

## UN SOFTWARE EDUCATIVO CON HERRAMIENTAS LIBRES Y ACCESO WEB PARA TEMAS DE CÁLCULO NUMÉRICO: PRIMER PROTOTIPO

**María E. ASCHERI, Rubén A. PIZARRO, Pablo GARCÍA,  
Gustavo J. ASTUDILLO, María E. CULLA**

*Facultad de Ciencias Exactas y Naturales - UNLPam  
Uruguay 151 - (6300) Santa Rosa - La Pampa - Argentina  
mavacheri@exactas.unlpam.edu.ar*

**Nivel Educativo:** Educación Continua, Superior y de Postgrado.

**Palabras Clave:** herramientas libres, web, método de bisección.

### RESUMEN

En este trabajo presentamos las principales características del software educativo que nos proponemos elaborar para el tratamiento de los temas “Interpolación y Aproximación Polinomial y Ajuste de Curvas por Mínimos Cuadrados”, empleando herramientas libres. Dicho software incluirá, además, el que ya hemos desarrollado para el tema “Resolución Numérica de Ecuaciones no lineales” utilizando programas comerciales. De este software previo, hemos presentado en reuniones científicas los resultados obtenidos a partir de su uso en el desarrollo de las clases de Cálculo Numérico.

Luego de realizar un relevamiento acerca de las herramientas disponibles que permitieran colocar software en línea con el fin de crear nuevos ambientes de aprendizaje, se decidió usar HTML, PHP y JGRAPH para desarrollar este nuevo software educativo.

Este recurso pedagógico, se utilizará también 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), aunque además podrá aplicarse a otras disciplinas científicas y, dado que está previsto su disponibilidad en la Web, podrá ser accedido por quien así lo desee. Presentaremos aquí un esquema del análisis y el diseño que se realizó previamente al desarrollo, el diagrama de estructuras obtenido, algunas herramientas de programación utilizadas y un primer prototipo obtenido para el método de bisección.

### INTRODUCCIÓN

La Informática ha cambiado la forma de relacionarnos con nuestros objetos de estudios y la Matemática no está ajena a esta situación. Es a través de software para Matemática que esta relación se establece. Este tipo de software aparece como una alternativa tanto para docentes como científicos que desarrollan sus actividades ligadas a la Matemática. De las distintas categorías de software para Matemática, nos interesa el software educativo, esto es, aquél que es desarrollado con una intencionalidad pedagógica (Marquès, 1999).

En el contexto de la clasificación de software, se puede hablar también de software libre y de software propietario. En la primera categoría, entran aquellos programas que permiten al usuario ejecutar, estudiar, redistribuir y mejorar el software. Y en la segunda categoría, se encuentran aquellos para los cuales debe comprarse una licencia para su utilización (Gómez Sánchez, 2004).

Para el desarrollo de software educativo para Matemática, hemos optado, en este caso, por la utilización de herramientas libres.

Diversos son los estudios realizados en los que se confirman las ventajas de que el alumno pueda visualizar los conceptos en cuyo aprendizaje está participando (Hitt, 2003; Font et al., 2007). Esta visualización garantiza que pueda construir el concepto estudiado e identificar las diversas variables que intervienen para su obtención.

La utilización de software educativo sin dudas proporciona un gran número de posibilidades que facilitan las diferentes representaciones gráficas de los contenidos estudiados. En nuestro caso y en una primera instancia, los diferentes métodos de resolución numérica y gráfica de ecuaciones no lineales.

Si a las posibilidades que le reconocemos al software educativo para facilitar la enseñanza y el aprendizaje de contenidos de Matemática (Maita Guedez, 2005) le sumamos la posibilidad de que dicho software sea de características libre y de acceso por medio de la Web, sin dudas estaremos mejorando su calidad. Habremos superado de esta manera, inconvenientes de instalación y acceso a los diferentes software que siempre son un escollo a salvar al momento de implementarlos; más aún, si consideramos que los usuarios no tienen que ser necesariamente expertos en la instalación de diversos programas.

En el caso de los docentes, en general, sólo lo usan para diseñar sus clases y, en menor medida, para el trabajo en el aula (Pérez Sanz, 2006). Una de las principales causas de que esto suceda es tener que utilizar una nueva interfase para cada tema que aborda, ya que debe cambiar de software. La utilización de una interfase Web, permite que tanto docentes como alumnos conozcan de antemano la manera de operar el programa, al menos básicamente.

## ANÁLISIS Y DISEÑO

La principal característica del software educativo que desarrollaremos es que se elaborará completamente con herramientas de software libre y de tal manera que pueda luego ser publicado para su consulta en línea.

El software educativo que haremos es una continuación de un proyecto anterior en el cual se desarrolló la parte correspondiente al contenido curricular de Cálculo Numérico *resolución de ecuaciones no lineales*, usando como soporte informático el paquete MATLAB (Ascheri y Pizarro, 2005 y 2006; Pizarro y Ascheri, 2007). Este contenido será incorporado al nuevo software que elaboraremos, al que se le agregarán los relacionados con *Interpolación y Aproximación Polinomial y Ajuste de Curvas por Mínimos Cuadrados* (Nakamura, 1997; Chapra y Canale, 2007).

La dificultad que presenta el software educativo desarrollado anteriormente es que necesita contar con licencias (no demasiado accesibles debido a sus costos) en todos los equipos en que se lo instale. Esto obstaculiza, sin dudas, las posibilidades de fácil y libre acceso.

Se analizaron entonces, distintas alternativas para resolver la problemática mencionada (Ascheri et al., 2007). De este estudio, surgieron como herramientas viables HTML, OCTAVE, PHP, JAVA y JPGRAPH. De todas ellas, seleccionamos:

- **HTML.** Es un tipo especial de texto que puede ser utilizado para que los navegadores de Internet muestren textos y gráficos (Raggett, 2005). Las páginas desarrolladas en este lenguaje pueden estar alojadas en un servidor Web o pueden ser descargadas y vistas con el navegador sin necesitar una conexión a Internet. A partir de la programación con este lenguaje, se desarrollará todo lo relativo a pantallas de interfase de manera totalmente visual.
- **PHP.** Es un lenguaje de programación interpretado que permite la creación de páginas Web dinámicas; esto es, las páginas PHP alojadas en el servidor son creadas y devueltas al cliente (navegador) en el momento que son pedidas (Lerdof y Tatroe, 2002). Utilizando este lenguaje se desarrollará lo relacionado con la aplicación de

cada método, ya que aporta potencia de cálculo, buenas estructuras de control, ejecución en el servidor y flexibilidad en el manejo de los datos.

- **JPGGRAPH.** Esta librería de PHP proporciona una alternativa muy importante para el manejo de la parte gráfica. Es un desarrollo de Aditus Consulting que es de uso libre para fines educativos (Persson, 2003), si bien exige una licencia para usos comerciales. La versión que estamos utilizando (2.3) es compatible con PHP5 y no funciona con PHP4 (Achour et al., 2003-2004).

Considerando el software desarrollado con anterioridad utilizando el paquete MATLAB, se mantendrá en esta nueva versión la estructura definida en aquella oportunidad, a saber:

- a) Menú de selección del método numérico que se va a utilizar.
- b) Para cada método, menú de ingreso de los datos y validación de los mismos.
- c) Aplicación del método numérico seleccionado con los datos elegidos.

Primero, se implementarán los mismos métodos numéricos incluidos en el software anterior:

- **Bisección**
- **Iterativo de punto fijo**
- **Secante**
- **Regula falsi**
- **Newton**
- **Newton modificado**

Luego, se implementarán los métodos numéricos correspondientes a los temas *Interpolación y Aproximación Polinomial y Ajuste de Curvas por Mínimos Cuadrados*, tales como:

- **Interpolación de Newton con diferencias divididas**
- **Interpolación de Newton con diferencias finitas**
- **Interpolación de Lagrange**
- **Regresión lineal**
- **Regresión polinomial**

### DIAGRAMA DE ESTRUCTURAS

En el siguiente diagrama (Figura 1), se consideran solamente dos métodos numéricos, bisección e iterativo de punto fijo (incluir los demás sólo implica el agregado de similares ramas en él, una para cada método):

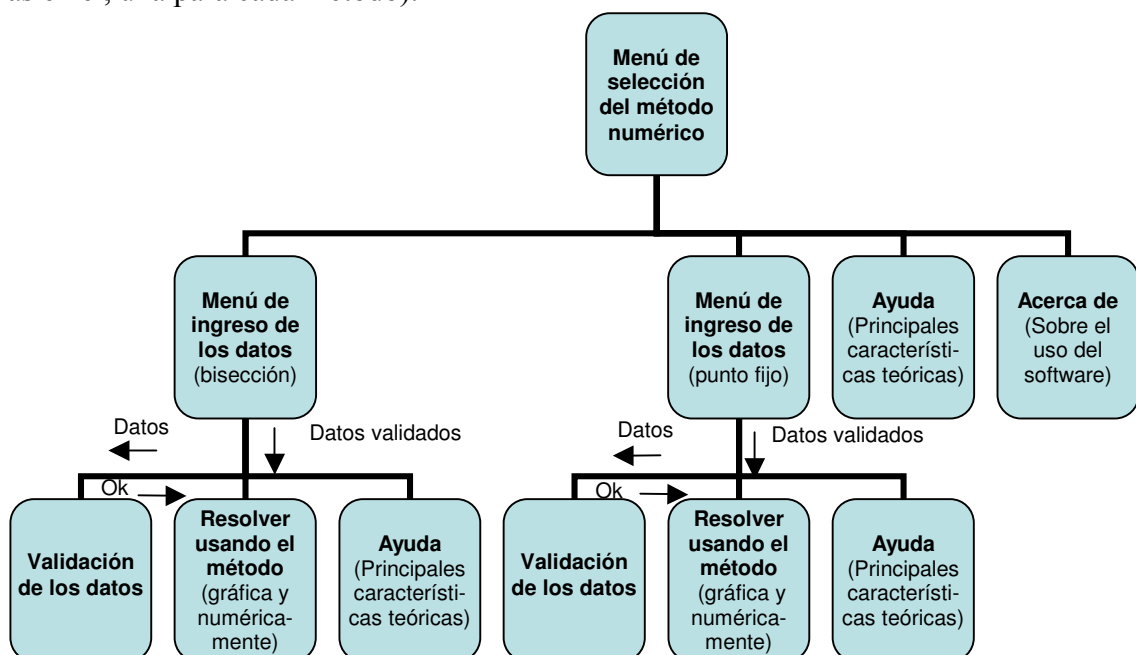


Figura 1. Diagrama de estructuras para los métodos de bisección e iterativo de punto fijo.

Explicaremos el significado de los datos que se muestran en el diagrama:

**Datos:** implica el conjunto de parámetros que cada método numérico necesita para operar.

**Ok:** indica que los datos que el módulo “Validación de los datos” recibió son correctos. Por lo tanto, la aplicación del método numérico puede llevarse a cabo.

**Datos validados:** son los datos que ya pasaron el proceso de validación y se pasan al módulo “Resolver usando el método”.

## Validación de los datos

Los métodos numéricos exigen la validez de determinadas condiciones. Por lo tanto, la validación de los datos debe apuntar a dos situaciones básicas:

- a) Si el tipo de dato ingresado es incorrecto, debe volver a solicitarlo.
- b) Si un dato en particular o el conjunto de los mismos presenta incompatibilidades con el método numérico utilizado, debe realizarse la validación sintáctica y solicitar el reingreso de una o más entradas.

Adicionalmente, cada método numérico utilizado, de acuerdo a su forma de operar, presenta características propias:

- c) Cada uno requiere una cantidad determinada de datos.
- d) Según el método numérico usado, la semántica de la validación toma formas diferentes.

Vamos a mostrar en dos casos cómo se debe llevar a cabo la validación de los datos:

### 1. Método de bisección

Los datos son:

- La función, de cuya ecuación se desea calcular la raíz.
- Valor inicial en la escala X del gráfico.
- Valor final en la escala X del gráfico.
- Número máximo de iteraciones.
- Error admitido.

Las reglas de validación a aplicar en este caso son:

- Validación de la corrección sintáctica de la función ingresada.
- Se debe verificar que la evaluación de la función en los extremos del intervalo seleccionado (valor inicial-valor final) tenga signos opuestos. Esta es una condición que exige este método numérico en particular.
- Se debe verificar que los extremos del intervalo y el error admitido sean reales.
- Se debe cumplir que el valor inicial sea menor que el final.
- Se debe exigir un valor entero y positivo para el número máximo de iteraciones.
- El error admitido debe ser una cantidad positiva. Cuánto más pequeño, mayor exactitud se logrará en la solución requerida.

### 2. Método iterativo de punto fijo

Los datos son:

- La función, de cuya ecuación se desea calcular la raíz.
- La función generadora que el método utiliza.
- Valor inicial en la escala X del gráfico.
- Valor final en la escala X del gráfico.
- Valor inicial para comenzar a iterar.
- Número máximo de iteraciones.
- Error admitido.

Para este método se requiere graficar ambas funciones en el intervalo elegido (valor inicial-valor final) para verificar que se cortan, ya que si esta condición no se cumple, el método no puede aplicarse pues divergirá.

Las reglas de validación a aplicar en este caso son:

- Validación de la corrección sintáctica de las funciones ingresadas.
- Verificación visual de que ambas funciones se cortan antes de solicitar la resolución usando el método numérico en el intervalo indicado (valor inicial-valor final). Volver a graficar hasta que esa condición se cumpla, modificando el intervalo. Esto ayudará a que el usuario comprenda de forma más integral la manera en que el método opera.
- Se debe verificar que los extremos del intervalo, el valor inicial para comenzar a iterar y el error admitido sean reales.
- Se debe exigir un valor entero y positivo para el número máximo de iteraciones.
- Se debe cumplir que el valor inicial sea menor que el final.
- El error admitido debe ser una cantidad positiva. Cuánto más pequeño, mayor exactitud se logrará en la solución requerida.
- En este método no se exige que el comienzo de las iteraciones se lleve a cabo en un lugar intermedio del intervalo indicado. Sin embargo, esto se recomienda a los efectos de obtener la solución requerida en un número menor de iteraciones.

## CONSTRUCCIÓN DEL PRIMER PROTOTIPO

Siguiendo la metodología para la creación de software educativo de Cataldi et al. (1999), se ha elegido el modelo denominado *prototipo evolutivo*, ya que es importante contar con un primer esbozo de lo que será el programa para satisfacer las demandas del usuario. Además, es necesario saber si se ha interpretado correctamente por parte de los programadores las necesidades de los usuarios. También, este prototipo puede realizar aportes para corregir o considerar aspectos que no se tuvieron en cuenta originalmente.

La evolución de este proyecto de trabajo consiste, básicamente, en ir desarrollando íntegramente cada método numérico antes de continuar con el siguiente. Actualmente, nos encontramos elaborando la parte correspondiente al método de bisección.

### Desarrollo del código

Para el desarrollo del código de este primer prototipo que implementa el método de bisección, se utilizaron las herramientas informáticas antes descritas.

Entre los principales aspectos, podemos mencionar que se utilizó PHP con el editor DEV-PHP 3. Si bien el desarrollo del código requiere las estructuras de programación tradicionales, citaremos algunas características fundamentales:

- Se da una función central a la instrucción **EVAL** de PHP. La misma tiene la función de evaluar código fuente y llevar a cabo las tareas asociadas. Si bien se utiliza mucho en cuestiones relacionadas con código colocado en archivos externos, en nuestro caso, su utilidad reside en la forma en que nos permite evaluar expresiones numéricas. Por ejemplo, el enunciado siguiente:

```
eval("\$b[$x] = $a;");
```

tiene el efecto (fundamental para nuestros propósitos) de almacenar en la posición \$x del arreglo \$b, el resultado de evaluar una expresión numérica cuyo string está almacenado en la variable \$a.

- La biblioteca JPGRAPH proporciona buen manejo de los gráficos, aunque en ocasiones aparecen algunas dificultades por cuestiones no documentadas o algún resultado inesperado.
- El código PHP tiene una sintaxis muy parecida a C y, por lo tanto, a MATLAB

(utilizado en la primera parte del proyecto). Esto nos permitió una rápida traducción del código fuente del método bisección desde MATLAB a PHP.

## Método de bisección

Para el diseño de la interfase del primer prototipo se pensó en que éste cuente con dos partes dentro de una misma ventana. La parte superior cuenta con un espacio para insertar los datos necesarios para aplicar el método de bisección y muestra una gráfica de la función (Figura 2). En la inferior, se ofrece un botón para ejecutar el método, el cual además permite validar los datos ingresados y observar los resultados numéricos y gráficos (Figura 3).

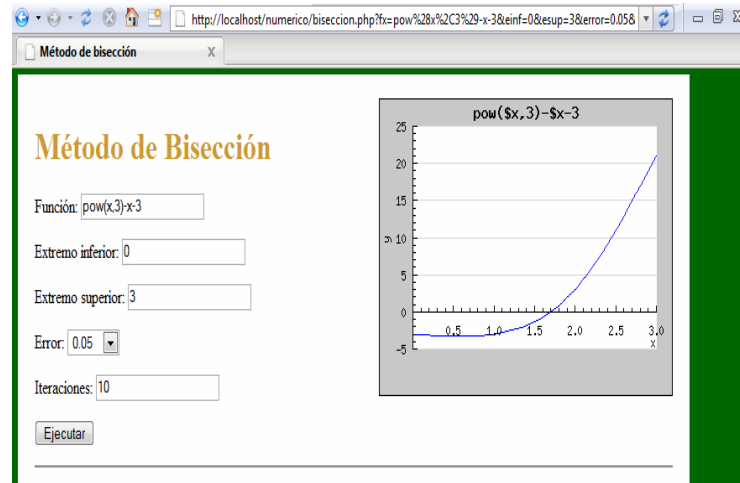


Figura 2. Captura de pantalla del método de bisección (parte superior).

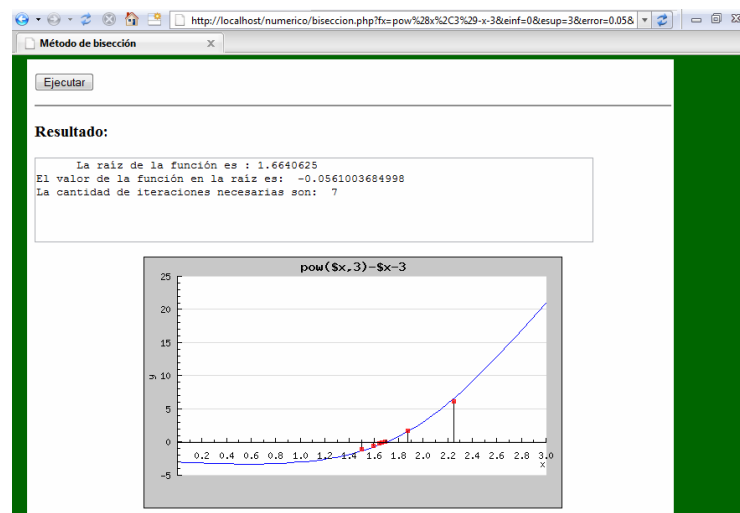


Figura 3. Captura de pantalla del método de bisección (parte inferior).

## RESULTADOS Y CONCLUSIONES

El grado de avance actual en el desarrollo del software educativo, nos permite enunciar algunos resultados preliminares y conclusiones:

- El método de bisección pudo ser desarrollado utilizando únicamente HTML, PHP y JPGRAPH. Pensamos que todo el software podrá hacerse en estas mismas condiciones.
- Inicialmente y teniendo como experiencia el software educativo previamente desarrollado, creíamos necesario el agregado de OCTAVE, una versión libre y

compatible con MATLAB. Sin embargo, los avances producidos nos indicarían que no será necesario utilizarlo y que la funcionalidad que brinda EVAL será suficiente para alcanzar los fines que se persiguen.

- El prototipo es multiplataforma y fue desarrollado por completo con herramientas gratuitas. Si bien es un prototipo, nos parece claro que es factible desarrollar integralmente el software educativo en esos términos y que, finalmente, podrá estar en línea y disponible para cualquier usuario de INTERNET.
- Debido a que estos resultados que hemos obtenido son incipientes, el prototipo aun no ha sido probado con los alumnos. Si bien la versión anterior del software educativo, desarrollado en MATLAB, fue ampliamente utilizado en varias Cátedras con resultados que consideramos satisfactorios, pensamos que el nuevo software propondrá un escenario diferente, y no vamos a tomar como necesariamente válidas aquellas observaciones obtenidas previamente. Consideramos, en cambio, que el software educativo finalizado debe pasar por las mismas evaluaciones que su antecesor.

## BIBLIOGRAFÍA

- Achour, M., Betz, F., Dovgal, A., Lopes, N., Olson, P., Richter, G., Seguy, D. y Vrana, J. 2003-2004. Manual de PHP. Grupo de Documentación de PHP. Recuperado el 16/03/07 de <http://es.php.net/manual/es/>
- Ascheri, M. E. y Pizarro, R. A. 2005. Software para la enseñanza - aprendizaje de algunos métodos numéricos. VII Edumat. EMAT Editora, 987-20239-3-X, 10 pgs. Chivilcoy. Buenos Aires. Argentina.
- Ascheri, M. E. y Pizarro, R. A. 2006. Uso de tecnología en la enseñanza-aprendizaje de temas de Cálculo Numérico. Acta Latinoamericana de Matemática Educativa, CMM-040505-IC7, CLAME, Vol. 19, I970-9971-08-05, pp. 879-885. México. También disponible en <http://www.clame.org.mx>
- Ascheri, M. E., Pizarro, R. A., Astudillo, G. J., García, P. y Culla, M. E. 2007. Relevamiento de software en línea para la enseñanza-aprendizaje de métodos numéricos. Herramientas para su desarrollo. V CIEMAC, Instituto Tecnológico de Costa Rica, Escuela de Matemática, pp. 20-24. Cartago. Costa Rica. También disponible en <http://www.cidse.itcr.ac.cr/ciemac/5toCIEMAC/Ponencias/Relevamientodesoftwareenlinea.pdf>
- Cataldi, Z., Lage, F., Pessacq, R. y García Martínez, R. 1999. Revisión de marcos teóricos educativos para el diseño y uso de programas didácticos. Recuperado el 27/09/07 de <http://www.fi.uba.ar/laboratorios/lisi/c-icie99-revisionde%20marcosteoriciseducativos>
- Chapra, S. y Canale, R. 2007. Applied Numerical Methods with MATLAB. 2ª Ed., Mc Graw - Hill / Interamericana de España, S. A. U. España.
- Font, V., Godino, J. D. and D'Amore, B. 2007. An onto-semiotic approach to representations in mathematics education. For the Learning of Mathematics, 27 (2): 2-7.
- Gómez Sánchez R. 2004. Software libre vs. Software propietario: Programando nuestro futuro. HAOL, 2, 125-140.
- Hitt, F. 2003. Una reflexión sobre la construcción de conceptos matemáticos en ambientes con tecnología. Boletín de la Asociación Matemática Venezolana, (X), 2.
- Joyner, D. 2007. Open Source Mathematical Software. Notices of the AMS. Recuperado el 05/03/08 de <http://www.ams.org/notices/200710/index.html>
- Lerdof, R. y Tatroe K. 2002. Programming PHP. O'Reilly. EEUU.
- Maita Guedez, M. 2005. El aprendizaje de funciones reales con el uso de un software educativo: una experiencia didáctica con estudiantes de educación de la ULA-Táchira. Información de detalles Acción Pedagógica (14), 1, 38-49.

- Marquès, P. 1999. El software educativo. Universidad de Barcelona. España. Recuperado en noviembre de 2007 de URL:<http://www.doe.d5.ub.es>
- Nakamura, S. 1997. Análisis Numérico y visualización gráfica con MATLAB. Pearson Educación. México.
- Pérez Sanz, A. 2006. Matemáticas en las Aulas de Secundaria. LA GACETA DE LA RSME, (9), 2, 521–544.
- Persson, J. 2003. JpGraph and DDDA. Aditus Consulting, S-128 46, Emagatan 16. Stockholm. Sweden.
- Pizarro, R. A. y Ascheri, M. E. 2007. Diseño e implementación de un software educativo en Cálculo Numérico. II Congreso de Tecnología en Educación y Educación en Tecnología, TE&ET 07, CD-ROM, 9 pgs. La Plata. Buenos Aires. Argentina.
- Raggett, D. 2005. Getting started with HTML. W3C. Recuperado el 11/08/07 de <http://www.w3.org/MarkUp/Guide/>
- W3C. 1999. HTML 4.01 Specification. Recuperado el 03/08/07 de <http://www.w3.org/TR/1999/REC-html401-19991224/>