

EVALUACIÓN DE CALIDAD DE HERRAMIENTAS UTILIZADAS EN LA ENSEÑANZA DE LA PROGRAMACIÓN BASADA EN ISO 25000

María C. Espíndola, Cristina L. Greiner, Gladys N. Dapozo,

Departamento de Informática. Facultad de Ciencias Exactas y Naturales y Agrimensura

Universidad Nacional del Nordeste. 9 de Julio N° 1449. Corrientes. Argentina.

{gndapozo, cgreiner}@exa.unne.edu.ar, mcespindola@yahoo.com

RESUMEN

En la actualidad la enseñanza de la programación ha adquirido una especial importancia. El Consejo Federal de Educación declaró de importancia estratégica para el sistema educativo argentino la enseñanza y el aprendizaje de la programación durante la escolaridad obligatoria, para fortalecer el desarrollo económico-social de la Nación. La demanda insatisfecha de la industria del software promueve la formación de perfiles profesionales para el desarrollo de software. Por otra parte, se sostiene que la programación fomenta habilidades vinculadas con el pensamiento computacional, habilidades requeridas en el siglo XXI para entender el mundo y la tecnología que nos rodea. En este proyecto se plantea el desarrollo de un marco de referencia para evaluar las características de las herramientas visuales o lúdicas utilizadas en la enseñanza inicial de la programación, y permita orientar a los docentes hacia la selección de las herramientas más adecuadas para incorporar en sus propuestas docentes, en función de los objetivos específicos de aprendizaje, nivel educativo de los niños o jóvenes y expectativas del docente.

Palabras clave: Enseñanza de la programación inicial. Herramientas visuales o lúdicas. Evaluación de calidad de software.

CONTEXTO

Esta propuesta es parte del plan de Trabajo Final de la Maestría en Tecnologías de la Información (UNNE), que se desarrolla en el marco del proyecto 16F016 “Promoción del Pensamiento Computacional para favorecer la formación en STEM”, acreditado por la Secretaría de Ciencia y Técnica de la Universidad Nacional del Nordeste (UNNE).

1. INTRODUCCIÓN

El crecimiento de la industria del software en nuestro país no es una novedad. “El sector industrial de software es uno de los que más ha crecido en Argentina durante la última década, e incluso actualmente Argentina ya exporta más software que carne” [1] El último reporte de la Cámara Empresarial de Software y Servicios Informáticos (CESSI), en función de datos del 2016 y expectativas del 2017, señala que en 2016 la industria del software volvió a generar empleo, dejando una demanda insatisfecha de más de 5.000 puestos sin cubrir [2].

Necesidad de Recursos Humanos

Por otra parte, un estudio reciente señala que es evidente la producción limitada de graduados en STEM respecto de las necesidades del campo productivo y, la falta de masa crítica en capacidades para trabajar en la innovación digital [3].

Consciente de esta situación, diversas políticas públicas en nuestro país promueven la industria del software. Sin

embargo, éstas no solo apuntan al desarrollo del software, sino que van más allá, porque en la actualidad el mundo que nos rodea depende cada vez más del uso de la tecnología, que alcanza a las actividades profesionales, de educación, la comunicación en el trabajo y con los vínculos afectivos, el tiempo de esparcimiento, entre otras, atravesadas por el uso de dispositivos tecnológicos, de los cuales pocos saben cómo funcionan, y del trabajo que realizan las personas que se desarrollan en ese campo. Es así que aprender a programar se ha transformado en los últimos años en un componente clave de las “competencias del Siglo XXI” [4]. Permite desarrollar habilidades de pensamiento que sirven para cualquier tipo de actividad, como la capacidad de abstracción y de planificación, la descomposición de problemas, el trabajo en equipo, entre otras.

Enseñanza de la programación

En apoyo a esta iniciativa, el Consejo Federal de Educación ha declarado de importancia estratégica para el sistema educativo argentino la enseñanza de la programación durante la escolaridad obligatoria, para fortalecer el desarrollo económico-social de la Nación.

Por lo tanto, la enseñanza de la programación es una problemática que atraviesa todos los niveles [5], [6], [7]. Sin embargo, no es una tarea trivial. La investigación existente plantea muchas discusiones e ideas sobre la mejor manera de enseñar la programación introductoria, y las dificultades que enfrentan los estudiantes. Entre estas, el hecho de que los estudiantes consideran a la programación como una actividad puramente técnica en lugar de un conjunto de habilidades combinadas de resolución de problemas. Tienden a desarrollar un conocimiento superficial y no crean estrategias de resolución de problemas a través de la programación [8]. Por este motivo se implementan diferentes estrategias, entre las que

sobresale la de enseñar con herramientas lúdicas [9], [10].

En este contexto, cobra relevancia la elección de la herramienta a utilizar. En esta línea de investigación se propone la elaboración de un marco de referencia que brinde una guía a la hora de planificar un curso de enseñanza de programación introductoria.

Herramientas lúdicas

Los NPEs (Novice Programming Environments - Entornos de programación para principiantes) han tenido amplio reconocimiento en los últimos años, ya que la investigación ha demostrado que desempeñan un papel importante en la motivación de los nuevos programadores. En lugar de codificar instrucciones, un NPE ofrece características visuales que ocultan la complejidad de la sintaxis del lenguaje de programación, facilitando a los principiantes el uso de las estructuras algorítmicas básicas [11].

En [12] los autores presentan un modelo para desarrollar el pensamiento computacional basado en la programación visual. Este concepto se basa en la idea What You See Is What You Get (WYSIWYG), que evita a los programadores la escritura de código, ya que pueden usar bloques visuales para construir funcionalidades.

Mediante una revisión de la literatura, se identificaron cinco factores de éxito para enseñar programación con un enfoque lúdico: la motivación del estudiante, la integración y la participación en clase, el enfoque centrado en el alumno, la interacción y retroalimentación, y la fluida integración del contenido educativo en el juego [13].

Sin embargo, en [14] los autores señalan que, si la herramienta utilizada y los objetivos de aprendizaje no están alineados, podría no lograrse un gran efecto en el aprendizaje deseado. Señalan también la falta de certeza en la elección de la mejor herramienta para el

logro de los objetivos de aprendizaje, basada en los atributos de la misma.

Marcos de referencia (MR)

La utilización de un MR [15]:

- Evita tener que crear indicadores, ya que están definidos en el modelo.
- Permite disponer de un marco conceptual completo.
- Proporciona objetivos y estándares iguales para todos, en muchos casos ampliamente contrastados.
- Posibilita medir con los mismos criterios a lo largo del tiempo, por lo que es fácil detectar si se está avanzado en la dirección adecuada.

Algunos marcos teóricos tienden a darle mucho énfasis a los aspectos técnicos del software educativo. Sin embargo, un diseño técnico sólido no garantiza el valor educativo. No solo se debe medir el software educativo como producto, sino también los procesos de enseñanza-aprendizaje que facilita o potencia [16].

En este trabajo se propone construir un MR que permita realizar una evaluación integral de herramientas lúdicas. Se considera evaluar la calidad del producto mediante un estándar internacional. Estos permiten la definición de criterios de evaluación, la especificación de requerimientos, la descripción de componentes en relación a ellos y la identificación de desajustes de manera sistemática facilitando el proceso de evaluación y selección del software [17].

Estándar ISO/IEC 25010

Permite identificar las características de calidad relevantes para establecer criterios de satisfacción y las medidas correspondientes. Está compuesto por ocho características y subcaracterísticas de calidad. Para este trabajo se propone el análisis y selección de las características y subcaracterísticas que resulten relevantes para una evaluación sistemática y práctica de las herramientas lúdicas para la enseñanza de la programación.

Existen numerosos trabajos sobre evaluación de calidad de dispositivos con fines educativos [16], [18], [19].

Todos los autores coinciden en que a la hora de evaluar estos artefactos es necesario considerar dos aspectos:

- calidad desde el punto de vista técnico
- calidad en cuanto al impacto en el aprendizaje.

En [20] se realiza un análisis de varios modelos en los que surgen dos categorías importantes en la evaluación de software educativo: la dimensión pedagógica y la técnica. La parte pedagógica contempla las características de los destinatarios, los objetivos de enseñanza, los contenidos y el aprendizaje que promueve la aplicación. Desde el punto de vista técnico se resaltan la usabilidad y las características de la interface.

Criterios para determinar el impacto en el aprendizaje

En [21] se analizan trabajos relacionados al uso de los JS, y los autores concluyen que su uso incrementa significativamente el aprendizaje en el área de la Informática. Señalan también que en los artículos que se revisaron, el aspecto que más se evaluó además del aprendizaje fue la motivación. Esto se debe a que la motivación es considerada como un factor importante que describe lo que hace un buen juego independientemente de sus cualidades educativas [22]. En una experiencia basada en el desarrollo de actividades lúdicas presentada en [9] se concluyó que esta metodología permitió despertar el interés de los alumnos y consolidar conceptos básicos de programación.

2. LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

En esta línea, se proponen las siguientes actividades:

- Profundizar el conocimiento sobre marcos metodológicos que sustentan la evaluación de software, desde el

punto de vista de la ingeniería del software y desde los procesos de enseñanza-aprendizaje que potencia.

- Relevar y evaluar herramientas utilizadas en la enseñanza inicial de la programación, y realizar una clasificación según sus características
- Definir criterios de evaluación de las herramientas, en función de los distintos aspectos que surgirán de la etapa de relevamiento y evaluación.
- Diseñar un marco de referencia que oficie de guía en la selección de la herramienta más apropiada para el logro de los objetivos de aprendizaje.
- Validar el marco de referencia en una situación concreta, a definir de acuerdo a las posibilidades de acceso a los distintos ámbitos educativos.

3. RESULTADOS ESPERADOS

Se espera obtener un marco de referencia que oriente la selección de las herramientas visuales o lúdicas orientadas a la enseñanza inicial de la programación.

Este aporte facilitará a los docentes de los niveles educativos no universitarios, contar con información que les orientará hacia la selección de las herramientas más adecuadas para incorporar en sus propuestas docentes, en función de los objetivos específicos de aprendizaje, nivel educativo de los niños o jóvenes y expectativas del docente.

Este recurso contribuirá a facilitar el logro de los objetivos de las políticas públicas que fomentan la incorporación de la programación en la escuela, como una competencia necesaria para la formación del siglo XXI.

4. FORMACIÓN DE RECURSOS HUMANOS

En esta línea participan dos investigadoras formadas y una tesista de posgrado que realiza su trabajo final para obtener el título de Magister en Tecnologías de la Información, oferta interinstitucional entre la UNNE y

UNAM, dictado 2016-2017.

5. REFERENCIAS

- [1] SW 48 - SG Software Guru #48 Published on Sep 22, 2015, Revista SG. Software Guru. Mexico, Mexico. ISSN: 1870-0888
- [2] Reporte anual sobre el Sector de Software y Servicios Informáticos de la República Argentina. (OPSSI – CESSI). Mayo 2017.
- [3] Katz, R. (2016). TIC, digitalización y políticas públicas. En Entornos Digitales y Políticas Educativas. IPE-UNESCO.
- [4] Borchardt, M.; Roggi, I. (2017) Ciencias de la Computación en los Sistemas Educativos de América Latina. SITEAL. <http://www.siteal.iipe-oei.org>
- [5] Dapozo, G.; Greiner, C.; Pedrozzo Petrazzini, G.; Chiapello, J. Despertando vocaciones TIC mediante juegos y animaciones. Revista: Extensionismo, Innovación y Transferencia Tecnológica. Claves para el Desarrollo. ISSN 2422-6424 Vol. 1 Año 2014. Pag. 80-89.
- [6] Dapozo, G.; Greiner, C.; Petris, R. “Enseñar programación desde los juegos a lo formal”. 1er Congreso de Educación y Tecnologías del Mercosur: de la digitalización a la virtualización. ISBN 978-987-3619-26-7. 2017.
- [7] Dapozo, G.; Petris, R.; Greiner, C. “Programación en las escuelas. Experiencia de formación docente en el Nordeste Argentino”. REMEIED: Memorias del Encuentro Internacional de Educación a Distancia. Desafíos de la cultura digital en América Latina. ISSN: 2395-8901. Diciembre 2016.
- [8] C. Kazimoglu, M. Kiernan, L. Bacon y L. Mackinnon, “A serious game for developing computational thinking and learning introductory computer programming,” *Procedia - Social and Behavioral Sciences* 47, 2012.
- [9] Dapozo, G.; Greiner, C.; Petris, R.; Espíndola, M.C.; Company, A. Introduction To Programming Based On Playful Activities In The University. XLIII Latin American Computer

- Conference (CLEI) IEEE Xplore digital library. December 2017.
- [10] Dapozo, G.; Greiner, C.; Petris, R.; Espíndola, M.C.; Company, A. Enseñanza de la Programación en la Universidad. Factores que inciden en el buen desempeño de los estudiantes. Actas 5to Congreso Nacional de Ingeniería Informática / Sistemas de Información (CONAIISI 2017). ISSN: 2347-0372. Noviembre de 2017.
- [11] J.C. Olabe, M.A. Olabe, X. Basogain y C. Castaño, "Programming and robotics with Scratch in primary education," *Education in a Technological World: Communicating current and Emerging Research and Technological Efforts*, Badajoz - Spain: Formatex, 356–363, 2011.
- [12] Min Xiao, Xiaohua Yu. A Model of Cultivating Computational Thinking Based on Visual Programming. The Sixth International Conference of Educational Innovation through Technology. 2017.
- [13] R. Heininger, L. Prifti, V. Seifert, M. Utesch and H. Krcmar, "Teaching how to program with a playful approach: A review of success factors," 2017 IEEE Global Engineering Education Conference (EDUCON), Athens, 2017.
- [14] Daisuke Saito, Ayana Sasaki, Hironori Washizaki, Yoshiaki Fukazawa, Yusuke Muto. Program Learning for Beginners: Survey and Taxonomy of Programming Learning Tools. 2017 IEEE 9th International Conference on Engineering Education (ICEED), Kanazawa, Japan. 2017.
- [15] García de la Torre, M.; Mingo Sarto, L.A.; Sáez de Eguilaz Larreta, M.J.; Arteaga Fernández, L.; Labarta Aizpún, M.; Aragón Marín, L; Ambrosio Flores, P.; Piernavieja Marrón, E. Modelo europeo de excelencia. Adaptación a los centros educativos del modelo de la fundación europea para la gestión de calidad. Ministerio de Educación, Cultura y Deporte
- [16] Carlos Vargas Castillo. Cuatro Modelos para Evaluar el Software Educativo. CACIC 2003.
- [17] Bermeo, J., Sánchez, M., Maldonado, J., & Carvallo, J. (2016). Modelos de Calidad de Software en la Práctica: Mejorando su Construcción con el Soporte de Modelos Conceptuales. Universidad de Cuenca, Ecuador.
- [18] Sánchez, M.A. (2016). Assessing the quality of MOOC using ISO/IEC 25010. 2016 XI Latin American Conference on Learning Objects and Technology (LACLO), 1-4.
- [19] Stella Maris Massa, Patricia Pesado. Evaluación de la usabilidad de un Objeto de Aprendizaje por estudiantes. Revista Iberoamericana de Educación en Tecnología y Tecnología en Educación N°8. ISSN 1850-9959. Diciembre 2012.
- [20] Cova, Ángela, Arrieta, Xiomara, & Reveros, Víctor. (2008). Análisis y comparación de diversos modelos de evaluación de software educativo. *Enlace*, 5(3), 045-067.
- [21] L. García-Mundo, J. Vargas-Enríquez, M. Genero y M. Piattini, "¿Contribuye el Uso de Juegos Serios a Mejorar el Aprendizaje en el Área de la Informática?," *Actas de las XX JENUI*. Oviedo, 2014 ISBN: 978-84-697-0774-6
- [22] T. Connolly, L. Boyle y T. Hainey, "A Survey of Students' Motivations for Playing Computer Games: A Comparative Analysis," *En Proceedings of the 1st European conference on games-based learning (ECGBL)*, Paisley, Scotland, 71-78, 2007.