

# Evaluación y Selección de Software Educativo utilizando el método LSP

Estela Fritz<sup>1</sup>; María Eva Ascheri<sup>1</sup>; Alejandra Zangara<sup>2</sup>

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

<sup>2</sup> Facultad 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.

El principal objetivo de este trabajo es generar un solo valor como resultado de la evaluación de un software, haciendo uso del método *LSP*. Este trabajo presenta además, una propuesta para incluir algunos criterios pedagógicos relacionados con la evaluación y selección del software educativo.

**Palabras clave:** Método LSP, evaluación de software, software educativo, métodos cuantitativos.

## Contexto

Por Resolución N° 160/18 del Consejo Directivo de la Facultad de Ciencias Exactas y Naturales de la Universidad Nacional de La Pampa se acredita el Proyecto de Investigación: "*Propuesta de Clasificación de*

*software libre utilizado en la enseñanza de la programación*". El mismo es dirigido por la Magister María Eva Ascheri de la Facultad de Ciencias Exactas y Naturales de la Universidad Nacional de La Pampa (U.N.L.Pam) y co-dirigido por la Magister Alejandra Zangara de la Facultad de Informática de la Universidad Nacional de La Plata (UNLP). El mencionado proyecto incluye a la Profesora Estela Marisa Fritz como investigadora. Constituye su Trabajo Final para alcanzar el grado de Especialista en Tecnología Informática Aplicada en Educación. Además, será la base para desarrollar su futura tesis de maestría a llevarse a cabo en la UNLP.

## 1. Introducción

Los rápidos avances en Tecnología de la Información y la Comunicación sumados a la diversidad de software educativo, tornan casi imposible tomar decisiones que conduzcan a una mejora académica óptima. Los criterios para la selección del software educativo deben ser considerados en una instancia previa al proceso de evaluación y selección del software. Identificar apropiadamente esos criterios es la clave para

la evaluación exitosa y posterior selección del software.

Desafortunadamente, no hay una lista genérica de criterios que puedan ser usados para evaluar cualquier software [1]. Los investigadores generalmente proveen una lista de criterios relacionados con la correctitud, eficiencia, usabilidad [2] y proponen métodos de evaluación basados en encuestas al usuario, cuyas respuestas organizan características según una jerarquía: buena, muy buena, aceptable, no tan buena, entre otras.

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.

Como fue mencionado antes, los criterios de selección de software educativo incluyen tanto aspectos técnicos del software como pedagógicos.

### **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 metas

instruccionales hasta el proceso de evaluación de estos. Se mencionan también las 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 Algunos principios del aprendizaje**

Algunos de estos principios expanden las ideas expuestas. Uno de ellos es el conocimiento de las condiciones del aprendizaje humano. Aún hay tres principios derivados de la teoría del aprendizaje, que son relevantes: (1) Contigüidad: las situaciones de estímulo deben ser presentadas en el mismo tiempo que las respuestas esperadas. (2) Repetición: esa situación de estímulo y la respuesta del sujeto, necesitan ser repetidas. (3) Refuerzo: para los propósitos de la

instrucción, es necesario que un nuevo aprendizaje sea seguido inmediatamente por otro ya logrado y que el sujeto que aprende sea capaz de relacionarlos, de modo que aquel aprendizaje ya logrado se constituya en un refuerzo de lo nuevo.

Estos tres principios deberían ser incluidos en el proceso de evaluación y selección de un software educativo.

### 1.3 Reseña del método LSP

El método LSP (Logic Scoring Preferences) [7] es una generalización y extensión de varias técnicas de scoring. Es un método cuantitativo basado en el empleo de la lógica continua que permite la creación de funciones complejas de evaluación y su aplicación en la evaluación, optimización, comparación y selección de sistemas de software de propósito general [8].

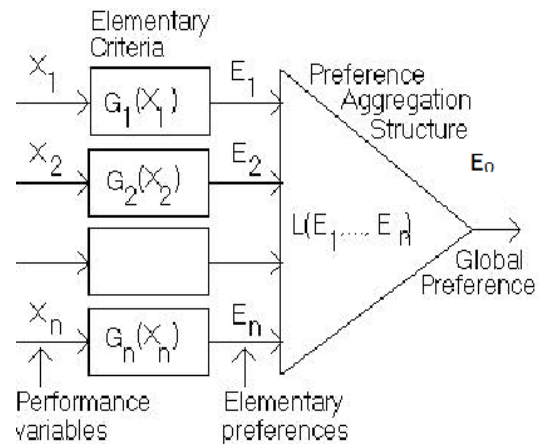
Una vez identificadas las características que intervienen en la evaluación del software, los datos obtenidos de las distintas características del software, de acuerdo con LSP, serían las variables de performance del sistema. En el caso del software educativo, algunas variables de performance que se pueden 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.

Los valores de dichas variables son mapeados por medio de funciones

denominadas *criterios elementales*, en *preferencias elementales*. Son estas preferencias las que van siendo agregadas mediante operadores de la Lógica continua en estructuras de agregación que nos permiten obtener un único valor final (*preferencia global final*  $E_0$  en la fig. 1).

Los criterios elementales son funciones que transforman un valor real de una variable de performance, en un valor perteneciente al intervalo  $[0,100]$  o  $[0,1]$  y expresan el grado de cumplimiento con un requisito del sistema que está siendo evaluado.

Figura 1. El proceso de evaluación de LSP



## 2. Líneas de Investigación y Desarrollo

El proyecto de Investigación citado aborda los siguientes aspectos relacionados:

- Relevamiento de software libre utilizado en la enseñanza de la programación.
- Descripción de un método cuantitativo para la evaluación y selección del software basándose en criterios tanto técnicos como pedagógicos.
- Evaluación del software relevado con el fin de clasificarlo según las habilidades intelectuales y estrategias cognitivas que permite desarrollar en el sujeto que aprende.

El presente trabajo corresponde al segundo de los aspectos mencionados.

### 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 cuantitativo que incrementa la eficiencia de la evaluación y la selección de sistemas de software en general y software educativo en particular. 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 sistemático y racional, el cual refleja explícita y cuantitativamente el nivel global de satisfacción de los requerimientos del usuario.

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 ( $E_0$ ) es usado para su evaluación y comparación.

Por otro lado, se espera que esta investigación brinde a los docentes herramientas cuantitativas para la evaluación y selección de software, en función de los objetivos que desean alcanzar.

### 4. Formación de Recursos Humanos

En el marco del presente proyecto, como ya se mencionó anteriormente, se espera que la Prof. 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 Prof. Fritz ha completado la etapa de búsqueda y análisis de métodos de evaluación del software, tema en el que no era experta. Durante 2019, realizará todos los aportes relacionados con la definición de criterios para evaluar software educativo para la enseñanza de la programación, principalmente desde el punto de vista pedagógico, y tomando como base alguna teoría del aprendizaje en particular. Esto constituye un insumo necesario para los nuevos avances en el proyecto.

### 5. Bibliografía

- [1] **Abohamad, W., Arisha, A.** (2010) “Evaluating and Selecting Optimization Software Packages: A Framework for Business Application” World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering. Vol. 4 <https://waset.org/publications/4426/evaluating-and-selecting-optimization-software-packages-a-framework-for-business-applications>
- [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 25012** (2008) Software Engineering – Software Quality Requirements and Evaluation (SQuARE) - Data Quality Model.
- [4] **International Standard ISO/IEC 25010** (2011) Systems and software engineering — Systems and software Quality

Requirements and Evaluation (SQuaRE) —  
System and software quality models.

- [5] **Cataldi, Z.**, (2000) “Una Metodología para el Diseño, Desarrollo y Evaluación de Software Educativo”, Universidad Nacional de La Plata, Facultad de Informática, Tesis de Maestría.  
En línea: <http://hdl.handle.net/10915/4055>
- [6] **Gagné, R.M., Briggs, L., Wager, W.** (1992) Ed. Harcourt Brace College Publishers, 4<sup>th</sup> edition.
- [7] **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).
- [8] **Dujmovic, J.J** “A Method for Evaluation and Selection of Complex Hardware and Software Systems  
<https://telin.ugent.be/~gdetre/Summer%20School%20Poland/Resources/Decision%20support.pdf>
- [9] **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.
- [10] **Squires,D., McDougall, A.**(1997) “Cómo elegir y utilizar software educativo”, Ediciones Morata y Fundación Paideia, Primera Edición.
- [11] **Dujmović, Jozo J., and Hajime Nagashima.** (2006) “LSP Method and Its Use for Evaluation of Java IDEs.” International Journal of Approximate Reasoning 41.1: 3–22. Web.