



PROGRAMACION ORIENTADA A OBJETOS

PARCIAL III

MARIA CIELO ACEVES VALTIERRA

22110390

3G



INDICE

| | |
|---------------------------------------|---|
| 1. Introducción | 3 |
| 2. Estructura del Proyecto | 3 |
| 3. Diagramas | 3 |
| 4. Requisitos del Sistema..... | 3 |
| 5. Tecnología Empleada | 4 |
| 6. Escalabilidad del Programa | 4 |
| 7. Configuración del Proyecto | 4 |
| 8. Uso del Proyecto | 5 |
| 9. Personalización del Proyecto | 5 |
| 10. Depuración y Pruebas | 6 |
| 11. Limitaciones conocidas | 6 |
| 12. Conclusión | 7 |

Manual de Desarrollador - Regla Falsa

1. Introducción

Este manual proporciona información detallada sobre el proyecto de programación en C++ denominado "Regla Falsa".

El proyecto consta de cuatro archivos: Proyecto_ReglaFalsa.cpp, ReglaFalsa.cpp, ReglaFalsa.h y colors.h.

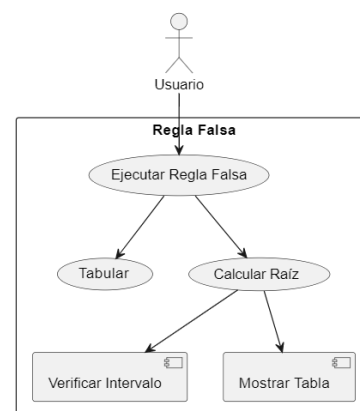
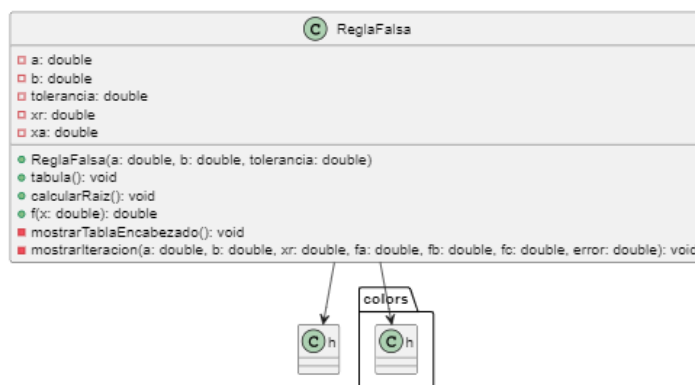
Estos archivos contienen el código fuente necesario para implementar el método de regla falsa y calcular raíces de funciones.

2. Estructura del Proyecto

El proyecto está organizado en los siguientes archivos:

- *Proyecto_ReglaFalsa.cpp*: Este archivo contiene la función principal `main()` y se encarga de llamar a la función `ejecutarReglaFalsa()` para iniciar la ejecución del método de regla falsa.
- *ReglaFalsa.cpp*: Este archivo implementa la clase `ReglaFalsa` y define sus métodos y funciones miembro. Contiene la lógica para tabular la función, calcular la raíz y mostrar los resultados.
- *ReglaFalsa.h*: Este archivo es el encabezado de la clase `ReglaFalsa` y contiene la declaración de la clase y sus métodos y variables miembro.
- *colors.h*: Este archivo define macros que se utilizan para establecer el color del texto en la salida de la consola. Proporciona una paleta de colores que se utiliza para resaltar la información en la ejecución del programa.

3. Diagramas



4. Requisitos del Sistema

Para compilar y ejecutar el proyecto, se requiere tener instalado un compilador de C++ compatible y un entorno de desarrollo integrado (IDE) o una interfaz de línea de comandos (CLI)

adecuada. Asegúrese de que el compilador y el entorno estén correctamente configurados en su sistema.

5. Tecnología Empleada

El proyecto "Regla Falsa" fue desarrollado utilizando el lenguaje de programación C++. C++ es un lenguaje de programación de propósito general que se utiliza ampliamente en el desarrollo de software. Proporciona un alto rendimiento y control de bajo nivel, lo que lo hace adecuado para aplicaciones que requieren eficiencia y manipulación directa de la memoria.

6. Escalabilidad del Programa

El programa "Regla Falsa" tiene una estructura modular y utiliza programación orientada a objetos (POO), lo que le permite ser escalable en cierta medida. Algunas consideraciones sobre la escalabilidad del programa incluyen:

- Añadir nuevas funciones: Si desea calcular raíces de diferentes funciones, puede agregar nuevas funciones en la clase ReglaFalsa modificando el método $f(x)$ en el archivo ReglaFalsa.cpp. Esto le permite extender el programa para calcular raíces de diversas funciones.
- Ampliar los métodos numéricos: Si desea implementar otros métodos numéricos para el cálculo de raíces, puede crear nuevas clases que hereden de la clase ReglaFalsa. Cada clase puede implementar un método numérico específico y proporcionar una interfaz común para el cálculo de raíces.
- Interfaz gráfica de usuario (GUI): Si desea mejorar la experiencia del usuario y permitir la interacción con el programa a través de una interfaz gráfica, puede desarrollar una GUI utilizando bibliotecas como Qt, wxWidgets o GTK+. Esto le brinda la flexibilidad de agregar características adicionales y facilitar la interacción con el programa.

La escalabilidad del programa dependerá de los requisitos específicos y del alcance del proyecto. Si se requiere una escalabilidad considerable, es posible que sea necesario considerar una arquitectura más modular o utilizar técnicas de diseño avanzadas.

7. Configuración del Proyecto

- Descargue los archivos del proyecto en una carpeta local en su sistema.
- Abra el IDE o la CLI y cargue los archivos del proyecto en el proyecto actual.
- Asegúrese de que los archivos ReglaFalsa.cpp, ReglaFalsa.h y colors.h estén incluidos en el proyecto y se compilen junto con el archivo Proyecto_ReglaFalsa.cpp.
- Configure correctamente las opciones de compilación y enlace para incluir las bibliotecas necesarias.

8. Uso del Proyecto

Para utilizar el proyecto "Regla Falsa" y calcular raíces de funciones, siga estos pasos:

1. Ejecute el programa compilado.
2. En la consola, se mostrará el mensaje "METODO REGLA FALSA" y se le pedirá que ingrese el intervalo inicial $[a, b]$ para tabular la función.
3. Ingrese el valor de a y presione Enter.
4. Ingrese el valor de b y presione Enter.
5. El programa tabulará la función en el intervalo ingresado y mostrará los valores x y $f(x)$ correspondientes.
6. Luego, se le pedirá que elija un intervalo adecuado para calcular la raíz.
7. Ingrese el nuevo valor de a y presione Enter.
8. Ingrese el nuevo valor de b y presione Enter.
9. El programa utilizará el método de regla falsa para calcular una aproximación de la raíz de la función dentro del intervalo dado.
10. Se mostrará una tabla con los valores de cada iteración, incluyendo los valores de a , b , c , $f(a)$, $f(b)$, $f(c)$, y el error absoluto $|E|$.
11. El programa continuará iterando hasta que la aproximación de la raíz alcance la tolerancia establecida en el archivo ReglaFalsa.h.
12. Una vez que se encuentre una aproximación dentro de la tolerancia, se mostrará el resultado final, indicando la raíz de la función para la tolerancia especificada.
13. Presione Enter para salir del programa.

9. Personalización del Proyecto

Puede personalizar el proyecto "Regla Falsa" realizando los siguientes cambios:

- Modificar la función $f(x)$ en el archivo ReglaFalsa.cpp para calcular el valor de la función en el punto x . Puede reemplazar la función existente $\text{pow}(x, 2) - \cos(x)$ por su propia función.
- Ajustar los valores de las constantes PRECISION y INTERVALOS en el archivo ReglaFalsa.h según sus necesidades. PRECISION define la cantidad de decimales a mostrar en los resultados, y INTERVALOS establece el número de intervalos para tabular la función.
- Cambiar los colores utilizados en la salida de la consola modificando las macros definidas en el archivo colors.h. Puede ajustar los colores del texto y del fondo según sus preferencias.

10. Depuración y Pruebas

Para depurar y probar el proyecto "Regla Falsa", puede seguir estos enfoques:

- Utilice un depurador integrado en su IDE para realizar un seguimiento paso a paso del programa y verificar el flujo de ejecución, los valores de las variables y las llamadas a las funciones.
- Asegúrese de ingresar intervalos válidos al ejecutar el programa. Verifique que los intervalos tengan un cambio de signo dentro del intervalo de tabulación para que se pueda aplicar el método de regla falsa.
- Pruebe con diferentes funciones al modificar la función $f(x)$ en el archivo ReglaFalsa.cpp. Verifique si el programa encuentra las raíces correctas para diferentes funciones.
- Experimente con diferentes valores de tolerancia y ajuste el número de intervalos para obtener resultados más precisos. Verifique la convergencia del método de regla falsa para diferentes configuraciones.
- Realice pruebas de límites y casos extremos para asegurarse de que el programa maneje correctamente situaciones como intervalos pequeños, intervalos sin cambio de signo y valores extremos de las funciones.

11. Limitaciones conocidas

El proyecto "Regla Falsa" tiene las siguientes limitaciones conocidas:

- El programa asume que el usuario ingresará intervalos válidos con un cambio de signo en la función dentro del intervalo de tabulación. No se realizan comprobaciones adicionales para garantizar que se cumpla esta condición.
- El método de regla falsa puede converger lentamente en algunos casos, especialmente cuando hay una curva plana cerca de la raíz o cuando la función tiene múltiples raíces cercanas.
- El programa no implementa técnicas avanzadas de optimización o aceleración del cálculo de raíces. Se basa en el método básico de regla falsa para encontrar aproximaciones de raíces.
- El proyecto está diseñado específicamente para resolver ecuaciones unidimensionales y puede no ser adecuado para problemas más complejos que requieran métodos numéricos más avanzados.

12. Conclusión

Este manual del desarrollador proporciona una descripción general del proyecto "Regla Falsa" implementado en C++. Le guía a través de la estructura del proyecto, los archivos involucrados y cómo utilizar el programa. Además, le ofrece pautas para personalizar el proyecto, depurar y probar el código, y destaca algunas limitaciones conocidas.

Recuerde que el proyecto "Regla Falsa" es solo un ejemplo y puede adaptarse o ampliarse para satisfacer sus necesidades específicas. A medida que continúe desarrollando sus habilidades en programación en C++ y explorando técnicas numéricas, podrá abordar problemas más desafiantes y crear soluciones más sofisticadas.