

# SOFTWARE EDUCATIVO PARA EL TRATAMIENTO DE ALGUNOS TEMAS DE CÁLCULO NUMÉRICO: PRIMERA ETAPA DE EVALUACIÓN

María E. Ascheri, Rubén A. Pizarro, Gustavo J. Astudillo, Pablo García, María E. Culla  
Facultad de Ciencias Exactas y Naturales - Universidad Nacional de La Pampa

## CONTEXTO

Nuestro Proyecto de Investigación, acreditado en la Facultad de Ciencias Exactas y Naturales, aborda líneas de investigación estrechamente vinculadas con varios de los Proyectos que se desarrollan en el Departamento de Matemática del cual formamos parte. Algunos de ellos se relacionan con la ingeniería didáctica, la inclusión de tecnologías en el proceso de enseñanza aprendizaje y la importancia de la visualización gráfica en la enseñanza de la matemática. Por tal motivo, el desarrollo del presente Proyecto demanda una continua interacción con los investigadores de los demás Proyectos de la Institución, actuando en beneficio de sus avances.

## RESUMEN

En el presente Proyecto de Investigación, estamos desarrollando un software educativo para la enseñanza y el aprendizaje de los temas “Resolución Numérica de Ecuaciones no lineales” e “Interpolación y Aproximación Polinomial y Ajuste de Curvas por Mínimos Cuadrados”.

En una primera etapa, se hizo un relevamiento de herramientas para desarrollo Web libres que permitieran el diseño de este nuevo software.

Luego de seleccionar las herramientas para el diseño de la aplicación, se elaboró un primer prototipo que evolucionó hasta convertirse en una versión beta que permite la enseñanza y el aprendizaje del tema “Resolución Numérica de Ecuaciones no lineales”.

Actualmente, dicha versión está siendo analizada y evaluada desde tres puntos de vista: el matemático, el pedagógico y de su usabilidad.

Concretado dicho análisis, este recurso pedagógico será utilizado y evaluado en el curso de Cálculo Numérico que se dicta para las carreras de Profesorado en Matemática (3°

Año), Licenciatura en Física (3° Año) e Ingeniería Civil (2° Año).

El objetivo final que se pretende alcanzar con la elaboración de este software es el de disponer de un material didáctico que permita facilitar el proceso de enseñanza – aprendizaje de las temáticas involucradas en Cálculo Numérico. Disponer de un sistema en línea amigable con el usuario ofrecerá un buen soporte al docente de esta asignatura y de otras vinculadas con los contenidos que aquí se abordan.

*Palabras clave:* software educativo, cálculo numérico, software libre

## 1. INTRODUCCIÓN

Habiendo finalizado exitosamente con el anterior Proyecto de Investigación, titulado *Software educativo para la enseñanza – aprendizaje de temas de Cálculo Numérico*, nos propusimos continuar trabajando dentro de la misma línea de investigación. En el Proyecto anterior desarrollamos un software educativo, utilizando el paquete MATLAB, para la enseñanza-aprendizaje del tema: “Resolución Numérica de Ecuaciones no Lineales”. Actualmente, estamos desarrollando un software educativo que incluya el tema mencionado anteriormente y además, “Interpolación y Aproximación Polinomial y Ajuste de Curvas por Mínimos Cuadrados”.

La diferencia fundamental entre el anterior Proyecto y el actual es que éste está siendo desarrollado íntegramente con software libre. Se está utilizando PHP, HTML, CSS, la librería JGraph y GIMP para el diseño y edición de imágenes (en formato PNG).

En la primera etapa del Proyecto, navegamos la Web en busca de aplicaciones destinadas a la enseñanza y el aprendizaje de temas de Cálculo Numérico (particularmente, de aquellos que se abordarán en este software).

Los resultados de esta búsqueda fueron decepcionantes, ya que no se hallaron aplicaciones de acceso libre que abordaran los temas dinámicamente. Existen, en general, un gran número de sitios que ofrecen los algoritmos implementados en diversos lenguajes o applets que aplican los métodos para funciones fijas, o simplemente arrojan los resultados de la aplicación del método. Esto nos dio la pauta de que deberíamos desarrollar nuestro software educativo desde cero.

El desafío no estaba únicamente en desarrollar una aplicación Web, sino que además debería ser un software educativo. Debíamos aclarar, entonces, qué entendemos por software educativo. Consideraremos software educativo a toda aplicación informática que se haya diseñado intencionalmente para impactar en el proceso de enseñanza-aprendizaje. Tomaremos la definición dada por Pere Marquès (1996, p.2): “*programas para ordenador creados con la finalidad específica de ser utilizados como medio didáctico, es decir, para facilitar los procesos de enseñanza y de aprendizaje*”.

Teniendo esto como referencia, avanzamos en la selección de herramientas de libre acceso que nos permitieran hacer el desarrollo de un software libre. Esto es, “*el software libre es aquél que puede ser distribuido, modificado, copiado y usado; por lo tanto, debe venir acompañado del código fuente para hacer efectivas las libertades que lo caracterizan*.” (Culebro Juárez, et al., 2006). Estas libertades, según la Free Software Foundation, son:

0. la libertad para ejecutar el programa sea cual sea nuestro propósito,
1. la libertad para estudiar el funcionamiento del programa y adaptarlo,
2. la libertad para redistribuir copias
3. la libertad para mejorar el programa y luego publicarlo para el bien de toda la comunidad.

Teniendo en vista los aspectos mencionados en los párrafos anteriores es que realizamos, durante el año 2008, el primer prototipo que evolucionó hasta convertirse en la primera versión beta de la aplicación (ver figura 1).

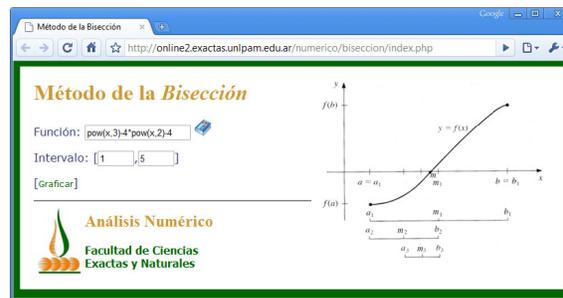


Figura 1. Captura de pantalla del primer prototipo

A finales del año 2008 y principios del 2009, se comenzó con la evaluación de la aplicación. Para esta primera etapa de la evaluación, se utilizó una técnica utilizada para evaluar sistemas denominada *caminata cognitiva*. En esta técnica “*un grupo de expertos simula la manera en como un usuario caminaría por la interfaz al enfrentarse a tareas particulares*.” (Baeza Yates & Rivera Loaiza, 2002, p.8). Esto nos permitió identificar un conjunto de fortalezas y debilidades en la aplicación desde los puntos de vista didáctico-matemático, del aprendizaje, del tratamiento del error y de la usabilidad.

Desde el punto de vista didáctico-matemático, se analizó cuál es el encuadre en el que se encuentra la aplicación acorde con la enseñanza de la matemática. También, se utilizó la clasificación propuesta por Kemmis et al. (1977) para identificar la fundamentación educativa de la aplicación, y clasificarla dentro del paradigma instruccional, revelatorio o conjetural. El primero, paradigma instruccional, deriva la instrucción programada (Skinner), y se caracteriza por el refuerzo inmediato y el avance en pequeños pasos debido a la atomización de tareas complejas. Los programas son del estilo drill-and-practice. Se le presenta al alumno actividades con el objetivo de avanzar en una dirección determinada. Por cada paso (en esa dirección) el alumno recibe un refuerzo en términos de correcto/incorrecto (retroalimentación). “*El rol de la computadora es la presentación de los contenidos, la prescripción de la tarea y la motivación del estudiante a través de una rápida retroalimentación*” (Kemmis et al., 1977, p.25). El segundo paradigma, el revelatorio, denominado así porque el conocimiento se le va revelando gradualmente

al alumno, se basa en las teorías de Bruner y Ausubel. Los programas del tipo simulación tienen la raíz en este paradigma. Las actividades que se proponen tienen la intención de reducir la brecha entre lo que sabe el alumno y el contenido (brecha cognitiva). Estas actividades deben estar diseñadas de manera que los alumnos accedan a los contenidos a partir del descubrimiento. “*La función de la computadora es la simulación o el manejo de información*” (Kemmis et al., 1977, p.26). El tercer paradigma, el conjetural, se basa en las teorías de Popper, Piaget y Paper. Propone la construcción del conocimiento a través de la experiencia. Las actividades deben propiciar el acceso al conocimiento a través de la generación y contratación de hipótesis. “*El rol de la computadora es manipular espacios, campos, lenguajes para la creación de modelos, programas, planes o estructuras conceptuales*” (Kemmis et al., 1977, p.27). Entrar en esta última clasificación los paquetes de modelización e inteligencia artificial.

Para identificar el tipo de aprendizajes que estimula el software se utilizó el análisis hecho por Squires y Mc Dougall (1994) de las relaciones entre los paradigmas antes mencionados y las habilidades cognitivas, también propuestas por Kemmis et al. La Tabla 1 muestra dichas relaciones.

Paradigma	Tipo de interacción				
	Reconocimiento	Recuerdo	Comprensión	Reconstrucción Global	Interpretación Constructiva
Instructivo	←—————■—————→				
Revelador	←—————■—————→				
De conjeturas	←—————■—————→				

Tabla 1. Posibles interacciones entre paradigmas y tipos de interacción cognitiva propuesto por Squires y Mc Dougall (1994, p.74)

Para evaluar la usabilidad, es decir, la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario en condiciones específicas de uso (Wikipedia, 2009), utilizamos las heurísticas adaptadas por Instone (1997) y citadas por Baeza Yates et al. (2002) para evaluar la usabilidad:

- Visibilidad del estado del sistema.

- Similitud entre el sistema y el mundo real.
- Control por parte del usuario y libertad.
- Consistencia y cumplimiento de estándares.
- Prevención de errores.
- Preferencia al reconocimiento frente a la memorización.
- Flexibilidad y eficiencia de uso.
- Estética y diseño minimalista.
- Ayuda para que el usuario reconozca, diagnostique y se recupere de los errores.
- Ayuda y documentación.

El análisis dio como resultado un conjunto de recomendaciones que nos permitieron realizar varias mejoras al software, que implican desde una mejor navegabilidad y opciones más claras para facilitar la resolución de los ejercicios y la visualización de los resultados, hasta una visión global de los métodos en la asignatura e información de sus autores (ver figura 2).



Figura 2. Captura de pantalla de la versión mejorada

## 2. LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

Las líneas de investigación y desarrollo que seguimos son:

- Búsqueda y análisis de herramientas libres que permitan desarrollar software educativo con acceso Web.
- Desarrollo de software educativo para la enseñanza-aprendizaje de algunos de los métodos de resolución de ecuaciones no lineales.
- Elaboración de una planificación adecuada que permita la eficiente inserción del software en el desarrollo de las clases, optimizando de esta forma los resultados esperados.

### 3. RESULTADOS OBTENIDOS/ESPERADOS

Hasta el momento se ha logrado obtener una completa bibliografía sobre los temas que nos ocupan y sobre los software existentes. Es decir, se han podido detectar las herramientas a utilizar para el desarrollo del software y conocer además, los software educativos existentes destinados a temas de Cálculo Numérico.

También, se ha desarrollado el software para la implementación de algunos de los métodos de resolución de ecuaciones no lineales, tales como bisección, regla falsi y secante.

Actualmente, se está trabajando en el diseño de los métodos de Newton-Raphson, de von Mises e iterativo de punto fijo, y en la planificación para poder implementar estos métodos en las clases de Cálculo Numérico en el transcurso del segundo cuatrimestre del corriente año. Con ello, se espera poder llegar hacer los reajustes necesarios a efectos de optimizar el diseño y posterior uso del software educativo que estamos desarrollando.

### 4. FORMACION DE RECURSOS HUMANOS

Se han presentado y publicado en Reuniones Científicas de carácter nacional e internacional, varios trabajos de investigación que han surgido de este Proyecto. A partir de estas presentaciones, se ha podido hacer un relevamiento sobre los avances producidos en los temas involucrados en este Proyecto y se han intercambiado opiniones con otros grupos de investigación del área.

De esta forma, se han adquirido compromisos tendientes a realizar intercambios de las producciones que surjan de los Proyectos implicados, logrando así ampliar la población a la que estará dirigida la implementación del software educativo y que no se restrinja sólo a la cátedra de Cálculo Numérico, docentes y alumnos.

### 5. BIBLIOGRAFIA

**Ausubel, D., Novak J., Hanesian H.** (1997). *Psicología educativa. Un punto de vista cognitiva*. México. Trillas.

**Baeza Yates, R. & Rivera Loaiza, C.** (2002). *Ubicuidad y Usabilidad en la Web*. Consultado en Febrero, 11, 2009 en <http://www.dcc.uchile.cl/~rbaeza/inf/usabilidad.html>.

**Bruner, J.** (1960). *The Process of Education*. Cambridge: Harvard University Press.

**Culebro Juárez, M., Gómez Herrera, W. y Torres Sánchez, S.** (2006). *Software libre vs software propietario. Ventajas y desventajas*. México: (CC) Creative Commons. Consultado en Marzo, 16, 2009 en: <http://www.softwarelibre.cl/drupal/files/32693.pdf>.

**Instone, K.** (1997). *Site Usability Evaluation – Part. 1*. Consultado en Febrero, 12, 2009 en <http://instone.org/siteeval>.

**Marqués, P.** (1996). *El software educativo*. Universidad Autónoma de Barcelona. Consultado en Enero, 26, 2009 en: [http://www.lmi.ub.es/te/any96/marques\\_software](http://www.lmi.ub.es/te/any96/marques_software)

**Papert, S.** (1987). *Desafío de la mente: Computadoras y educación*. Buenos Aires, Galápagos.

**Piaget, J.** (1985): *Psicología y Pedagogía*. Barcelona: Ariel.

**Skinner, B.** (1985). *Aprendizaje y comportamiento*. Barcelona. Martínez-Roca.

**Squires D. y Mc Dougall A.** (1994). *Cómo elegir y utilizar software educativo*. Morata. Barcelona.

**Usabilidad.** (2009, 28 de enero). *Wikipedia, La enciclopedia libre*. Fecha de consulta: 22:41, febrero 11, 2009 from <http://es.wikipedia.org/w/index.php?title=Usabilidad&oldid=23617834>.