

RECOMENDACIÓN PERSONALIZADA DE EJERCICIOS EN SCRATCH: UNA ESTRATEGIA PARA EL APRENDIZAJE DE LA PROGRAMACIÓN EN LA UNIVERSIDAD

Jesennia Cárdenas-Cobos¹, Pavel Novoa-Hernández^{1,2}, Yaima Trujillo-Reyes², Amilkar Puris-Cáceres^{1,2}

¹Universidad Estatal de Milagro, Milagro-Guayas-Ecuador

²Universidad Técnica Estatal de Quevedo, Quevedo-Los Ríos-Ecuador

*Corresponding Author: pnovoa@uteq.edu.ec

Resumen: Este trabajo tiene por objetivo mejorar el aprendizaje de la programación en estudiantes universitarios a través de Scratch, un lenguaje de programación visual que facilita el aprendizaje informal. A pesar del progreso logrado en cursos previos con Scratch, se ha observado algunos problemas con respecto a la usabilidad de Scratch por parte de los estudiantes. En primer lugar, existe una brecha entre los enfoques de aprendizaje empleados, ya que los profesores están limitados a las actividades del aula. En segundo lugar, ciertos estudiantes se sienten desmotivados porque se enfrentan a ejercicios de programación que no satisfacen sus expectativas individuales. De manera que, para resolver este problema, se propone un enfoque integrado que consiste en una simple aplicación web que incluye Scratch como editor de proyectos junto con un sistema de recomendación para ejercicios.

Palabras Clave: Sistemas de recomendación, Scratch, Aprendizaje de la programación, Aprendizaje mejorado por tecnologías.

PERSONALIZED RECOMMENDATION OF EXERCISES IN SCRATCH: A STRATEGY FOR LEARNING TO PROGRAM IN THE COLLEGE

Abstract: This work aims to improve the learning of programming in university students through Scratch, a visual programming language that facilitates informal learning. Despite the progress made in previous courses with Scratch, some problems have been observed regarding the usability of Scratch by students. First, there is a gap between the learning approaches employed since teachers are limited to classroom activities. Second, some students feel unmotivated because they face programming exercises that do not meet their particular expectations. So, to solve this problem, we propose an integrated approach consisting of a simple web application that includes Scratch as a project editor and a recommendation system for exercises.

Key words: Recommender systems, Scratch, Programming learning, Technology Enhanced Learning.

I INTRODUCCIÓN

En los enfoques de educación tradicional los profesores desempeñan un papel importante ya que guían el proceso de aprendizaje, esto es, de acuerdo a una estructura determinada. En otras palabras, el conocimiento se revela al estudiante de una manera organizada. Con frecuencia, a esta forma de aprender se le denomina aprendizaje formal [1]. Independientemente de los beneficios de dicho modelo educativo, en los últimos 20 años el denominado Aprendizaje mejorado por la tecnología (TEL por sus siglas en inglés) ha planteado modelos alternativos. TEL se define como un dominio de aplicación que cubre tecnologías para apoyar todas las formas de enseñanza y aprendizaje [2].

Un ejemplo importante de TEL en el contexto de la educación de programación es Scratch [3], un lenguaje de programación visual desarrollado y mantenido por el Instituto de Tecnología de Massachusetts (MIT). Gracias a su estilo intuitivo, Scratch reduce significativamente la curva de aprendizaje en niños y principiantes en la programación de computadoras. El enfoque educativo clave promovido por Scratch es el aprendizaje informal. Básicamente, esto se debe a que los estudiantes son libres de crear, explorar y reutilizar proyectos de programación sin la presencia del docente. De esta manera, Scratch también se puede utilizar para apoyar la enseñanza formal o tradicional de la programación en entornos educativos.

En este contexto, Scratch se ha incluido como un recurso educativo para enseñar la materia de Fundamentos de la Programación de Computadoras en la Universidad Estatal de Milagro (UNEMI) (Ecuador). Desde que comenzó en 2015, se ha apreciado una mayor motivación y rendimiento académico entre los estudiantes. Sin embargo, también se notaron algunos problemas relacionados con la usabilidad de Scratch: (i) existe una brecha entre los enfoques de aprendizaje empleados ya que los profesores se centran solo en actividades de la clase, y (ii) ciertos estudiantes se sienten desmotivados porque se enfrentan a ejercicios de programación que no cumplen con sus expectativas individuales (p. ej., los ejercicios son demasiado fáciles o demasiado complejos).

Por lo tanto, surge la pregunta: ¿cómo mejorar el aprendizaje con Scratch, mientras se salva la brecha entre los enfoques de aprendizaje formal e informal, y se permite a los estudiantes hacer ejercicios de acuerdo con sus niveles de conocimiento y experiencias previas? Una solución intuitiva para este problema proviene de los Sistemas de Recomendación (RS) [4], [5]. La literatura refleja varios ejemplos de beneficios significativos

de RS en entornos educativos [6], [7].

Teniendo en cuenta las motivaciones anteriores, en este trabajo se propone un enfoque integrado para mejorar el proceso de aprendizaje con Scratch para estudiantes universitarios. Este enfoque consiste en una aplicación web simple que incluye Scratch como editor de proyectos junto con un RS para ejercicios (es decir, definiciones de problemas). Estos ejercicios son creados por el profesor e incluidos en el sistema. Por lo tanto, el alumno puede acceder a los ejercicios, resolverlos con Scratch y evaluarlos según dos criterios: gusto (taste) y complejidad (complexity). Con base en el registro de los ejercicios evaluados, el sistema recomienda nuevos ejercicios para el alumno a través de un enfoque de filtrado colaborativo. La suposición es que los estudiantes con evaluaciones similares sobre un conjunto dado de ejercicios son una buena fuente de recomendación. Desde una perspectiva pedagógica, esta propuesta puede considerarse un mediador entre los enfoques de aprendizaje formal e informal. En otras palabras, no solo llena el vacío entre las conferencias de los profesores y la interacción del alumno con Scratch, sino que también ayuda a desarrollar el aprendizaje autónomo.

II PROPUESTA

El sistema propuesto consta de dos elementos generales: una aplicación Web que le permite al estudiante interactuar con Scratch y los ejercicios diseñados por el docente; y en segundo lugar, un sistema de recomendación de ejercicios. En lo que sigue se describen ambos.

2.1 Aplicación Web

La aplicación web tiene una interfaz gráfica de usuario (GUI) fácil de usar, y se desarrolló utilizando las tecnologías libres PHP y MySQL.



Fig. 1. Intefaz gráfica del sistema propuesto.

En la Fig. 1, puede apreciarse que en la zona central de la ventana, Scratch aparece incrustado como un editor, mientras que en las zonas superior e inferior aparece el resto de los módulos agregados. En la zona superior, el área de usuario proporciona funcionalidades en términos de acceso del usuario al sistema. La interfaz de usuario para el ejercicio RS aparece en el área derecha de la pantalla. Este módulo, de arriba hacia abajo, consiste en:

- 1.un reloj para medir el tiempo que el estudiante pasó resolviendo el ejercicio,
- 2.la definición de ejercicio,
- 3.un grupo de ejercicios recomendados,
- 4.un conjunto de todos los ejercicios disponibles
- 5.un sistema para evaluar el gusto y la complejidad del ejercicio activo.

2.2 Aspectos del sistema de recomendación

El RS incluido en la aplicación web, está basado en un enfoque de filtrado colaborativo que emplea tanto usuarios como ítems [8], [9]. Evidentemente, en este contexto, los usuarios son los estudiantes, mientras que los elementos son los ejercicios a resolver en Scratch. El objetivo de este RS es explotar la experiencia de los estudiantes ya registrados en el sistema, para sugerir convenientemente ejercicios a un estudiante en particular.

III RESULTADOS Y DISCUSIÓN

La evaluación de RS en entornos educativos es aún un tema de investigación en desarrollo. De hecho, es un proceso complejo porque involucra varios objetivos que son difíciles de evaluar (por ejemplo, el conocimiento adquirido por el alumno, la satisfacción del usuario, entre otros). Estos objetivos de evaluación se pueden agrupar en tres amplias categorías: rendimiento del RS, efectos centrados en el usuario, y efectos del aprendizaje [10].

Debido a las características de nuestro entorno educativo y al tiempo relativamente corto de uso del sistema propuesto, se adoptó la metodología de Prueba en la vida real (Real-Life Testing) [10]. Su objetivo principal es evaluar el sistema de acuerdo con la satisfacción percibida por los usuarios reales, esto es, los estudiantes.

En este contexto, un total de 64 estudiantes de Ingeniería en Sistemas e Ingeniería Industrial participaron en la evaluación del sistema propuesto. Ambas carreras incluyen el curso Fundamentos de Programación en sus planes de estudios. Sin embargo, es de esperar más motivación en los estudiantes de Ingeniería en Sistemas que en los estudiantes de Ingeniería Industrial. Tal diferencia puede ser útil para evaluar el sistema desde dos puntos de vista distintos.

Después de usar el sistema durante un período de tres meses, se le pidió a los estudiantes que respondie-

sen un cuestionario con nueve afirmaciones.

La Tabla I muestra estas afirmaciones de acuerdo con los tres objetivos de evaluación mencionados anteriormente [10]. Se hizo más énfasis en la evaluación de los efectos de aprendizaje y en los efectos centrados en el usuario (por ejemplo, se incluyen más preguntas relacionadas con estos objetivos). Sin embargo, se han incluido dos afirmaciones relacionadas con la precisión y el tiempo de respuesta del sistema (las afirmaciones A1 y A2).

Está claro que ambas afirmaciones son menos precisas que los resultados que se pudieran obtener a partir de un experimento o simulación, pero ambas brindan información relevante sobre cómo los usuarios perciben al sistema.

Las nueve afirmaciones fueron evaluadas a través de las siguientes categorías: totalmente de acuerdo, de acuerdo, ni de acuerdo ni en desacuerdo, en desacuerdo, y totalmente en desacuerdo.

Tabla I. Objetivos y afirmaciones empleadas para evaluar cualitativamente al sistema propuesto.

Objetivo de evaluación	Código	Afirmación
Rendimiento del SR	A1	El sistema me recomiendan ejercicios de acuerdo a mis habilidades de programación.
	A2	La interacción con el sistema es lo suficientemente rápida.
Efectos centrados en el usuario	A3	Los indicadores <i>gusto</i> y <i>complejidad</i> son criterios simples y efectivos para evaluar los ejercicios.
	A4	El sistema presenta una interfaz gráfica de usuario cómoda y la navegación es fácil.
	A5	Considero que este nuevo sistema (incluyendo recomendación) es superior a Scratch solo (sin recomendaciones).
Efectos del aprendizaje	A6	Tendré más posibilidades de aprobar la asignatura si utilizo el sistema.
	A7	Pienso que el sistema ayudará a mejorar mi rendimiento académico en la asignatura Fundamentos de la Programación de Computadoras.
	A8	El sistema ayuda a personalizar mi aprendizaje en programación.
	A9	Pienso que al usar el sistema he mejorado mi aprendizaje autónomo.

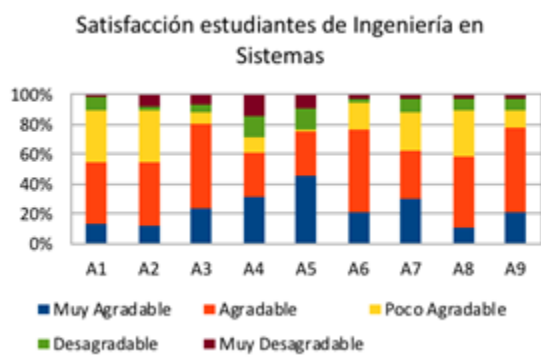


Fig 2. Resultados de las encuestas en estudiantes de Ingeniería en Sistemas.



Fig 3. Resultados de las encuestas en estudiantes de Ingeniería Industrial.

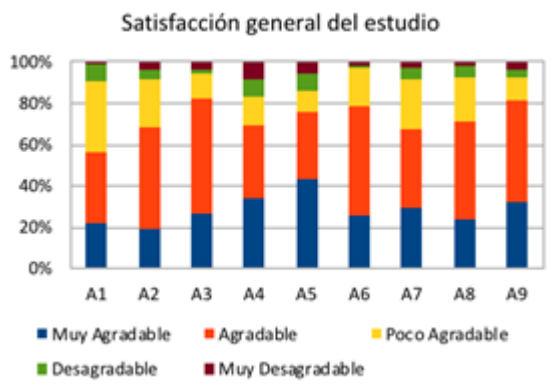


Fig 4. Resultados generales del estudio.

Los resultados de las encuestas realizadas se resumen en las Fig. 2, Fig. 3 y Fig. 4. Nótese que estas gráficas corresponden a tres grupos: Satisfacción de los estudiantes de Ingeniería en Sistemas, Satisfacción de los estudiantes de Ingeniería Industrial y Satisfacción en general respectivamente. La Fig. 3 es la agregación de los primeros dos grupos. En general, puede apreciarse un grado de satisfacción aceptable. Por ejemplo, más del 50% de los estudiantes estuvieron de acuerdo con todas las afirmaciones.

Sin embargo, existen diferencias entre los primeros dos grupos. Como se esperaba, los estudiantes de Ingeniería en Sistemas fueron más críticos que los estudiantes de Ingeniería Industrial (véase Fig. 2 y Fig. 3). En ambos casos, un número significativo de estudiantes no está de acuerdo con A1, lo que indica que se debe trabajar mucho más con respecto a la precisión del sistema. Una conclusión similar puede derivarse de la respuesta temporal del sistema (A2).

En cuanto a las afirmaciones centradas en el usuario (A3, A4 y A5), más del 60% de los estudiantes de ambas carreras están de acuerdo, y el 40% está totalmente de acuerdo con la afirmación A5, esto es, que el sistema propuesto es mejor que Scratch solamente.

Finalmente, con respecto a las afirmaciones para evaluar los efectos de aprendizaje (A6, A7, A8 y A9), existe una clara diferencia entre ambas áreas de estudio. Por ejemplo, alrededor del 60% de los estudiantes de Ingeniería en Sistemas están de acuerdo con esos indicadores, mientras que en el caso de Ingeniería Industrial esta cifra alcanza el 70%.

En general, hubo un nivel de satisfacción adecuado por parte de los estudiantes con respecto al sistema propuesto (Fig. 4).

IV CONCLUSIONES

En este trabajo, se propone un enfoque integrado

para mejorar el proceso de aprendizaje de los estudiantes de programación de computadoras con Scratch. La experiencia previa con respecto al uso de Scratch para complementar el proceso de enseñanza de la Fundación de Programación de Computadoras en la Universidad Estatal de Milagro, Ecuador, proporcionó una evidencia importante de que dicha estrategia de aprendizaje puede mejorarse. Específicamente, para resolver dos problemas: (1) la brecha actual entre los enfoques formales e informales de aprendizaje y (2) una interacción personalizada entre el estudiante y Scratch.

Por lo tanto, se desarrolló una aplicación web fácil de usar que incluye Scratch en conjunto con un sistema de recomendación de ejercicios. Se adoptó una metodología de Real-Life Testing para validar este enfoque. En particular, los estudiantes evaluaron el sistema propuesto empleando nueve aspectos relacionados con tres objetivos: (1) desempeño del sistema de recomendación; (2) efectos centrados en el usuario; y (3) efectos en el aprendizaje. En general, se observó un nivel significativo de satisfacción entre los estudiantes.

Sin embargo, estos resultados se consideran como un primer paso para tener un mejor sistema. Los trabajos futuros estarán orientados a mejorar tanto el rendimiento del sistema de recomendación como la usabilidad de la aplicación web. Además, se explorará la inclusión de la propuesta en el código fuente de Scratch con el objetivo de compartir sus beneficios con la comunidad.

V AGRADECIMIENTOS

P. Novoa-Hernández y A. Puris-Cáceres cuentan con el apoyo del Proyecto de investigación FOCICYT (2017-2018): Aplicaciones de la Soft Computing en escenarios de decisión de la Educación Superior, de la Universidad Técnica Estatal de Quevedo (Ecuador).

VI REFERENCIAS

- [1] European Centre for the Development of Vocational Training, Terminology of European education and training policy: a selection of 130 key terms, 2nd ed. Luxembourg: Publications Office of the European Union, 2014.
- [2] N. Manouselis, H. Drachler, R. Vuorikari, H. Hummel, and R. Koper, "Recommender Systems in Technology Enhanced Learning," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and B. P. Kantor, Eds. Boston, MA: Springer US, 2011, pp. 387–415.
- [3] D. J. Malan and H. H. Leitner, "Scratch for Budding Computer Scientists," in *38th SIGCSE Technical Sym-*

posium on Computer Science Education, 2007, vol. 39, no. 1, pp. 223–227.

[4]J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating Collaborative Filtering Recommender Systems,” *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, 2004.

[5]F. Ricci, L. Rokach, and B. Shapira, “Recommender Systems Handbook,” in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and B. P. Kantor, Eds. Boston, MA: Springer US, 2011, pp. 1–35.

[6]N. Manouselis, H. Drachsler, K. Verbert, and E. Duval, “Survey and Analysis of TEL Recommender Systems,” in *Recommender Systems for Learning*, New York, NY: Springer New York, 2013, pp. 37–61.

[7]A. Klačnja-Milićević, M. Ivanović, and A. Nanopou-

los, “Recommender systems in e-learning environments: a survey of the state-of-the-art and possible extensions,” *Artif. Intell. Rev.*, vol. 44, no. 4, pp. 571–604, 2015.

[8]J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, “Recommender systems survey,” *Knowledge-Based Syst.*, vol. 46, pp. 109–132, 2013.

[9]J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, “Recommender system application developments: A survey,” *Decis. Support Syst.*, vol. 74, pp. 12–32, 2015.

[10]M. . Erdt, A. . Fernández, and C. . Rensing, “Evaluating Recommender Systems for Technology Enhanced Learning: A Quantitative Survey,” *IEEE Trans. Learn. Technol.*, vol. 8, no. 4, pp. 326–344, 2015.