Modelado Orientado a Objetos para evaluar Métodos Numéricos utilizando Interfaces Visuales.

F. Meneses, W. Fuertes
Departamento de Ciencias de la Computación, Escuela Politécnica del Ejército, Sangolquí-Ecuador
{fmenesesb, wfuertesd}@espe.edu.ec

RESUMEN: La programación orientada a objetos (POO) desde hace varios años se ha convertido en uno de los motores clave para implementar cálculos de ingeniería en sus diversas aplicaciones. El presente trabajo de investigación tiene como propósito evaluar varios métodos numéricos, diseñando un modelo único, implementando los algoritmos orientados a objetos de estos métodos y evaluando sus resultados. Para validar este modelo, se ha implementado una interfaz visual (interfaz gráfica de usuario, GUI) incorporando técnicas orientadas a objetos. Los resultados experimentales permiten comparar la eficiencia computacional de los métodos numéricos para el cálculo de raíces reales de funciones trascendentes y polinómicas.

ABSTRACT: The object-oriented programming (OOP) for several years has become one of the key drivers for implementing engineering calculations in its various applications. This research work proposes to evaluate various numerical methods, designing a unique model, implementing the object-oriented algorithms of these methods and evaluate their results. To validate this model, we have implemented a visual interface (graphical user interface, GUI) incorporating object-oriented techniques. The experimental results allow us to compare the computational efficiency of numerical methods for calculating real roots of polynomial and transcendental functions.

1. INTRODUCCIÓN

La programación orientada a objetos (POO) desde hace varios años se ha convertido en uno de los motores clave para implementar cálculos de ingeniería en sus diversas aplicaciones. El desarrollo de la POO es un enfoque basado en la creación de modelos del mundo real y la implementación de programas informáticos basados en esos modelos.

Debido a la factibilidad de resolución de problemas de complejidad matemática que tiene la POO, se está utilizando en aplicaciones de cálculo numérico como son: interpolación de funciones, derivación e integración numérica, sistemas de ecuaciones lineales y no lineales, regresión lineal, resolución de ecuaciones diferenciales, etc. En concreto, la POO ha permitido la optimización de esfuerzos en la implementación de los algoritmos, su reutilización, e inclusive su aplicación en el Web con ligeros cambios.

Sin embargo, en la actualidad, los métodos numéricos específicamente, han sido implementados por lo general utilizando programación Estructurada en una interfaz a modo de texto. Del mismo modo, al tratar un problema completo (regresión lineal por ejemplo), no se considera la integración de los métodos numéricos. Así mismo, los métodos numéricos se tratan de una manera aislada y no se establece la relación que existe entre ellos provocando errores en los cálculos y duplicación de código.

Ante estos problemas, la presente investigación tiene como propósito evaluar varios métodos numéricos, diseñando un modelo único, implementando los algoritmos orientados a objetos de estos métodos y evaluando sus resultados. Para llevarlo a cabo, se ha realizado un análisis de los métodos numéricos que permiten calcular raíces reales de ecuaciones de la forma *F(X)=0*, donde *F(X)* es una función con una variable independiente trascendente o polinómica. Una vez identificados, se ha diseñado un modelo (diseño de un diagrama de clases) que integre en el mismo operaciones para lectura-escritura en archivos; manipulación de componentes visuales, los métodos numéricos para el cálculo de raíces reales, métodos para evaluar polinomios y sus respectivas derivadas. Para validar este modelo, se ha implementado una interfaz visual (aplicativo) incorporando técnicas orientadas a objetos.

En consecuencia, las contribuciones de este estudio son: *i)* obtener una comparación de la eficiencia de la aplicación de diversos métodos numéricos; *ii)* el diseño de un modelo único de interfaz visual orientada a objetos; *iii)* la implementación de una librería común de clases, que permite la reutilización de código, tanto para funciones trascendentes como para polinómicas.

El resto del artículo ha sido organizado como sigue: La sección 2 describe los fundamentos teóricos de los métodos numéricos que han sido modelados. La sección 3 detalla el diseño, e implementación de algoritmos. En la sección 4 se muestran y discuten los resultados experimentales. En la sección 5, se analizan algunos trabajos relacionados. Finalmente, en la sección 6, se presentan las conclusiones y líneas de trabajo futuro sobre la base de los resultados obtenidos.

2. METODOS NUMERICOS

Existen varios métodos numéricos para calcular raíces reales [1]. Aquellos que han sido considerados en este artículo, ya que son los que mas se utilizan en centros de investigación fueron: Bisección, Aproximaciones sucesivas, Aproximaciones sucesivas modificado, Régula-Falsi, Secante y el método de Newton-Raphson.

La implementación de los algoritmos orientados a objetos de estos métodos necesita de las siguientes entradas: *F(X), XI, XD, PRECISION*, donde *F(X)* es la función continua; *XI* y *XD* son las abscisas izquierda y derecha, respectivamente, entre las que puede estar por lo menos una raíz real; *PRECISION* es el margen de error que se produce al efectuar el cálculo de una raíz. A continuación se describen brevemente algunos de ellos y de una manera detallada el método Newton-Raphson, por ser el de mayor eficiencia:

2.1 Método de la Bisección:

Consiste en acortar el intervalo de las abscisas, calculando su valor medio Xm, luego se calcula F(Xm) y se aproxima a la abscisa cuya ordenada tiene el mismo signo de F(Xm).

2.2 Método de aproximaciones sucesivas:

Para este método, se transforma la ecuación F(X)=0, en Y= fm(X) = X, a continuación se aplican las fórmulas recurrentes:

X1=fm(X0); X2=fm(X1); X3 = fm(X2); ...; Xi+1 = fm(Xi), tal que se cumpla:

$| Xi+1 - Xi | <= PRECISION$

Este método tiene el inconveniente de que no siempre se converge a la raíz; solo existe convergencia para cuando $|fm'(X)| \leq 1$.

2.3 Método de aproximaciones sucesivas modificado:

Este método, aplica el mismo principio de su antecesor, pero introduce un factor de corrección (β), que permite una convergencia mas rápida (ver ecuaciones (1)-(7)).

$\Delta X = Xi - Xi\text{-}1 = fm(Xi\text{-}1) - Xi\text{-}1$ *(1);*

$Xi = Xi\text{-}1 + \Delta X$ *(2);*

Aplicando el factor de corrección (2), se convierte en:

$Xi = Xi\text{-}1 + \beta * \Delta X$ *(3);*

Reemplazando el miembro de la derecha de (1) en (3), se obtiene la fórmula de recurrencia de este método:

$Xi = Xi\text{-}1 + \beta * (fm(Xi\text{-}1) - Xi\text{-}1)$ *(4);*

$Tg(\theta) = (\beta - 1) * \Delta X / (\beta * \Delta X) = (\beta - 1) / \beta$ *(5);*

Despejando de (5), el valor β: $\beta = 1 / (1 - Tg(\theta))$ *(6);*

$Tg(\theta) = (fm(Xi) - fm(Xi\text{-}1)) / (Xi - Xi\text{-}1)$ *(7).*

Con estos elementos, se puede plantear el algoritmo correspondiente.

2.4 Método de Régula-Falsi:

Consiste en trazar una secante que pasa por los puntos A(XI, F(XI)) y B(XD, F(XD)); luego, mediante la ecuación de la línea recta, se calcula la abscisa en el punto C(Xi-2, 0), a continuación se acorta el intervalo, dependiendo del lado donde se encuentre la ordenada. Seguidamente se deduce la fórmula de recurrencia:

La ecuación de recta es:

$(F(XD) - F(XI)) / (XD - XI) = (F(Xi\text{-}2) - F(XI)) / (Xi\text{-}2 - XI)$ *(8);*

además $F(Xi\text{-}2) = 0$ *(9);*

Reemplazando (9) en (8) y despejando de (8) Xi-2:

$Xi\text{-}2 = XI - F(XI) * (XD - XI) / (F(XD) - F(XI))$ *(10);*

Reemplazando Xi-2 por Xm, se obtiene la fórmula de recurrencia:

$Xm = XI - F(XI) * (XD - XI) / (F(XD) - F(XI))$ *(11).*

2.5 Método de la Secante:

Este método utiliza la fórmula de recurrencia del de Régula-Falsi, calculándose valores consecutivos en las abscisas de uno de los extremos y manteniendo fijo el otro extremo. Aunque este método reduce las comparaciones es ineficiente.

2.6 Método de Newton-Raphson:

Este método se basa en el cálculo de la derivada (Ver Figura 1).

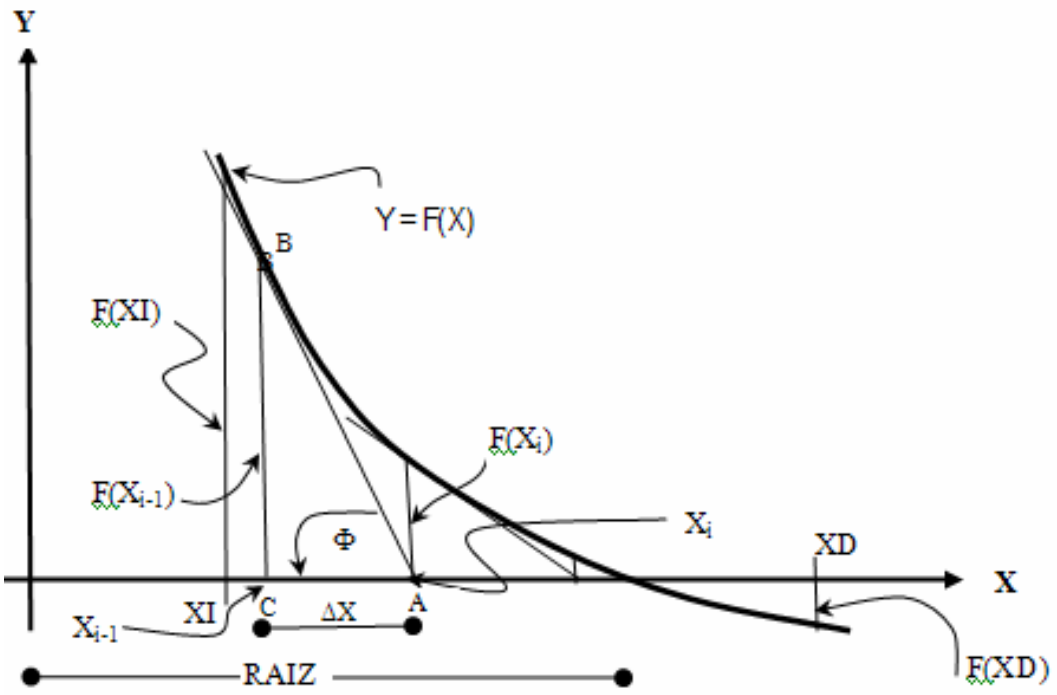

Fig. 1. Representación de la función Y=F(X) y su derivada. Esta figura muestra los elementos requeridos para calcular raíces reales por el método de Newton-Raphson.

De acuerdo a la Figura 1: ΔX = Xi - Xi-1 *(12);*
entonces: Xi = Xi-1 + ΔX *(13);*

En el triángulo ABC: Tg(Φ) = - F(Xi-1) / ΔX = F'(Xi-1) *(14);*

Despejando de (14) ΔX: ΔX = - F(Xi-1) / F'(Xi-1) *(15);*

Sustituyendo (15) en (13): Xi = Xi-1 - F(Xi-1) / F'(Xi-1) *(16); la cual es la fórmula de recurrencia para calcular raíces reales por este método.*

El algoritmo para el método de Newton-Raphson es el siguiente:

(a) Xm = (XI + XD) / 2

(b) Ym = F(Xm)

(c) YP = F'(Xm)

(d) Xm = Xm - Ym / YP

(e) REPETIR DESDE EL PASO (b) MIENTRAS |Ym| > PRECISION

(f) ESCRIBIR RESULTADOS.

Este es uno de los métodos más simples porque involucra pocas operaciones, y a la vez el más eficiente como se demostrará en las secciones siguientes.

Para acceder al análisis de funciones polinómicas, favor acceder al URL: http://dcc.espe.edu.ec/~wfuertesd.

3. DISEÑO E IMPLEMENTACIÓN

3.1 Diseño del Modelo:

Se ha considerado el estándar UML (Lenguaje de modelado unificado) en razón de que el objetivo de UML es que sus modelos sean independientes del lenguaje de implementación, de tal forma que los diseños se puedan implementar en cualquier lenguaje que soporte las posibilidades de UML [2], de manera particular los lenguajes orientados a objetos, como es el caso del lenguaje de esta interfaz visual (Aplicación). La Figura 2 muestra el diagrama de clases de las operaciones para lectura-escritura en archivos, cuadros de diálogo; manipulación de componentes visuales, los métodos numéricos para el cálculo de raíces reales, métodos para evaluar polinomios y sus respectivas derivadas, así como la interrelación entre ellos.

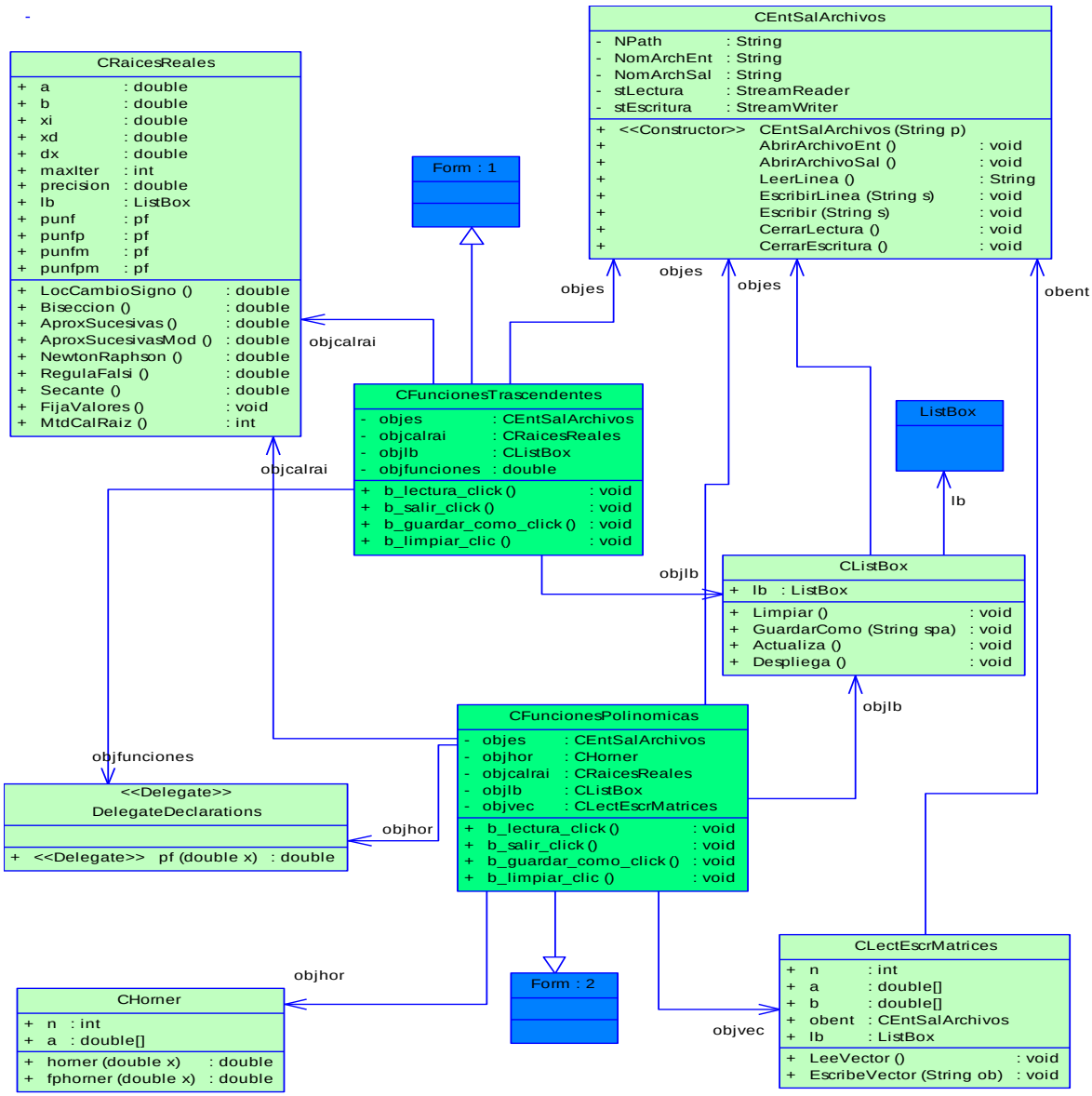

Fig. 2. Diagrama de Clases de la interfaz visual (Aplicación) y sus interrelaciones.

A continuación se describe brevemente cada una de las clases:

- **CEntSalArchivos**.- Clase para entrada-salida desde archivos, incluye presentación de cuadros de diálogo para seleccionar archivos, para abrirlos, para lectura o guardar como;
- **CListBox**.- Clase que añade a un listbox la funcionalidad de borrar todos sus elementos y para guardar en un archivo la información de sus elementos;
- **CRaicesReales**.- Clase que contiene los métodos numéricos para calcular las raices reales de ecuaciones de la forma $F(x) = 0$;
- **CLectEscrMatrices**.- Clase que contiene métodos para lectura-escritura de vectores o matrices sobre archivos;
- **CHorner**.- Clase que contiene funciones para calcular el valor o la derivada de un polinomio, dados los coeficientes y el valor de la variable independiente;
- **DelegateDeclarations**.- Clase que permite incorporar delegados (punteros a función);
- **CFuncionesTrascendentes**.- Clase que contiene las funciones trascendentes y los objetos que permiten probar el proceso de calculo de raíces reales a través de los métodos de las otras clases;
- **CFuncionesPolinomicas**.- Clase que contiene los objetos que permiten probar el proceso de cálculo de raíces reales a través de los métodos de las otras clases.

3.2 Implementación:

La implementación del modelo se la ha realizado en Microsoft Visual C#.NET, versión 2005, en razón de su facilidad para implementar interfaces visuales y debido a que se basa en C++, el cual es ampliamente utilizado. Para cualquier prueba de concepto, toda la información (programas fuente, clases, miembros de las clases, resultados, etc.) se encuentra disponible en el URL: http://dcc.espe.edu.ec/~wfuertesd. Para la evaluación del modelo de clases, se implementaron algoritmos separados en dos fases:

i.) **Fase 1:** Localización del cambio de signo de la función F(X), que consiste en tomar un intervalo cerrado de valores para las abscisas y para dos valores consecutivos, se calculan las respectivas ordenadas, luego se establece el producto entre dichos valores, el mismo que si es menor o igual a cero entonces significa que se ha localizado la raíz, de lo contrario, se toma el siguiente par de valores consecutivos en las ordenadas y así sucesivamente hasta alcanzar el límite superior del intervalo.

ii.) **Fase 2:** Afinamiento de los valores de las abscisas (XI y XD), entre las que se encuentra la raíz real, la cual se ha basado en la aplicación de criterios para encontrar una raíz en base a una precisión requerida.

Cabe señalar que para el cálculo de raíces reales, la una a través de funciones trascendentes y la otra mediante funciones poli nómicas, se comparten una librería común de clases. Esto debido a la aplicación de delegados (punteros a función) los cuales permiten reutilizar el código en los Aplicativos tanto para funciones trascendentes como para las polinómicas.

La Fig. 3 muestra el diagrama de secuencia del cliente, que representa la interacción entre el usuario y la interfaz grafica principal, con el objetivo de probar el modelo. En primer lugar, el usuario solicita una operación de gestión (por ejemplo, el despliegue) (1). Entonces el usuario registra el rango y el intervalo donde se encuentran las raíces reales (2). El formulario captura los datos registrados (3). El formulario fija valores iniciales en la clase *CRaicesReales* (4). Para cada intervalo del rango ingresado se localiza el subintervalo donde existe una raíz real (5). Para cada método de cálculo se invoca al respectivo método de *CRaicesReales* (6). Una vez procesada la raíz se actualiza *CListBox* (7). Finalizados los intervalos del rango se despliega en el formulario los elementos de *CListBox* (8). A continuación se despliega los resultados en la interfaz grafica de usuario (9).

El usuario requiere grabar resultados en un archivo (10). La interfaz activa el mensaje *GuardarComo* de *CListBox* (11). El método citado despliega al usuario un cuadro de dialogo para que seleccione el nombre del archivo donde desea guardar los resultados (12). El usuario selecciona el nombre del archivo y ordena guardar resultados (13). *CListBox* activa el mensaje para guardar resultados y reporta a la interfaz (14). El usuario obtiene los resultados guardados en el archivo (15).

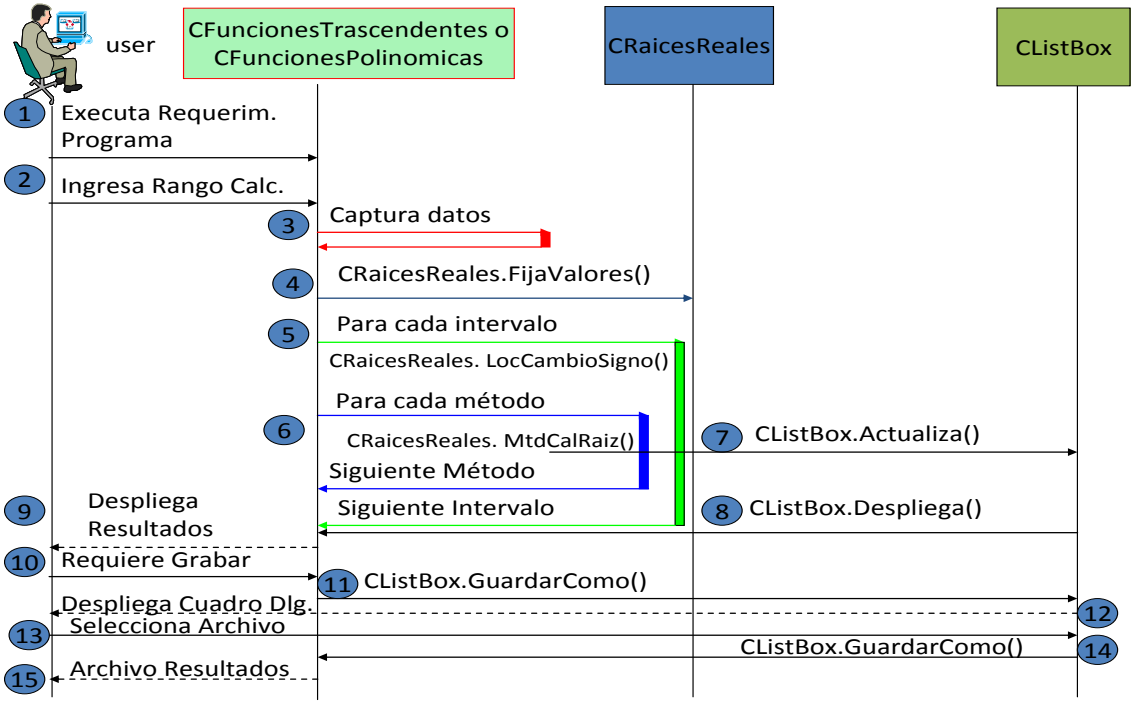

Fig. 3. Diagrama de Secuencia de la interfaz visual (aplicación)

4. EVALUACIÓN DE RESULTADOS

Con el fin contrastar el modelo y evaluar el Aplicativo, se realizaron varias pruebas de validación utilizando diversas ecuaciones. Sin embargo para efectuar el análisis comparativo de los resultados, se empleó una misma función, la cual se la expresa mediante operaciones para la función trascendente, y vector de coeficientes para la función polinómica. Así por ejemplo, al calcular de raíces reales de la función trascendente ***F(x) = x3 – 7x + 4***, para la Fase 1, se generó la Tabla 1:

Tabla 1. Resultados de la aplicación del algoritmo para la localización de cambios de signo.

XI	XD	YI = F(XI)	YD = F(XD)	P = YI*YD
-4.00	-4.00		-32.00	
-4.00	-3.00	-32.00	-2.00	(+)
-3.00	-2.00	-2.00	10.00	(-)
-2.00	-1.00	10.00	10.00	(+)
-1.00	0.00	10.00	4.00	(+)
0.00	1.00	4.00	-2.00	(-)
1.00	2.00	-2.00	-2.00	(+)
2.00	3.00	-2.00	10.00	(-)
3.00	4.00	10.00	40.00	(+)
4.00	5.00	40.00	94.00	(+)

Por consiguiente, existen raíces en los intervalos *[-3, -2], [0, 1] y [2, 3]*.

Para la **Fase 2**, los resultados utilizando dos métodos diferentes fueron los siguientes (ver Tablas 2 y 3):

Tabla 2. Cálculo de raíces reales utilizando el método de Aproximaciones Sucesivas.

ITERACION	Xm	Ym = fm(Xm)	YP = fm'(Ym)	|Ym - Xm|
0		-2.500000		
1	-2.500000	-2.780649	0.854063	0.280649
2	-2.780649	-2.862886	0.840374	0.082237
3	-2.862886	-2.886109	0.836607	0.023223
4	-2.886109	-2.892600	0.835561	0.006491
5	-2.892600	-2.894408	0.835271	0.001809
6	-2.894408	-2.894912	0.835190	0.000504
7	-2.894912	-2.895052	0.835167	0.000140
8	-2.895052	-2.895091	0.835161	0.000039
9	-2.895091	-2.895102	0.835159	0.000011
10	-2.895102	-2.895105	0.835159	0.000003
11	-2.895105	-2.895106	0.835159	0.000001

Tabla 3. Cálculo de raíces reales utilizando el método de Newton-Raphson.

ITERACION	Xm	Ym = F(Xm)	YP = F'(Xm)
0	-2.500000		
1	-3	5.875000	11.750000
2	-2.9	-2.000000	20.000000
3	-2.895118	-0.089000	18.230000
4	-2.895107	-0.000207	18.145124
5	-2.895107	0.000000	18.144925

Al comparar las Tablas 2 y 3, se puede observar que el método de Aproximaciones Sucesivas (Tabla 2) debe desarrollar un mayor esfuerzo computacional, para determinar una raíz, además de que no siempre se producirá la convergencia. En cambio mediante el método de Newton-Raphson (Tabla 3), se determina la raíz con un menor número de iteraciones. En el caso de existir interés por revisar la eficiencia de los otros métodos, favor acceder al URL: http://dcc.espe.edu.ec/~wfuertesd.

En relación a la comparación de los métodos numéricos, la Figura 4 muestra los resultados obtenidos. Como se puede observar, Newton-Raphson es más eficiente.

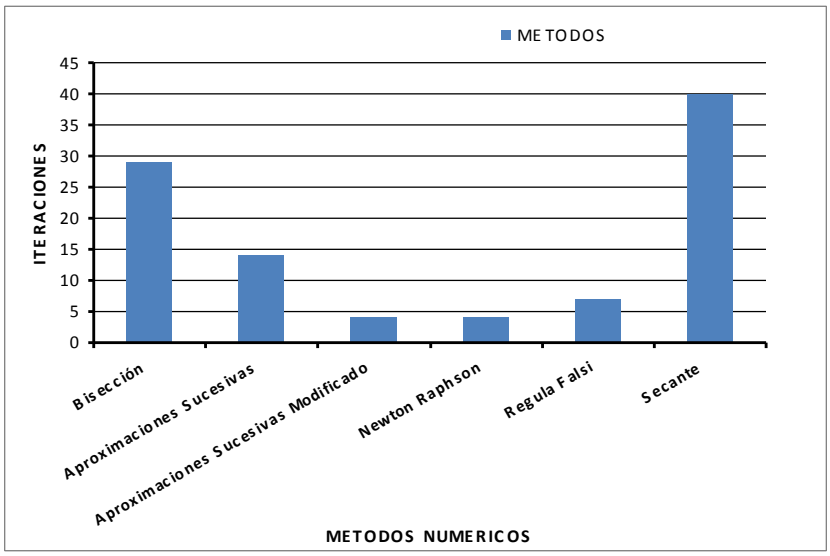

Fig. 4. Diagrama de barras de los Métodos Numéricos para calcular la raíz que se encuentra entre *-3 y -2* de la ecuación $X^3 - 7x + 4 = 0$.

En relación a la interfaz de resultados de la Aplicación (ver Figuras 5 y 6), al ser visual, muestra cuadros de edición para ingresar el rango dónde se requiere encontrar raíces con intervalo de variación; la lista (control visual) dónde se presentan los resultados del proceso y que corresponde a las raíces reales obtenidas en la ejecución de los métodos numéricos descritos en la sección 2. Además brinda la posibilidad de que estos resultados puedan almacenarse en un archivo de salida.

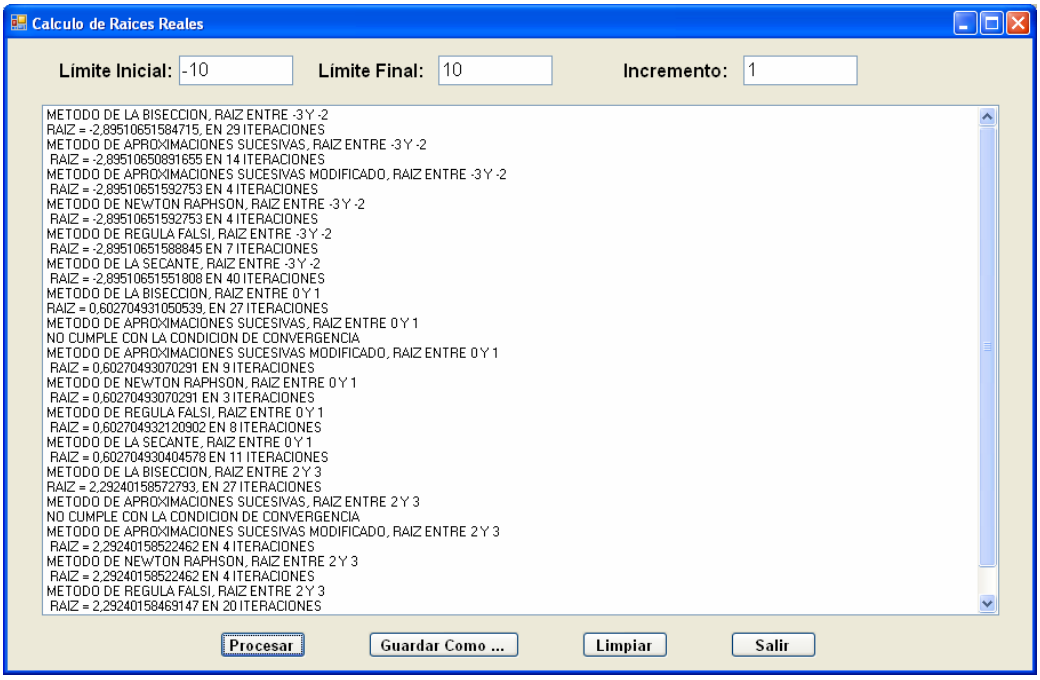

Fig. 5. Interfaz gráfica para cálculo de raíces reales de funciones trascendentes.

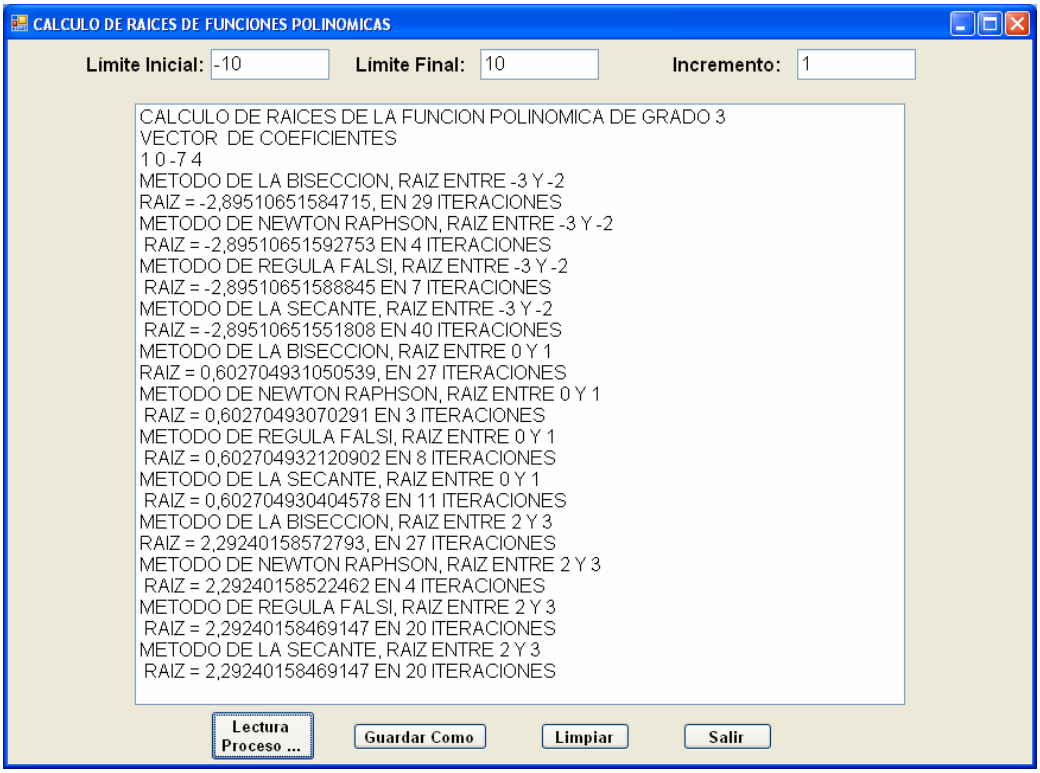

Fig. 6. Interfaz gráfica para cálculo de raíces reales de funciones polinómicas.

5. TRABAJOS RELACIONADOS

El trabajo propuesto por Corral y Fernández en [3], presenta un análisis de la evolución de los lenguajes de POO, sus aplicaciones en cálculo estructural, pero no implementan aplicación alguna ni muestran resultados experimentales. En el mismo contexto en el trabajo formulado por Bretones y Huerta en [4] se realiza un análisis con ejemplos en el que se considera el elemento finito como un objeto. Comparado con nuestro trabajo, fue desarrollado con interfaz de texto y no muestran el proceso de implementación.

En lo referente a trabajos similares, en [5], se desarrolló un algoritmo orientado a objetos para el cálculo de Flujo de Potencia utilizando el método de Newton Raspón, mediante la resolución de un sistema de ecuaciones algebraicas no lineales, en interfaz de texto. Así mismo en [6] se presenta una implementación orientada a objetos de un Framework existente, para resolución de problemas derivados de métodos discretos. Comparados con nuestro trabajo, se ha evaluado solo un método en el primer caso; y solo se mide la eficiencia del lenguaje C++, en el segundo caso.

6. CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se ha diseñado e implementado el modelado orientado a objetos con interfaz gráfica de usuario, de diversos métodos numéricos, lo que ha permitido comparar la eficiencia computacional de dichos métodos, en el cálculo de raíces reales de funciones trascendentes y polinómicas. En cuanto al análisis de resultados se observa que uno de los métodos mas eficientes es el de Newton-Raphson. Se ha determinado además que cuando se trata de funciones polinómicas, para calcular raíces reales, se puede usar cualquier método numérico que aplique la ecuación F(x)=0. Finalmente, se ha implementado una librería común de clases, que permite su reutilización (compartir las aplicaciones) para funciones trascendentes y polinómicas.

Como trabajo futuro se contempla investigar el modelado orientado a objetos de métodos para efectuar operaciones matriciales.

Referencias

[1.] Smith, W Allen. "Análisis Numérico". Prentice-Hall, ISBN: 519.4 B896, 1998.

[2.] Distributed Management Task Force, "UML profile for CIM", version 1.0, DMTF Document, DSP0219, June 2007.

[3.] Corral, M. Romera, Fernández Ruiz, Miguel. "Métodos numéricos para cálculo y diseño en ingeniería": Publicado en la Revista internacional, Vol. 21, Nº 2, pags. 179-192, 2005.

[4.] Rodríguez, F., Bretones, M. Ángel, Huerta A., "La programación orientada al objeto aplicada al cálculo por el método de los elementos finitos", Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería. Vol. 11,3, 423-449(1995). Updated: 11-03-2009.

[5.] Rodríguez, G., Aramataris, L. "Aplicación del Método de Newton Raphson Orientado a Objetos para análisis de sistemas eléctricos de potencia". Artículo Técnico. 2004.

[6.] Cardona, A., Storti, M. Zuppa, C., "Un Framework Orientado a objetos para la implementación de Métodos Discretos". Mecánica Computacional Vol 27, Argentina, 11-2008.