

## Sistema de información para la resolución y graficación de funciones por el método de bisección y falsa posición”

Jhon Kevint Rodríguez Baquero<sup>1</sup>, Julián David Vera Guzmán<sup>1</sup>

1: Ingeniería de sistemas, Universidad Piloto de Colombia. Girardot-Cundinamarca, Colombia

\* julian-vera@upc.edu.co

### Resumen

El presente artículo tiene como tema principal el desarrollo de un software que permita resolver cálculos por medio de dos métodos numéricos: método de bisección y método de la falsa posición, para que de esta manera se pueda lograr un resultado aproximado demasiado preciso en menos de un segundo.

Para desarrollar dicho software, se hizo uso del entorno de desarrollo Netbeans en el que, mediante el lenguaje de programación Java y las opciones de interfaz gráfica que este entorno trae, se logró diseñar y programar las soluciones de los métodos mencionados en el párrafo anterior, teniendo como datos de entrada una función y valores iniciales y finales que el usuario digita y como salida una gráfica de dicha función y una tabla con los resultados de cada iteración utilizada para llegar al valor que el usuario necesite.

Los resultados que se obtienen al hacer uso de

este programa son demasiado precisos y exactos, además que permite al usuario digitar gran variedad de funciones y ver la solución que se ofrece por ambos métodos sin necesidad de ir a otro programa.

**Palabras clave:**  
**Software, métodos numéricos, ingeniería, Java, programación, bisección, falsa posición.**

### Abstract

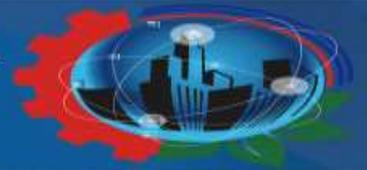
The main topic of this article is the development of a software that allows solving calculations by means of two numerical methods: bisection method and false position method, so that in this way an approximate result that is too precise can be achieved in less than a second.

To develop this software, the Netbeans development environment was used in which, through the Java programming language and the graphical interface options that this

environment brings, it was possible to design and program the solutions of the methods mentioned in the previous paragraph, having as input data a function and initial and final values that the user types and as output a graph of said function and a table with the results of each iteration used to arrive at the value that the user needs.

The results obtained when using this program are too precise and exact, and it also allows the user to enter a wide variety of functions and see the solution offered by both methods without having to go to another program.

**Keywords:** **Software, numerical methods, engineering, Java, programming, bisection, false position.**



## 1 PROBLEMA

El desarrollo de una solución a un método numérico cualquiera es una tarea que consume demasiado tiempo, si por ejemplo se desarrolla esta acción de forma manual en una hoja de papel, podría llevar demasiadas horas en trabajos de gran escala, debido a la gran cantidad de iteraciones que se necesitan realizar, además, puede ser confuso realizar cálculos a mano y llevar a una equivocación demasiado grande si no se tienen en cuenta la cantidad de decimales suficientes como para dar un resultado, debido a que el error se va acumulando mediante se va avanzando con el cálculo.

Si bien esto se puede realizar de forma manual en una hoja de cálculo que después permita automatizarla, de igual forma sería un gasto de tiempo en el diseño de esta misma porque habría que hacer una para cada método y puede llegar a tenerse dispersa la información.

## 2 INTRODUCCIÓN

Desde el inicio de los tiempos las personas han inventado sus propios métodos para diversas ocasiones, sea la temática que se esté manejando. Los métodos numéricos se dice que son aquellos procedimientos que hayan la solución, algunas veces, aproximada de un problema matemático utilizando solamente cálculos aritméticos y lógicos; cálculo de funciones, operaciones aritméticas elementales, entre otras. (Vasquez, 2009)

Además, a la hora de fenómenos complejos en la ciencia, física y en la ingeniería requieren de estos procesos para así facilitar lo que se necesite encontrar, en parte, los métodos numéricos colaboran en “adivinar” situaciones que sucederán en el futuro con sus cálculos, gestionándolos en base a operaciones que se aprenden estando desde la primaria; como la suma, la resta, multiplicación y división, sin dejar a un lado aquellas que se ven en secundaria.

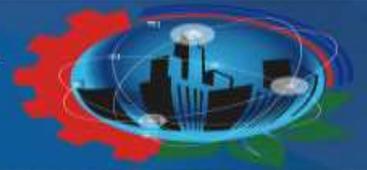
Teniendo en cuenta lo anterior mencionado, para hacer aún más versátil este tema, en este documento se estará tratando unos de los métodos más

usados y convencionales; el método de la bisección y el método de la falsa posición. También se diseñará un software con el que se calculará de una forma más rápida los métodos anteriormente nombrados, por lo que se usará un programa llamado Java, el cual se tratará más a detalle con paso se vaya avanzando en el documento.

## 3 METODOLOGÍA

Para comenzar, es de gran importancia primero conocer el tema del proyecto sobre el cual se está manejando el problema, para posteriormente darle una solución, como primer paso, es necesario profundizar primero en el método de Bisección y el método de la falsa posición, teniendo en cuenta fuentes externas de información apropiadas, como libros de personas con experiencia en el tema, videos y como fuente esencial se tomará el libro del curso, con el objetivo de comprender lo máximo posible las operaciones y procedimientos de ambos métodos.

De acuerdo con lo anterior dicho, se puede proceder ahora con la



definición de ambos métodos, comenzando primero con el de Bisección.

“Es el método más elemental y antiguo para determinar las raíces de una ecuación. Está basado directamente en el teorema de Bolzano explicado con anterioridad. Consiste en partir de un intervalo  $[x_0, x_1]$  tal que  $f(x_0) f(x_1) < 0$ , por lo que sabemos que existe, al menos, una raíz real. A partir de este punto se va reduciendo el intervalo sucesivamente hasta hacerlo tan pequeño como exija la precisión que hayamos decidido emplear.” (Villanueva, 1998)

Según lo anterior, el método de bisección haya la raíz de una función haciendo una serie de iteraciones en donde el resultado de la raíz se va reduciendo a tal punto de dar una respuesta aproximada en la mayoría de veces.

Dando paso a la falsa posición, Díaz (1998) relata que: “El método de la falsa posición pretende conjugar la seguridad del método de la bisección con la rapidez del método de la secante. Este método, como en el método de la bisección, parte de dos puntos que

rodean a la raíz  $f(x) = 0$ , es decir, dos puntos  $x_0$  y  $x_1$  tales que  $f(x_0) f(x_1) < 0$ .”

Sin embargo, este método tiene una convergencia demasiado lenta a la solución, esto quiere decir que no es un método muy eficaz para encontrar la solución de manera rápida.

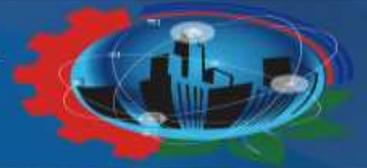
No obstante, para dejar un lado todos los procesos manuales al manejar alguno de estos métodos, a continuación se comentará las herramientas que se utilizaron para el diseño del software anteriormente mencionado. Para empezar, se hablará un poco sobre el lenguaje de programación que se utilizó para el debido desarrollo que se empleó en el programa, el cual es denominado como Java, teniendo en cuenta que existen numerosos lenguajes, se optó por manejar este ya que es el que más conocimiento se tiene, es el que se ha venido trabajando desde principios de la formación profesional.

“La tecnología Java se usa para desarrollar aplicaciones para un amplio alcance de entornos, desde dispositivos del consumidor hasta sistemas empresariales heterogéneos. El

lenguaje Java es un derivado del lenguaje C, por lo que sus reglas de sintaxis se parecen mucho a C: por ejemplo, los bloques de códigos se modularizan en métodos y se delimitan con llaves (`{ y }`) y las variables se declaran antes de que se usen.” (Perry, 2012)

Además, se tuvo que adjuntar unas librerías externas al programa para así poder que el software funcione correctamente a lo que se esperaba. A continuación, se muestran varios conceptos importantes para poder entender un poco más el funcionamiento y la metodología para la creación del programa:

- “Una librería Java se puede entender como un conjunto de clases que facilitan operaciones y tareas ofreciendo al programador funcionalidad ya implementada y lista para ser usada través de una Interfaz de Programación de Aplicaciones, comúnmente abreviada como API (por el



anglicismo Application Programming Interface). Las librerías Java nos permiten reutilizar código, es decir, podemos hacer uso de las clases, métodos y atributos que componen la librería evitando así tener que implementar nosotros mismos esas funcionalidades.” (Durante, 2016)

- Las librerías JMathPlot y la JME\_6\_0 hacen un papel muy importante en el programa; la JME\_6\_0 sirve para evaluar las funciones que se ingresen, ya que

se sabe que una función lleva la letra “x”, y por sí solo el lenguaje no recibe este proceso por sí solo, por ende se tiene que agregar, y la JMathPlot es aquella que recibe la función y gráfica dicha función.

- También se utilizó la librería Swing, este se añadió para crear la clase JFrame, la cual nos sirve para crear la ventana que sale a la hora de correr el programa. En ese mismo sentido se utilizaron más clases para seguir con la

elaboración; como lo es el JPanel, JLabel, JText, entre otras.

- Lenguaje de programación, un lenguaje de programación es el cual “mediante una serie de instrucciones, le permite a un programador escribir un conjunto de órdenes, acciones consecutivas, datos y algoritmos para, de esa forma, crear programas que controlen el comportamiento físico y lógico de una máquina” (Redactor Rock Content, 2019).

## 4 AVANCES

Como avances del proyecto, se tiene la interfaz gráfica totalmente funcional para que el usuario digite la función que desea analizar y resolverla por el método de la falsa posición, faltando así el otro método que es el de la bisección.

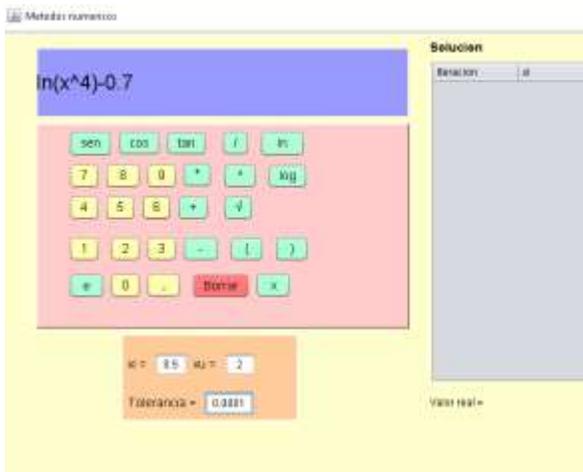
A continuación, se mostrarán distintas capturas del programa funcionando y posteriormente ideas de diseño que se tienen para implementar el método de la bisección





**Figura 1. Interfaz de la calculadora para el método de falsa posición**

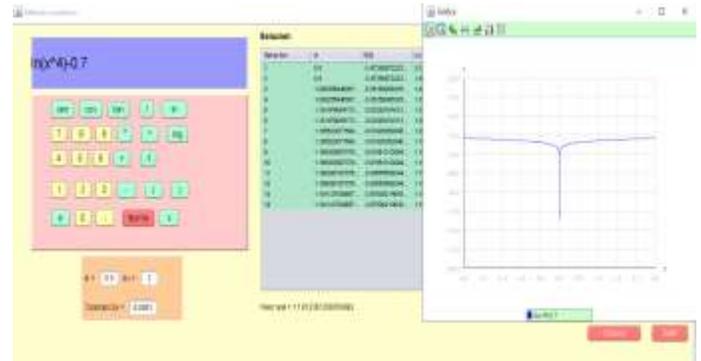
En la figura 1, se ve la interfaz inicial del programa, sin haber digitado ninguna función, con los valores por defecto en 0 y la tabla inicializando sus columnas.



**Figura 2. Ingreso de valores, la función y el valor inicial y final.**

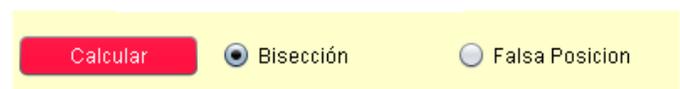
En la figura 2 se puede ver que el usuario puede digitar la función a evaluar y el valor inicial, final y de la tolerancia. Como dato relevante, esta interfaz se hizo de manera que solo se pueda digitar la función con los botones en pantalla y no con teclado, de esta manera, hacer que

el programa reconozca los caracteres de una forma universal y así no dar errores.



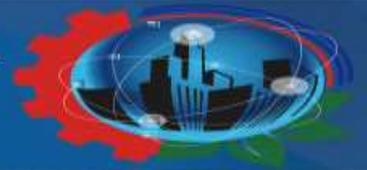
**Figura 3. Solución y gráfica de la función.**

En la figura 3 se puede observar que, al darle a botón de calcular, este ejecuta el código que lleva a la solución del problema, arrojando como resultado la gráfica correspondiente de la función y la tabla de los resultados con sus iteraciones.



**Figura 4. Diseño de la implementación del método de bisección**

En la figura 4 se puede evidenciar el avance del diseño para la implementación del método de la bisección en la misma interfaz mediante un botón tipo JRadioButton que sirve para que el



usuario tenga la capacidad de escoger el método y así al darle al botón de calcular lo resuelva por el método seleccionado.

#### 4 CONCLUSIONES Y RECOMENDACIONES

- Para concluir, se tiene que este es un proyecto que sigue en desarrollo pero que hasta el momento ha sido viable, debido a que cumple con lo que se planteó en un inicio y funciona arrojando resultados demasiado exactor.
- Este proyecto sirve para todos los estudiantes de ingeniería de sistemas o que tengan que realizar estudios con métodos numéricos, facilitando así su trabajo.

#### 5 AGRADECIMIENTOS

Agradecemos al programa de ingeniería de sistemas de la universidad piloto por su apoyo y colaboración para el proceso de formación profesional, de igual manera, agradecemos al semillero de software libre y web-in que es el espacio a través del cual el programa nos apoya y orienta para la participación en eventos de carácter nacional e internacional.

#### 6 REFERENCIAS

##### **Libros:**

- [1]. CANALE, S. CHAPRA. Métodos numéricos para ingenieros - Quinta edición. McGraw-Hill companies, Ciudad De México, 2007.
- [2]. CANALE, S. CHAPRA. Métodos Numéricos para ingenieros - 6ta Edición. McGraw-Hill companies, Ciudad De México, 2008.

- [3]. NAVARRETE – Desarrollo de dos librerías Java: JavaCheckCode y JavaCheckStyle. Valencia, 2016.

##### **Referencias de páginas web:**

- [4]. Ana Navarrete Durante. Desarrollo de dos librerías Java: JavaCheckCode y JavaCheckStyle. Obtenido de: <https://riunet.upv.es/bitstream/handle/10251/70839/NAVARRETE%20-%20Desarrollo%20de%20dos%20librer%C3%ADas%20Java%3A%20JavaCheckCode%20y%20JavaCheckStyle.pdf?sequence=1#:~:text=Una%20librer%C3%ADa%20Java%20se%20puede,el%20anglicismo%20Application%20Programming%20Interface>). (Fecha de recuperación: noviembre de 2021).
- [5]. Steven Perry. Conceptos Básicos del lenguaje Java. Obtenido de: <https://developer.ibm.com/es/tutorials/j-introtojava1/> (Fecha de recuperación: noviembre de 2021).
- [6]. Ricardo Vasquez. METODOS NUMERICOS PARA INGENIERIA. Obtenido de: <https://disi.unal.edu.co/~lctorress/MetNum/LiMetNu2.pdf> (Fecha de recuperación: noviembre de 2021).
- [7]. Wladimiro Diaz Villanueva. Método de la bisección – Método de la falsa posición. Obtenido de: <https://www.uv.es/~diaz/mn/node18.html> (Fecha de recuperación: noviembre de 2021).