

Modelo de Evaluación de Software Educativo Libre Basado en un Método Sistemático

Estela Fritz¹; Pablo García¹; María Eva Ascheri¹; Alejandra Zangara²

¹Departamento de Matemática
Universidad Nacional de La Pampa
Av. Uruguay 151 – (6300) Santa Rosa – La Pampa – Argentina
Tel.: +54-2954-245220 – Int. 7125
[fritzem, pablogarcia, mavacheri]@exactas.unlpam.edu.ar

²Facultad de Informática
Universidad Nacional de La Plata
50 y 120 – (1900) La Plata – Argentina
Tel.: +54-221-4277270 / 4277271
alejandra.zangara@gmail.com

Resumen

El presente trabajo se enmarca en el Proyecto de Investigación: "*Propuesta de Clasificación de software libre utilizado en la enseñanza de la programación*", que se desarrolla en el ámbito de la Facultad de Ciencias Exactas y Naturales de la Universidad Nacional de La Pampa y que fue acreditado por Resolución N° 160/2018 del Consejo Directivo.

Su principal objetivo es construir un modelo para evaluar cualitativa y cuantitativamente cualquier software educativo utilizado en la enseñanza de la programación y poder incluir en ese modelo también, las características deseables de un software libre. Por otra parte, es importante mencionar que se pretende una evaluación desde el punto de vista pedagógico más que desde el punto de vista técnico, o desde la Ingeniería del software. La construcción de dicho modelo se basa en el método LSP (*Logic Scoring Preferences*)

Palabras clave: Método LSP, evaluación de software, software educativo, software libre, evaluación cuali-cuantitativa.

Contexto

La Directora del proyecto es la Mg. María Eva Ascheri de la Facultad de Ciencias

Exactas y Naturales de la Universidad Nacional de La Pampa (UNLPam) y la Co-Directora es la Dra. Alejandra Zangara de la Facultad de Informática de la Universidad Nacional de La Plata (UNLP). El mencionado proyecto incluye al Mg Ing. Pablo Marcelo García y a la Profesora Estela Marisa Fritz como investigadores. Esta última elabora su Trabajo Final para alcanzar el grado de Especialista en Tecnología Informática Aplicada en Educación en el ámbito del presente proyecto. Además, será la base para desarrollar su futura tesis de maestría a llevarse a cabo en la UNLP.

La incorporación del Mg. García se produce en 2022 y se relaciona con su larga experiencia en el ámbito de la programación (está a cargo, desde 2000, de la asignatura "Lenguajes de Programación" de la carrera "Profesor Universitario en Computación" de la FCEyN – UNLPam). De la misma manera ha trabajado en proyectos de investigación relacionados con el software libre y ha realizado diversas publicaciones en ese ámbito.

1. Introducción

Los rápidos avances en la Tecnología de la Información y la Comunicación y las cuestiones propias de esta época de pandemia, que obligaron a adoptar la virtualidad en la

enseñanza como una modalidad casi excluyente, han incrementado la relevancia de seleccionar software educativo apropiado para poder sostener y enriquecer la labor docente en el ámbito académico, en un espacio donde hasta hace algún tiempo, la presencialidad como modalidad en la enseñanza tenía un rol importante.

Los criterios para la evaluación del software educativo deben ser considerados en una instancia previa al proceso de selección del mismo. Identificar apropiadamente esos criterios es la clave para la evaluación exitosa y posterior selección del software. Por ello este trabajo propone la construcción de un modelo elaborado a través de un método sistemático: *el método LSP*

En este modelo, el docente puede ponderar cada una de las características deseables para el software, de acuerdo con la importancia o relevancia que posea dicha característica en el sistema a seleccionar.

Desafortunadamente, no hay una lista genérica de criterios que puedan ser usados para evaluar cualquier software.

Existen en la bibliografía numerosos modelos para evaluar software educativo. Algunos con énfasis en el proceso de diseño y desarrollo tal como se propone en [1]. Otros enfoques se rigen por los principios propios de la Ingeniería del Software [12]. Se busca en otros casos la calidad del software según los estándares internacionales [3] y [4]. Muchas publicaciones presentan un modelo basado en listas de criterios que se formulan en forma de preguntas. Son ampliamente difundidas en este sentido las listas de control (*check-lists*), aunque contrariamente a lo que propone el método del presente trabajo, la técnica de las listas de control no permite ponderar por relevancia los diferentes ítems. Squires y Mc Dougall [5], proponen un modelo basado en interacciones entre los actores intervinientes desde la etapa de diseño del software hasta su utilización: interacciones entre diseñadores, estudiantes y profesores tomados de dos en dos; algunas de esas interacciones tienen sentido bidireccional. Este modelo propone un

enfoque cualitativo y aporta un modo de pensar acerca de la selección del software.

Sin embargo, cuando evaluamos y seleccionamos software educativo, hay otros aspectos relacionados con el proceso de aprendizaje, que deben ser considerados, a saber:

1. El rol desempeñado por el docente en el contexto de uso del software.
2. Contribución del software educativo al desarrollo de habilidades intelectuales, estrategias cognitivas, información verbal, actitudes y otras.
3. La aproximación a los principios de enseñanza y aprendizaje subyacentes en el sistema de software.

Es por eso que los criterios técnicos ([3], [4]) son insuficientes para determinar la aptitud del software, y los criterios educacionales también deben ser considerados ([2], [5]). Ambas clases de criterios están relacionadas.

1.1 Algunos Aspectos del Diseño Instruccional

Antes de hacer una reseña del método LSP, se consideran algunos tópicos sobre principios de diseño instruccional [6]. Se mencionan aquellos considerados relevantes para establecer criterios elementales y complejos para LSP. En principio, la identificación de los objetivos o metainstruccionales hasta el proceso de evaluación de estos. Existen, según Gagné [6] cinco categorías principales de resultados del aprendizaje, tales como habilidades intelectuales, estrategias cognitivas, información verbal, actitudes y habilidades motoras. La medida en la que el software objeto de la evaluación contribuya a desarrollar alguna o algunas de estas categorías de resultados del aprendizaje, es una característica para considerar como parte de la evaluación. Una mención especial merecen las condiciones de aprendizaje necesarias para la adquisición de esas capacidades. Las diferencias individuales entre los sujetos que aprenden y cómo esas diferencias afectan a la planificación de los

escenarios de aprendizaje, es un tópico para incluir entre los criterios considerados cuando se evalúa un software.

También deben considerarse criterios relativos a las posibilidades de un software de incluir procedimientos de evaluación.

El aprendizaje es un proceso y como tal, provoca cambios en las capacidades del sujeto que aprende. Por eso, es deseable que un software educativo promueva dichos cambios.

Cuando se examina cuidadosamente, la situación de aprendizaje tiene dos partes: una externa al sujeto y una interna al sujeto que aprende. Esa parte interna deriva de aquello que el sujeto puede recordar o recuperar de su memoria. Esas partes (la externa y la interna) están relacionadas. Esa relación existente se denomina *condiciones del aprendizaje*.

1.2 Modelo Cuantitativo

Los modelos cuantitativos buscan objetividad en la evaluación. Para ello utilizan un conjunto de criterios preestablecidos relevantes y coherentes en contraste con evaluaciones subjetivas basadas en puntos de vista particulares (el del docente, el del estudiante, el del experto). Estos criterios preestablecidos valoran diferentes componentes del software, características deseables relacionadas con aspectos pedagógicos y didácticos: adecuación a los objetivos de aprendizaje, enfoque pedagógico (constructivista o cognitivista), orientación hacia los alumnos, grado de intervención docente, inclusión de módulos de evaluación y seguimiento, entre otros.

También es posible incluir en el modelo cuantitativo, variables asociadas con las cuatro libertades de un software para ser considerado software libre. Asimismo, sería posible incluir variables (o criterios) relacionados con aspectos técnicos de diseño de software: accesibilidad, adaptabilidad, interoperabilidad, reusabilidad, por nombrar algunos.

Este modelo supone una evaluación previa del software y los criterios

seleccionados no están relacionados con la utilización de éste por parte de los estudiantes, ya que el proceso de evaluación tiene como finalidad la selección del software más adecuado y que reúna las características buscadas o se aproxime a ello, lo más posible. Sin embargo, el modelo es muy flexible y podrían incluirse variables (criterios) relacionadas con el uso del software una vez que el estudiante interactúa con el mismo.

1.2.1 Construcción del Modelo

Se describe a continuación, en forma muy general, el método en el que se basa el modelo cuantitativo propuesto en este trabajo: El método *LSP (Logic Scoring Preferences)* [8].

LSP es un método cuantitativo basado en el empleo de la lógica continua que posibilita la creación de funciones complejas que intervienen en la evaluación, comparación y posterior selección de sistemas de software de propósito general [9].

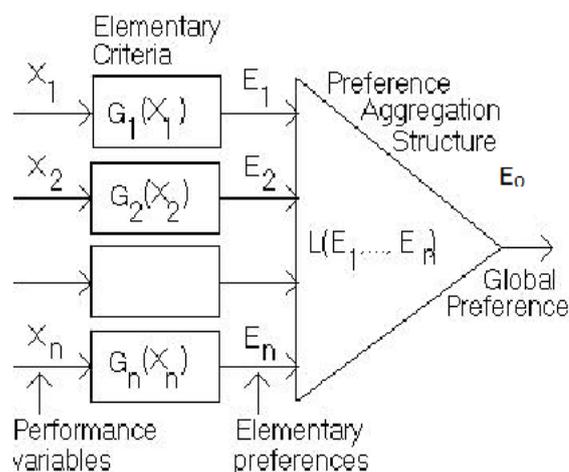


Figura 1. El proceso de evaluación de LSP

Una primera etapa en la construcción de esta función compleja consiste en establecer las características deseables para el software a evaluar. Esas características se transforman según LSP en *variables de performance* (X_1, \dots, X_n). En el caso del software educativo, algunas variables de

performance que se pueden describir son las siguientes: en qué medida es tenido en cuenta el aprendizaje individual, cuantificar fases de aprendizaje inmediato y de largo alcance, gradualidad del software para el aprendizaje (¿Es posible identificar esa gradualidad?), en qué medida el software está basado en el conocimiento de cómo aprende el ser humano, en qué grado contribuye el software a los principios de contigüidad, repetición y refuerzo, qué contenidos de la memoria o capacidades aprendidas el software contribuye a recuperar, qué categorías (en cuanto a resultados del aprendizaje) el software tiende a desarrollar, cuál es el rol del docente en el proceso, en qué medida el sistema guía al sujeto hacia el autoaprendizaje. Esto por mencionar algunas.

Para cada variable de performance es necesario definir un rango aceptable de valores. Los valores de dichas variables, se transforman, por la aplicación de funciones llamadas *criterios elementales* G_i , $1 \leq i \leq n$, en *preferencias elementales* E_i , $1 \leq i \leq n$. Esos criterios elementales transforman el valor de una variable de performance en un valor perteneciente al intervalo $[0,100]$ o $[0,1]$, es decir, $0 \leq E_i \leq 100\%$ y expresan el grado de cumplimiento con un requisito del sistema que está siendo evaluado. Estas preferencias elementales van siendo agregadas mediante operadores de la lógica continua en estructuras de agregación más complejas que permiten obtener un único valor final denominado *Preferencia global E* (E_0 en la fig. 1) [10]. Existe la posibilidad de ponderar las preferencias elementales dentro de la estructura de agregación, mediante pesos W_i , que son coeficientes que multiplican a las preferencias elementales y les otorgan mayor o menor relevancia en la E . La preferencia global se interpreta como el grado global de satisfacción de todos los requerimientos especificados.

2. Líneas de Investigación y Desarrollo

El proyecto de Investigación antes citado aborda los siguientes aspectos relacionados a:

- Relevamiento de software libre utilizado en la enseñanza de la programación. Se prevé incluir aquí varias herramientas que el Mg. García utiliza en la enseñanza de los cuatro paradigmas de programación.
- Construcción de un modelo para la evaluación y selección del software libre, a partir de un método sistemático y basándose en criterios, fundamentalmente pedagógicos.
- Evaluación del software relevado con el fin de clasificarlo según los criterios preestablecidos.

3. Resultados Obtenidos/Esperados

Después de considerar los tópicos explicados anteriormente, como resultado, puede decirse que el método *LSP* propuesto constituye un muy buen método para construir a partir de él, un modelo cuantitativo de evaluación que incrementa la eficiencia de esta y la posterior selección de sistemas de software en general y software educativo en particular con filosofía GNU. Representa un método más ventajoso cuando se lo compara con otros métodos cuantitativos [10]. *LSP* tiene principalmente las siguientes ventajas relativas a los métodos cuantitativos para evaluación de sistemas:

- (1) La especificación de requerimientos es sistemática, flexible y completa.
- (2) El proceso de evaluación es sistemático y racional, el cual refleja explícita y cuantitativamente el nivel global de satisfacción de los requerimientos del usuario.
- (3) Es un modelo que permite objetividad en la evaluación.

El método *LSP* puede ser usado para la evaluación y comparación de una amplia variedad de sistemas complejos [11]. El principal poder de la estrategia de *LSP* es la capacidad de construir un modelo versátil de un conjunto de preferencias. Mediante la combinación apropiada de operadores

(lógicos) de agregación con el conjunto de preferencias, es posible derivar criterios complejos que tienen poder expresivo y flexibilidad.

El software educativo debe satisfacer una variedad de requerimientos. El nivel global de satisfacción de esos requerimientos (E) es usado para su evaluación y comparación.

4. Formación de Recursos Humanos

En el marco del presente proyecto, como ya se mencionó anteriormente, se espera que la Profesora Fritz alcance el grado de Especialista en Tecnología Informática Aplicada en Educación y que, en un futuro, pueda desarrollar su tesis de maestría en la UNLP.

Actualmente, la Profesora Fritz ha completado la etapa de búsqueda y análisis de criterios relevantes para la posterior construcción del modelo de evaluación del software principalmente, tema en el que no era experta. Esto constituye un insumo necesario para los nuevos avances en el proyecto. Se espera que durante el año 2022 se realice un fuerte trabajo de evaluación de múltiples herramientas de software libre que se utilizan en la cátedra a cargo del Mg. García.

5. Bibliografía

- [1] **Zaragoza, J., Casado, A.** (1992) “Aspectos técnicos y pedagógicos del ordenador en la escuela”. Editorial Bruno, Madrid, España.
- [2] **Gorga, G., Madoz, M., Pesado P.** (2000) “Hacia una propuesta de métrica para la evaluación de Software Educativo”, CACIC.
En línea: <http://hdl.handle.net/10915/23514>
- [3] **International Standard. ISO/IEC 9126(1991)** Information Technology – Software Product Evaluation – Quality characteristics and guidelines for their use.
- [4] **International Standard ISO/IEC 9001(1991)** Quality Systems – Model for quality assurance in design/development production, installation and servicing.
- [5] **Squires, D., McDougall, A.** (1997) “Cómo elegir y utilizar software educativo”. Traducción Pablo Manzano. Ediciones Morata y Fundación Pandeia, Madrid, 1° edición.
- [6] **Gagné, R.M., Briggs, L., Wager, W.** (1992) Ed. Harcourt Brace College Publishers, 4th edition.
- [7] **Stallman, R.** (2004) “Software libre para una Sociedad libre”. Traficantes de Sueños, Madrid.
- [8] **Dujmovic, J.J. and Elnicki, R.** (1982) “A DMS Cost/Benefit Decision Model: Mathematical Models for Data Management System Evaluation, Comparison, and Selection”. National Bureau of Standards, Washington. DC., No NBS-GCR, NTIS No PB82- 170150 (155 pages).
- [9] **Dujmovic, J.J.** (1996) “A Method for Evaluation and Selection of Complex Hardware and Software Systems”. Proceedings of the 22nd International Conference for the Resource Management and Performance Evaluation of Enterprise CS, CMG 96, 1, 368-378.
- [10] **Daso, A. et al** (2013) “Desarrollo de Modelos de Evaluación Usando Operadores de una Lógica Continua”. XV WICC (Workshop de Investigadores en Ciencias de la Computación), págs. 420-424.
- [11] **Dujmović, J. J., Hajime Nagashima.** (2006) “LSP Method and Its Use for Evaluation of Java IDEs.” International Journal of Approximate Reasoning 41.1: 3–22. Web.
- [12] **Galvis, A.** (1995). “Fundamentos de tecnología educativa”. Editorial de la Universidad Estatal a Distancia. San José, Costa Rica.