemente autónoma, las actividades del taller siguen el abordaje de aprendizaje basado en problemas (ABP) [21]. Según Merril [21], el aprendizaje basado en problemas es una estrategia centrada en el estudiante, que trabaja de manera colaborativa en la solución de algún problema, en la cual la figura del profesor sirve como apoyo y la construcción del conocimiento es gradual y empírica. De esta forma, en cada actividad del taller, los estudiantes reciben instrucciones sobre los objetivos propuestos para el juego; además de eso, es presentado un ejemplo del juego propuesto siendo ejecutado y a partir de ahí inician el trabajo. El profesor actúa como un facilitador observando el trabajo e interviniendo a medida que los estudiantes solicitan mayores apoyos.

En las primeras semanas del taller, las actividades son relacionadas con la utilización del entorno *Scratch* y para reforzar los conceptos vistos en la cátedra de FP. Los juegos propuestos son: Piedra Papel y Tijera, Simulación de Guerra y prototipos de los famosos juegos *Breakout* y *Pacman*. Para finalizar los estudiantes debían presentar su versión del juego *Aero Fighters*.

En el cuadro 1 se presenta la programación de actividades del taller de programación de juegos con *Scratch.*

Sesión	Actividades / contenido
1	Familiarización con el ambiente *Scratch* (conceptos *sprite* y colisión entre *sprites*)
2	Variables y estructuras repetitivas
3	Estructuras repetitivas y estructuras condicionales
4	Crear juego Piedra-Papel-Tijera
5-6	Crear el juego Simulación de Guerra
7-8	Crear el juego Breakout
9	Pacman – Crear la mecánica básica de los movimientos de los personajes
10-11	Pacman – Implementar las demás características del juego final
12	Presentación del proyecto Final (Aero Fighters)

Cuadro 1: Actividades taller con *Scratch*

3.2. Sección de programación de robots con Lego Mindstorms

Para la sección de Programación de Robots con *Lego Mindstorms*, se consideró que la metodología de Aprendizaje Basado en Problemas (ABP) [22] sería adecuada para este objetivo. En ABP, el proceso de aprendizaje se basa en la resolución de un problema no trivial propuesto inicialmente por el profesor. Con el fin de resolver el problema, los estudiantes deben organizarse en grupos pequeños y buscar autónomamente información que puede ayudar a resolver el problema [22]. El ABP también fue escogido debido

Andorra La Vella, del 8 al 10 de julio 2015
ISBN: 978-99920-70-10-9

a su potencial para ayudar a los estudiantes a desarrollar habilidades de trabajo en equipo y el auto-aprendizaje que son de gran valor en un contexto profesional.

Las actividades del taller se dividieron en cinco unidades, cada una compuesta por un conjunto de tareas. Los estudiantes debían completar todas las tareas de una unidad con el fin de avanzar a la siguiente unidad. Cada tarea fue diseñada para ser desarrollada en aproximadamente 90 minutos. Además, las tareas de cada unidad fueron diseñadas para involucrar un concepto fundamental de programación necesario para los estudiantes en las disciplinas de FP. En el cuadro 2 se presenta el tema principal de cada unidad y los experimentos incluidos.

Unidad 1 – Introducción al taller de Robótica	1: Roles en los equipos, introducción a los kit de Lego Mindstorms NXT y su lenguaje de programación (NXC).
	2: *Lego Digital Designer* para la construcción de los robots Lego Mindstorms NXT.
Unidad 2 - Servomotores	3: Tipos de datos, constantes, estructuras de control (selectivas y repetitivas), funciones básicas de los servomotores.
	4: Programación de movimientos del robot, funciones avanzadas de los servomotores.
	5: Ejercicios
Unidad 3 – Sensores de luz	6: Funciones básicas y aplicaciones de los sensores de luz. Algoritmos básicos para trabajar con el robot seguidor de luz.
	7: Algoritmos básicos para trabajar con el robot seguidor de luz.
	8: Algoritmos básicos para trabajar con el robot seguidor de luz.
	9: Ejercicios
Unidad 4 – Sensores de ultrasonido	10: Funciones básicas y aplicaciones del sensor de ultrasonido. Algoritmos para el robot explorador (detecta y esquiva obstáculos).
	11: Programación del robot explorador
	12: Programación del robot explorador
	13: Ejercicios y Evaluación
Unidad 5 – Sensores táctiles y de sonido	14: Instrucciones básicas y sus aplicaciones en sensores táctiles y de sonido
	15: Desarrollo del proyecto final
	16: Desarrollo del proyecto final
	17: Ejercicios y evaluación

Cuadro 2: Unidades del taller y sus actividades asociadas.

Los materiales didácticos para el taller de programación de robots fueron diseñados teniendo en cuenta los diferentes estilos de aprendizaje [8]. El material generado estaba compuesto por los siguientes elementos:

- **Tutorial de la unidad**, con los contenidos necesarios para que los estudiantes lleven a cabo las tareas de la unidad. Consta de una explicación de las funciones del lenguaje de programación y el hardware del robot que será utilizado en el contexto de la unidad.
- **Guía de actividad**, que contiene un conjunto de ejercicios que los estudiantes deben resolver durante la unidad. Los primeros ejercicios incluyen la solución y los siguientes, que son de mayor dificultad, deben ser resueltos por los alumnos con el fin de mejorar sus habilidades de programación. La guía de actividades incluye videos e ilustraciones que representan las funcionalidades que deben ser implementadas.
- **Programas de ejemplo**, que se componen de códigos en NXC previamente probados, que se ponen a disposición de los estudiantes para analizar, modificar y probar los robots *Lego Mindstorms NXT* antes de construir su propio código. Además de ser el tipo de material más apreciado por los estudiantes según nuestra encuesta preliminar, también es parte de una estrategia didáctica (el ciclo Usar - Modificar - Crear) recomendado por [23] para mejorar la habilidad de los estudiantes en la construcción de artefactos computacionales.
- Una **presentación visual** basada en diapositivas, utilizada por el instructor del taller para presentar los objetivos de cada unidad a los estudiantes.
- **Videos** que presentan una explicación para una actividad o un programa de ejemplo.

Andorra La Vella, del 8 al 10 de julio 2015
ISBN: 978-99920-70-10-9

4. Resultados preliminares

El taller se ofreció a los estudiantes durante el primer semestre del año 2014. A todos los estudiantes de primer año matriculados en Fundamentos de programación se les permitió participar en el taller, sin embargo, su participación no fue obligatoria. El taller fue considerado como una actividad extra en el programa del curso. Sin embargo, fue necesaria la elaboración de un proyecto de término de sección *Aero Fighters* (Fig. 1, Fig. 2, Fig. 3) en el caso de *Scratch* y un robot de rescate que debe pasar por un laberinto y encontrar un objeto negro puesto al azar dentro de él en el caso de Lego (Fig. 4). Esto debido a que el programa de curso contempla la elaboración de 2 proyectos en un lenguaje de programación. Por lo que ambos ponderaron un 10% del promedio final cada uno.

Figura 1: Interfaz Juego Aero Fighters

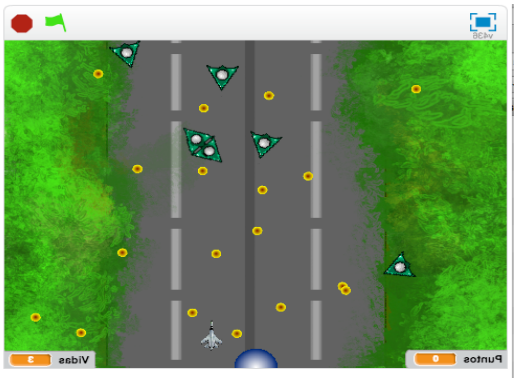

Figura 2: Interfaz Juego Aero Fighters Alumno

Figura 3: Instrucciones Aero Fighters Alumno

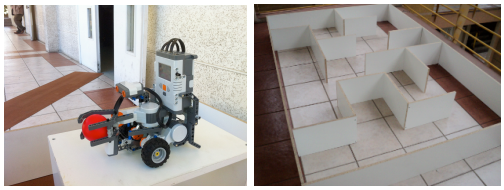

Figura 4: Proyecto 2 – Laberinto con Lego Mindstorms NXT.

Se comparó el desempeño de los estudiantes de Fundamentos de programación con los resultados de 2 años anteriores, cabe mencionar que esta comparación tiene como objetivo determinar si existe alguna mejoría en el desempeño de los estudiantes. Se presenta en el cuadro 3, el número de estudiantes que inscribieron la cátedra (n), las calificaciones obtenidas en ambos certámenes (C1, C2) y el porcentaje de aprobación del curso.

Año	n	C1	C2	Porcentaje aprobación
2012	75	2,6	2,5	34%
2013	76	1,8	3,2	45%
2014	76	3,7	2,9	50%

Cuadro 3: Histórico de la cátedra de Fundamentos de Programación en Universidad de Valparaíso.

Los resultados en una primera instancia son alentadores, ya que permitió subir al menos 1 punto en una escala de 1, a 7,0. En el C2 no se visualizaron grandes mejoras, esto debido a que el taller no cubre todas las unidades vistas en la cátedra de Fundamentos de Programación. Sin embargo la tasa de aprobación histórica la cuál bordea entre un 30% y 45% también se vio incrementada.

Como medio de evaluación de las actividades con *Scratch* se les presentó, de manera *online* y voluntaria, a los estudiantes 6 afirmaciones, las cuales debían responder si estaban Altamente en desacuerdo (*AD*), Desacuerdo (*D*), Neutro o Indiferente (*N*), De acuerdo (*A*) o Altamente de Acuerdo (*AA*). Estas afirmaciones fueron respondidas por 20 estudiantes. Posterior a ello se dio espacio para una sección de comentarios. La distribución de las respuestas a las preguntas realizadas se presentan desde el cuadro 4 al cuadro 9.

Creo que la programación con Scratch es fácil de entender				
AD	D	N	A	AA
5%	0%	0%	40%	55%

Cuadro 4: Afirmación 1

ISBN: 978-99920-70-10-9

Desarrollar los proyectos en Scratch en el curso de Fundamentos de Programación me apoyó a aprender programación				
AD	D	N	A	AA
5%	0%	20%	45%	30%

Cuadro 5: Afirmación 2

Creo que personalizar mi juego en Scratch me apoyó a desarrollar distintas técnicas de programación				
AD	D	N	A	AA
5%	0%	10%	50%	35%

Cuadro 6: Afirmación 3

Creo que me ayudó en mi aprendizaje el compartir y/o discutir con mis compañeros estrategias de resolución en el proyecto realizado con Scratch				
AD	D	N	A	AA
5%	5%	25%	45%	20%

Cuadro 7: Afirmación 4

Me gustó el programar bajo el esquema drag and drop (arrastrar y soltar) que se utiliza en scratch				
AD	D	N	A	AA
5%	0%	25%	35%	35%

Cuadro 8: Afirmación 5

Yo recomendaría seguir utilizando Scratch en INC102 Fundamentos de Programación				
AD	D	N	A	AA
5%	10%	10%	25%	50%

Cuadro 9: Afirmación 6

En los resultados se pueden visualizar resultados no concluyentes pero si alentadores, esto debido que en todas las afirmaciones sobre el 65% de los encuestados estuvo de acuerdo o altamente de acuerdo.

Por otra parte, rescatando comentarios de los estudiantes se puede destacar el realizado por el/la estudiante A1: *En lo personal Scratch fue de gran ayuda al aprendizaje de lenguaje de programación, que me permitía poder comprender de que se trataba ya que los conocimientos de programación eran escasos. Scratch de manera didáctica me apoyaba, porque si uno lo hacía correctamente obtenía los resultados deseados. Crear el juego compartir con los compañeros y discutir el tema ayudo más a la comprensión de este.* O el generado por A2: *Scratch es altamente útil y fácil de enseñar y aprender, además que es una gran herramienta para comenzar a pensar con lógica de programador para los estudiantes nuevos ingresados a la carrera*

5. Conclusiones y Trabajo Futuro

Adquirir los conceptos fundamentales de la programación de computadores es una habilidad crítica para los estudiantes de cursos relacionados con la informática. Sin embargo, las asignaturas relacionadas con este tema suelen tener altas tasas de deserción. Estrategias didácticas orientadas a la aplicación de los conceptos de programación abstracta como la construcción de juegos, pueden ser útiles en este contexto. Partiendo de esta premisa, los autores desarrollaron un taller de programación de juegos digitales con *Scratch* y Robots *Lego Mindstorms NXT* para apoyar y complementar la asignatura inicial de Fundamentos de Programación. Si bien la estructura del taller debe refinarse, los resultados son bastante alentadores. Como trabajo futuro se espera incorporar la programación de dispositivos móviles debido al alto interés y penetración que éstos tienen en los adolescentes. También se espera poder replicar este taller en otras carreras de la facultad que incorporen Fundamentos de Programación en sus mallas curriculares.

Agradecimientos

Roberto Muñoz es beneficiario de la Beca de Doctorado INF-PUCV 2015.

Referencias

[1] L. González, D. Jorquera, and S. González, “Estudio sobre la repitencia y deserción en la educación superior chilena,” *IESALC informes*, Apr-2005.

[2] Escuela de Ingeniería Civil en Informática de la Universidad de Valparaíso, “Inducción Alumnos,” Mar-2012. [Online]. Available: http://informatica.uv.cl/documentos/induccionICI2012.html.

[3] ACM-IEEE Software Engineering 2008, “Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering.” IEEEComputer Society and Association for Computing Machinery, 2008.

[4] A. J. Mendes and M. J. Marcelino, “Tools to support initial programming learning,” presented at the International Conference on Computer Systems and Technologies- CompSysTech’06 Tools, 2006.

[5] R. Muñoz, M. Barría, R. Noël, E. Providel, and P. Quiroz, “Determinando las Dificultades en el

ISBN: 978-99920-70-10-9

Aprendizaje de la Primera Asignatura de Programación en Estudiantes de Ingeniería Civil Informática," in *Memorias del XVII Congreso Internacional de Informática Educativa, TISE*, Santiago, Chile, 2012, pp. 120–126.

[6] A. Carbone, J. Hurst, I. Mitchell, and D. Gunstone, "An exploration of internal factors influencing student learning of programming," in *Proceedings of the Eleventh Australasian Conference on Computing Education - Volume 95*, Darlinghurst, Australia, Australia, 2009, pp. 25–34.

[7] R. M. Felder and L. K. Silverman, "Learning and Teaching Styles in Engineering Education," *Eng. Educ.*, vol. 78, no. 7, pp. 674–681, 1988.

[8] R. M. Felder and R. Brent, "Understanding Student Differences," *J. Eng. Educ.*, vol. 94, no. 1, pp. 57–72, Jan. 2005.

[9] D. J. Malan and H. H. Leitner, "Scratch for budding computer scientists," *SIGCSE Bull*, vol. 39, no. 1, pp. 223–227, Mar. 2007.

[10] M. Rizvi, T. Humphries, D. Major, M. Jones, and H. Lauzun, "A CS0 course using Scratch," *J Comput Sci Coll*, vol. 26, no. 3, pp. 19–27, Jan. 2011.

[11] S. Uludag, M. Karakus, and S. W. Turner, "Implementing IT0/CS0 with Scratch, App inventor for Android, and Lego Mindstorms," 2011, p. 183.

[12] J. González and J. Jiménez, "La robótica como herramienta para la educación en ciencias e ingeniería," *Rev. Iberoam. Informática Educ.*, no. 10, pp. 31–36, 2010.

[13] S. Monsalves, "Estudio sobre la utilidad de la robótica educativa desde la perspectiva del docente," *Rev. Pedagog.*, vol. 32, no. 90, 2011.

[14] H. Urrutia López, H. Bustos Andreu, M. Villalobos Abarca, and E. Jaramillo C, "Aprendizaje de la programación mediante el uso de robot Lego por alumnos de computación de la Universidad de Tarapacá," in *Actas del XXIV Congresso Chileno de Educación en Ingeniería*, Valdivia, 2010.

[15] J. Malagelada, R. Toledo Morales, E. Valderrama Vallés, J. Sorribes Gomis, and J. Pujol Capdevila, "Un ABP basado en la robótica para las ingenierías informáticas," in *Actas de las XV Jornadas de Enseñanza Universitaria de la Informática*, Barcelona, 2009.

[16] J. S. Artal and J. M. Artacho, "La Robótica como herramienta PBL en la enseñanza de la Electrónica en Ingeniería," in *Actas I Congreso Internacional sobre Aprendizaje, Innovación y Competitividad*, Madrid, 2011.

[17] K. Peppler and Y. Kafai, "Gaming Fluencies: Pathways into Participatory Culture in a Community Design Studio," *Int. J. Learn. Media*, vol. 1, no. 4, pp. 45–58, Nov. 2009.

[18] I. Lee, F. Martin, J. Denner, B. Coulter, W. Allan, J. Erickson, J. Malyn-Smith, and L. Werner, "Computational thinking for youth in practice," *ACM Inroads*, vol. 2, no. 1, pp. 32–37, Feb. 2011.

[19] J. Nakamura and M. Csikszentmihalyi, "Flow theory and research," in *Oxford Handbook of Positive Psychology*, 2nd ed., Oxford: Oxford University Press, 2009, pp. 195–206.

[20] L. S. Vygotsky, "Zone of Proximal Development," in *Mind in society: The development of higher psychological processes*, Oxford: Harvard University Press, 1978, pp. 52–91.

[21] D. Merril, "A Pebble-in-the-Pond Model For Instructional Design," *Perform. Improv.*, vol. 41, pp. 41–46, 2002.

[22] H. S. Barrows, "A taxonomy of problem-based learning methods," *Med. Educ.*, vol. 20, no. 6, pp. 481–486, 1986.

[23] I. Lee, F. Martin, J. Denner, B. Coulter, W. Allan, J. Erickson, J. Malyn-Smith, and L. Werner, "Computational thinking for youth in practice," *ACM Inroads*, vol. 2, no. 1, pp. 32–37, Feb. 2011.