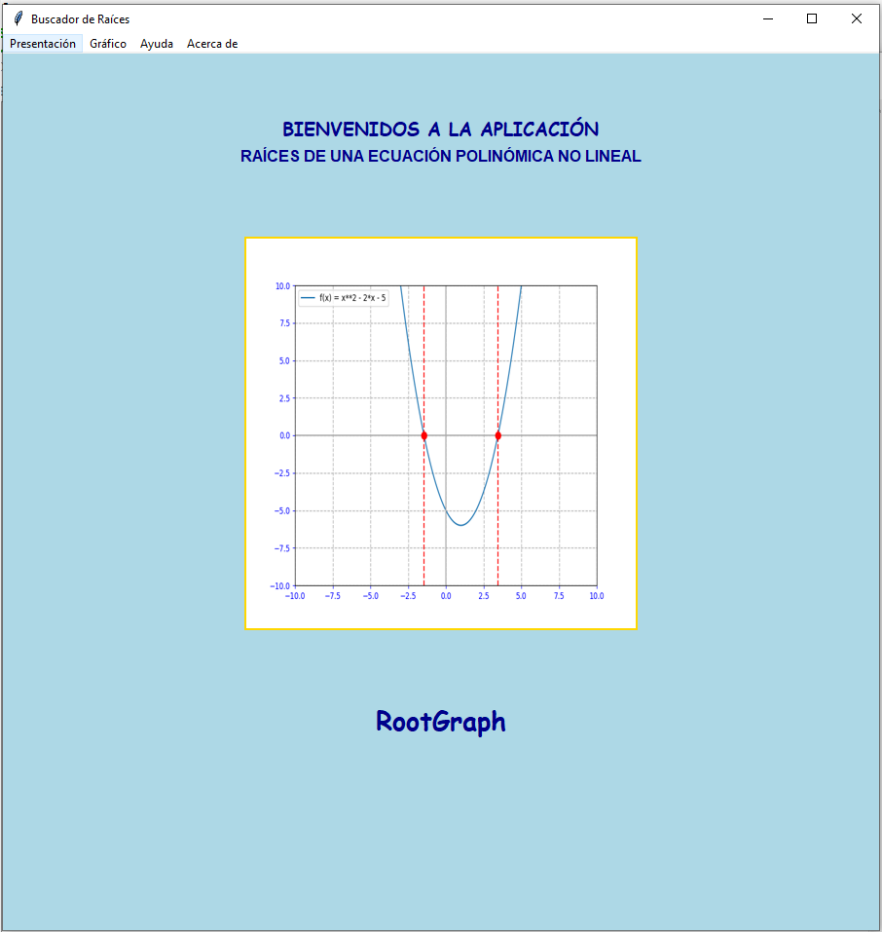


SOFTWARE DE APLICACIÓN

RAÍCES DE UNA ECUACIÓN NO LINEAL



MANUAL DE USUARIO

SOFTWARE DE APLICACIÓN

RootGraph v1.0: Software de dibujo de pares ordenados en un Plano cartesiano

AUTORES:

- LEONEL COYLA IDME
- ALFREDO MAMANI CANQUI
- ELQUI YEYE PARI CONDORI
- JUAN REYNALDO PAREDES QUISPE
- JOSÉ PÁNFILO TITO LIPA

ÍNDICE

1.	INTRODUCCIÓN	4
2.	REQUISITOS DEL SISTEMA	5
2.1.	SISTEMA OPERATIVO.....	5
2.2.	HARDWARE	5
2.3.	SOFTWARE	5
3.	INSTALACIÓN Y CONFIGURACIÓN	5
3.1.	INSTALAR PYTHON	5
3.2.	EJECUTAR LA APLICACIÓN	5
3.3.	CONFIGURACIÓN OPCIONAL	6
3.4.	SOLUCIÓN DE PROBLEMA.....	6
3.5.	ACTUALIZACIÓN	6
4.	DESCRIPCIÓN DE LA APLICACIÓN RootGraph v1.0.....	8
4.1.	MENÚ PRESENTACIÓN	8
4.3.	MENÚ AYUDA	12
4.4.	MENÚ ACERCA DE	12
5.	PREGUNTAS FRECUENTES (FAQS) SOBRE EL SOFTWARE	14
5.1.	¿QUÉ ES ROOTGRAPH Y CUÁL ES SU PROPÓSITO?.....	14
5.2.	¿CUÁLES SON LOS REQUISITOS MÍNIMOS DEL SISTEMA PARA EJECUTAR?	14
5.3.	¿CÓMO INSTALO O CONFIGURO ROOTGRAPH?.....	14
5.4.	¿QUÉ SE DEBE REALIZAR SI EL SOFTWARE NO FUNCIONA CORRECTAMENTE?.....	15
6.	ADVERTENCIAS Y SUGERENCIAS DEL SOFTWARE DE APLICACIÓN.....	15
6.1.	ADVERTENCIAS.....	15
6.2.	SUGERENCIAS.....	15
7.	CÓDIGO	16

1. INTRODUCCIÓN

RootGraph v1.0 es un software educativo diseñado para facilitar el aprendizaje de los estudiantes que cursan la asignatura de Métodos Numéricos. Su principal función es graficar ecuaciones no lineales y calcular de manera aproximada la raíz de dichas ecuaciones. Estos procedimientos son fundamentales en el análisis numérico, ya que permiten encontrar soluciones a problemas matemáticos que no se resuelven fácilmente de forma algebraica.

La aplicación ofrece una interfaz gráfica amigable y didáctica, donde el usuario puede ingresar una función matemática y observar su comportamiento en un sistema de coordenadas. El gráfico generado permite identificar visualmente el punto donde la curva corta al eje x , es decir, la raíz de la ecuación. Además, el software realiza el cálculo de la raíz, lo que ayuda al estudiante a comprobar sus resultados y mejorar su comprensión de los métodos numéricos.

Gracias a su diseño práctico y funcional, RootGraph v1.0 se convierte en una herramienta de apoyo ideal para el proceso de enseñanza-aprendizaje. Su uso permite reforzar conceptos teóricos mediante la experimentación visual e interactiva, contribuyendo al desarrollo de habilidades analíticas y al entendimiento de la importancia de los métodos numéricos en la resolución de problemas matemáticos reales.

Adicionalmente, RootGraph v1.0 está pensado para ser utilizado tanto en entornos académicos como de autoaprendizaje. Su facilidad de instalación, sus herramientas intuitivas y sus resultados inmediatos permiten que cualquier estudiante o docente pueda utilizar el software sin complicaciones, promoviendo un aprendizaje autónomo y dinámico dentro y fuera del aula.

PROYECTO REGISTRADO CON URL:

<https://github.com/leonelcoyla/RootGraph>

2. REQUISITOS DEL SISTEMA

Para garantizar un rendimiento óptimo de RootGraph v1.0, el software diseñado para construir gráficos de ecuaciones no lineales y calcular la raíz de dicha ecuación, es fundamental que tu sistema cumpla con los requisitos mínimos y recomendados. Asegúrate de que tanto el hardware como el software de tu equipo estén correctamente configurados antes de instalar y utilizar el programa.

2.1. Sistema Operativo

Al aplicativo RootGraph v1.0 es compatible con los siguientes sistemas operativos:

- Windows 7, 8, 10 o superior
- Linux (Ubuntu 18.04 o superior)
- macOS (versión 10.12 o superior)

2.2. Hardware

- Procesador: Intel Core i3 o superior
- Memoria RAM: 2 GB mínimo (recomendado 4 GB)
- Espacio en Disco: 100 MB libres
- Resolución de pantalla mínima: 1024x768

2.3. Software

- Python 3.10 o superior
- Librerías de Python necesarias:
 - matplotlib
 - tkinter
 - pandas

3. INSTALACIÓN Y CONFIGURACIÓN

Para poner en funcionamiento RootGraph v1.0, debes asegurarte de que los siguientes componentes estén correctamente instalados en tu equipo:

3.1. Instalar Python

- Descargar Python desde la página oficial:
<https://www.python.org/downloads/>
- Durante la instalación, marcar la opción "Add Python to PATH"
- Finalizar el proceso de instalación.
-

3.2. Ejecutar la Aplicación

- Guarde el código dado en un archivo con el nombre RootGraph.py.
- Abra la terminal, diríjase a la carpeta donde almacenó el archivo y ejecútelo.:

python RootGraph.py

- También es posible ejecutar el archivo RootGraph.exe directamente desde su ubicación.
- Otra opción es instalar el programa ejecutando mysetup.exe desde la carpeta donde se encuentra.

Archivo principal

RootGraph.py #código fuente

RootGraph.exe archivo ejecutable

mysetup.exe archivo instalador de la aplicación

CONTACTO DEL DESARROLLADOR: +51951679658

3.3. Configuración opcional

RootGraph v1.0 está configurado para funcionar con una resolución de pantalla mínima de **1024x768 px**. Si deseas usar una resolución mayor, puedes modificar el tamaño de la ventana en el código.

- Abre el archivo **RootGraph.py** en un editor de texto.
- Busca la línea donde se establece el tamaño de la ventana (en el código proporcionado, es `root.geometry("900x900")`)
- Cambia las dimensiones a tu preferencia. Por ejemplo:

```
root.geometry("900x900") # Ajusta el tamaño de la ventana
```

3.4. Solución de problema

Problema	Solución
Python no se reconoce como comando	Asegurarse de haber marcado <i>Add Python to PATH</i> en la instalación.
Error al instalar librerías	Ejecutar <code>pip install --upgrade pip</code> antes de instalar.
La aplicación no abre	Verificar que el archivo esté completo y en la ubicación correcta.

3.5. Actualización

- Para actualizar RootGraph v.1.0:
- Verificar si existe una nueva versión disponible en el sitio oficial o repositorio.

- Descargar la nueva versión.
- Reemplazar los archivos antiguos.
- Ejecutar nuevamente la aplicación.

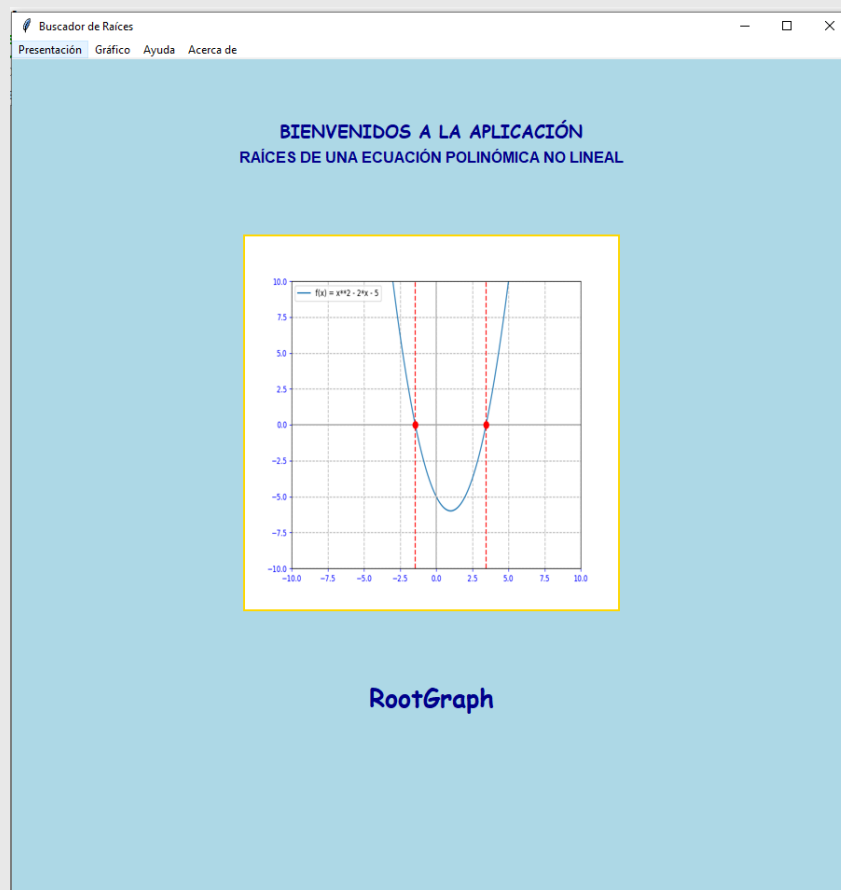
4. DESCRIPCIÓN DE LA APLICACIÓN RootGraph v1.0

RootGraph v1.0 es una aplicación educativa desarrollada con el objetivo de facilitar el aprendizaje de conceptos matemáticos fundamentales, como la representación gráfica de funciones no lineales y el cálculo de raíces de ecuaciones. Esta herramienta está orientada principalmente a estudiantes que desean comprender de manera visual y práctica cómo se comportan las funciones en el plano cartesiano.

El software permite al usuario ingresar una ecuación no lineal, visualizar su gráfica de forma precisa y determinar de manera aproximada la raíz o solución de dicha ecuación. RootGraph v1.0 ofrece una interfaz intuitiva, amigable y con colores llamativos, lo que hace que la experiencia de aprendizaje sea más dinámica y accesible.

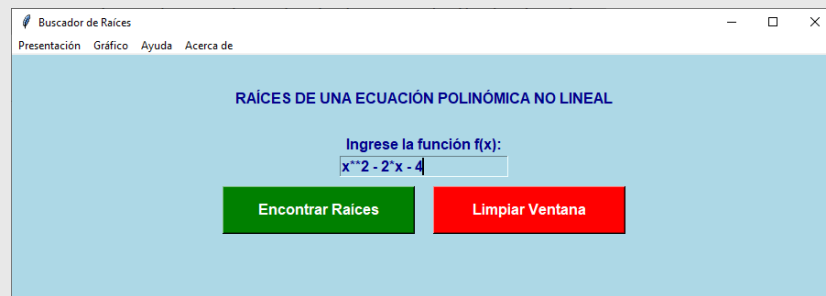
4.1. Menú Presentación

Al iniciar RootGraph v1.0, los usuarios son recibidos en el menú de presentación, donde se muestra un mensaje de bienvenida, acompañado del logotipo representativo de la aplicación y su nombre resaltado. Este espacio está diseñado para introducir al usuario a la interfaz y al propósito principal del software, permitiéndole conocer sus funciones antes de proceder a graficar ecuaciones no lineales y calcular sus respectivas raíces.

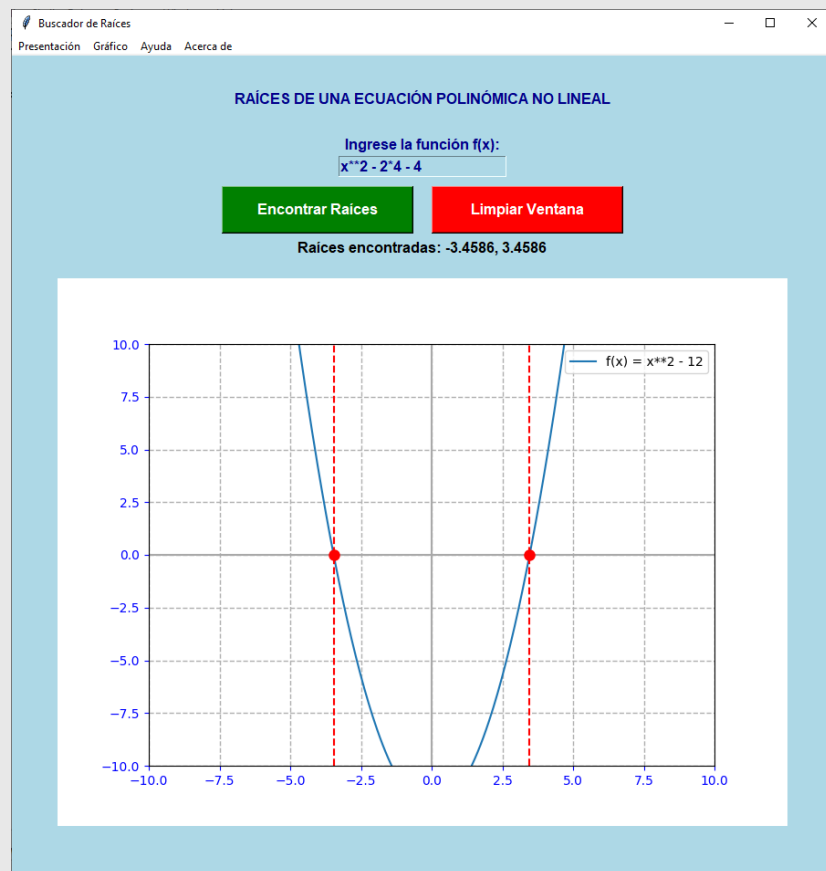


4.2. RootGraph v1.0

La pestaña "Gráfico" de *RootGraph v1.0* es la sección central de la aplicación, diseñada para interactuar con ecuaciones no lineales. En primer lugar, el usuario debe ingresar la ecuación matemática que desea graficar en un campo de texto específico.

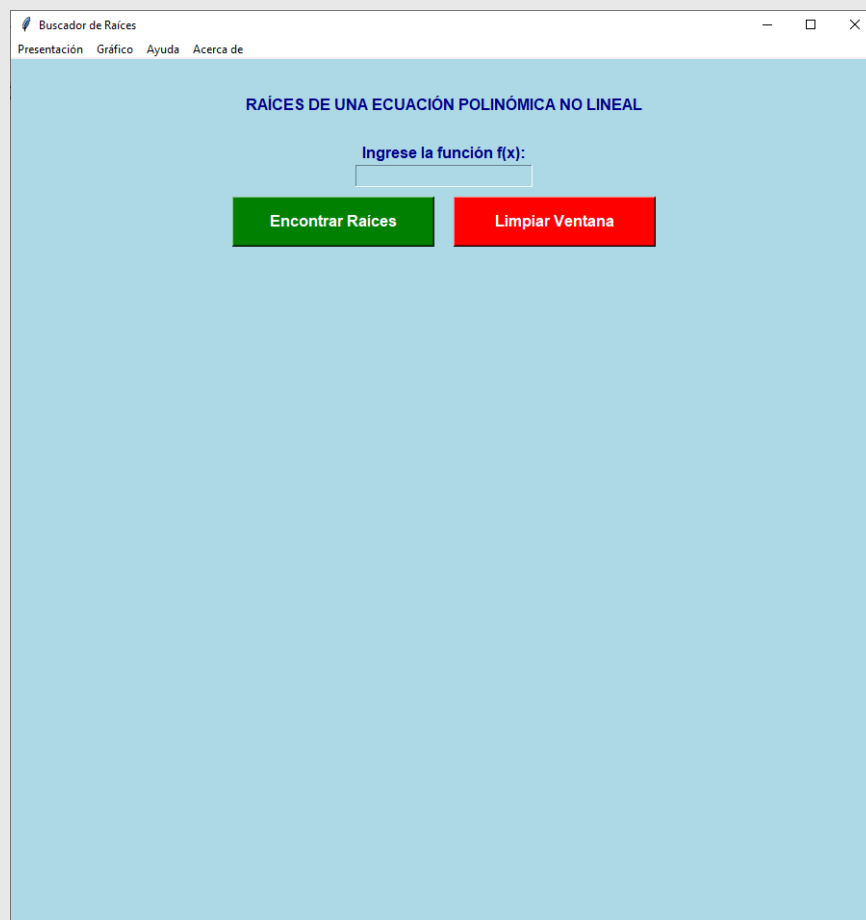


Una vez ingresada la ecuación, haga clic en el botón 'Encontrar Raíces'. Esto generará el gráfico de la ecuación y mostrará la(s) raíz(es) en el plano cartesiano, permitiendo al usuario visualizar el comportamiento de la función de manera clara y precisa.



Al generar el gráfico y mostrar las raíces de la ecuación en el intervalo de interés, el software indica los puntos donde la curva intersecta el eje x, lo que representa visualmente las soluciones de la ecuación. Además de crear el gráfico, RootGraph v1.0 resalta las raíces con marcas visuales, ayudando a los estudiantes a comprender cómo se encuentran las soluciones de las ecuaciones no lineales en el contexto de los métodos numéricos

Al finalizar el análisis, el usuario tiene la opción de borrar el gráfico generado, lo que le permite ingresar una nueva ecuación y realizar un nuevo cálculo. Este ciclo interactivo facilita el aprendizaje, permitiendo que los estudiantes experimenten con diferentes funciones y valores, y mejoren su comprensión de los conceptos teóricos mediante la visualización dinámica y el cálculo de raíces.



El código de RootGraph v1.0 está diseñado para trabajar con una amplia variedad de **funciones matemáticas**, siempre que sean compatibles con la sintaxis de **SymPy** y **NumPy**. A continuación, te menciono los principales tipos de funciones que puedes ingresar:

Tipos de funciones compatibles:

1. Funciones polinómicas:

Ejemplo: $x^{**2} - 4*x + 3$, $x^{**5} + 2*x - 7$

2. **Funciones racionales:**

Ejemplo: $(x^2 - 1)/(x + 1)$

3. **Funciones trigonométricas:**

Ejemplo: $\sin(x)$, $\cos(x)$, $\tan(x)$, $\sin(x) - x/2$

4. **Funciones exponenciales y logarítmicas:**

Ejemplo: $\exp(x) - 2$, $\log(x + 1)$

5. **Funciones combinadas:**

Ejemplo: $\sin(x) + x^2 - \log(x + 2)$

6. **Funciones raíz o radicales:**

Ejemplo: $\sqrt{x + 4}$, $x^{1/3}$

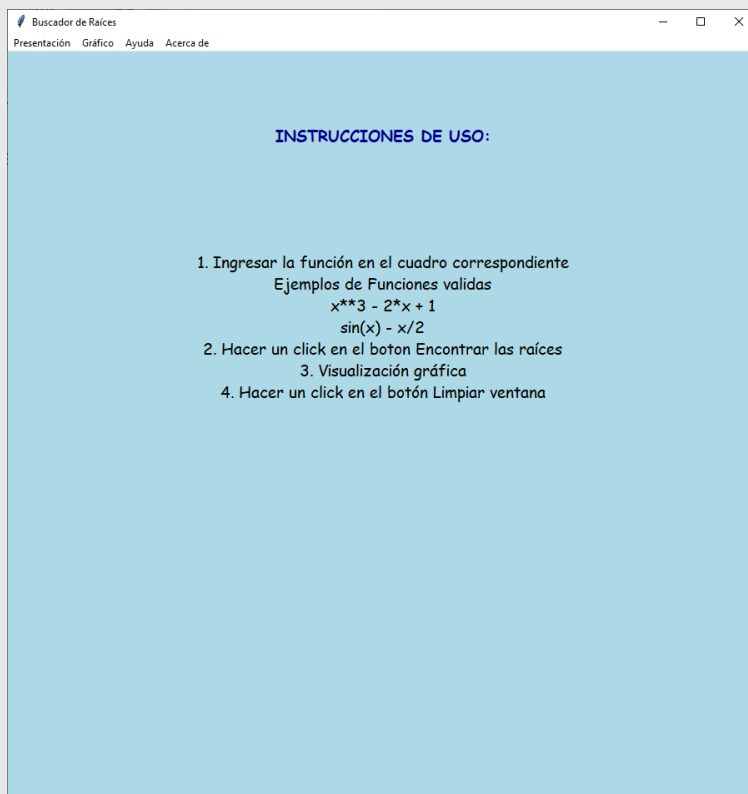
Importante:

- La variable debe ser escrita como x.
- Usa ** para potencias, no ^ (por ejemplo, escribe x^2 y no x^2).
- No ingreses funciones fuera del dominio válido (por ejemplo, $\log(x)$ con $x \leq 0$) ya que puede generar errores.

4.3. Menú Ayuda

El menú "Ayuda" ofrece una guía práctica y accesible que facilita a los estudiantes la comprensión y el uso adecuado del software RootGraph v1.0. Este recurso está diseñado para proporcionar información clara y directa a los usuarios.

Su objetivo principal es brindar orientación de manera sencilla, permitiendo a los estudiantes familiarizarse con las principales funciones de la aplicación y así aprovechar al máximo las herramientas que ofrece el programa.

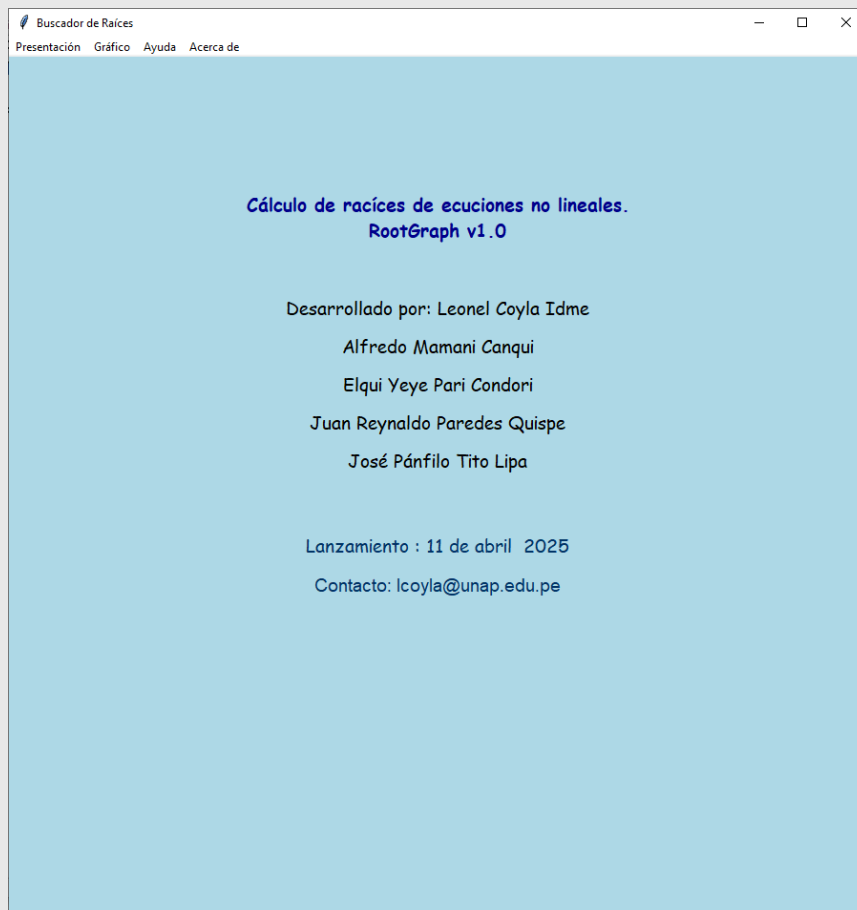


4.4. Menú Acerca de

En la sección "Acerca de", los usuarios podrán encontrar información importante sobre RootGraph v1.0, una aplicación educativa desarrollada para facilitar la elaboración de gráficos de ecuaciones no lineales y determinar sus respectivas raíces. Este apartado proporciona detalles como el nombre completo del software, la versión actual, el nombre del desarrollador principal, Leonel Coyla Idme, la fecha de lanzamiento, así como los datos de contacto disponibles para consultas, sugerencias o soporte técnico.

Este apartado tiene como finalidad brindar detalles sobre la autoría y procedencia de la herramienta, promoviendo la confianza de los usuarios al utilizarla. Además, facilita un canal de comunicación directo con el desarrollador para resolver dudas, realizar consultas o compartir sugerencias. Gracias a esta

sección, los estudiantes, docentes y padres pueden conocer de manera clara la procedencia del software y confirmar que están utilizando una herramienta confiable y diseñada con fines educativos específicos.



5. PREGUNTAS FRECUENTES (FAQS) SOBRE EL SOFTWARE

5.1. ¿Qué es RootGraph y cuál es su propósito?

RootGraph v1.0 es un software educativo diseñado para facilitar el aprendizaje de los estudiantes que cursan la asignatura de Métodos Numéricos. Su propósito principal es graficar ecuaciones no lineales y calcular de manera aproximada la raíz de dichas ecuaciones. Estos procedimientos son fundamentales en el análisis numérico, ya que permiten encontrar soluciones a problemas matemáticos que no se resuelven fácilmente de forma algebraica.

5.2. ¿Cuáles son los requisitos mínimos del sistema para ejecutar?

Para ejecutar correctamente la aplicación RootGraph v1.0, se requiere un sistema operativo Windows 7 o superior, aunque también es compatible con macOS y distribuciones de Linux. Es necesario contar con al menos 2 GB de memoria RAM y un procesador de doble núcleo a 1.5 GHz o superior. Se recomienda una resolución de pantalla mínima de 1024x768 píxeles para una visualización adecuada de la interfaz. Además, debe tener instalado Python 3.8 o superior, junto con las bibliotecas Tkinter, SymPy, NumPy, Matplotlib y Pillow. Es fundamental tener acceso a un entorno gráfico que permita ejecutar interfaces basadas en Tkinter. El espacio en disco requerido es mínimo, aproximadamente 100 MB. Se sugiere mantener el sistema actualizado para garantizar compatibilidad y rendimiento óptimo.

5.3. ¿Cómo instalo o configuro RootGraph?

Para instalar y configurar RootGraph v1.0, primero debes tener Python 3.8 o una versión superior instalado en tu equipo; puedes descargarlo desde la página oficial de Python y asegurarte de marcar la opción "Add Python to PATH" durante la instalación. Luego, crea una carpeta exclusiva donde guardarás el archivo principal RootGraph.py junto con la imagen Graph.png. Abre una terminal y ejecuta el comando `pip install sympy numpy matplotlib pillow` para instalar las bibliotecas necesarias. Una vez instaladas, navega en la terminal hasta la carpeta del proyecto y ejecuta el comando `python RootGraph.py` para iniciar la aplicación. RootGraph funcionará correctamente en sistemas Windows, macOS o Linux que cuenten con al menos 2 GB de RAM y un procesador de doble núcleo. Se recomienda una resolución mínima de pantalla de 1024x768 píxeles para una visualización óptima. Es importante contar con un entorno gráfico disponible, especialmente en sistemas Linux. También puedes crear un acceso directo o archivo .bat para facilitar su ejecución.

5.4. ¿Qué se debe realizar si el software no funciona correctamente?

Si el software RootGraph v1.0 no funciona correctamente, lo primero que se debe hacer es verificar que todas las bibliotecas necesarias estén correctamente instaladas, incluyendo sympy, numpy, matplotlib, pillow y que se esté utilizando Python 3.8 o una versión superior. También es importante comprobar que no existan errores de escritura en la función ingresada y que la imagen Graph.png esté ubicada en la misma carpeta del programa. Si el problema persiste, se recomienda reiniciar la aplicación y, de ser necesario, el sistema operativo. En caso de que los errores continúen, se debe contactar al desarrollador principal a través del correo electrónico proporcionado en la sección "Acerca de" del software (lcoyla@unap.edu.pe), describiendo detalladamente el inconveniente, incluyendo capturas de pantalla o mensajes de error si es posible.

6. ADVERTENCIAS Y SUGERENCIAS DEL SOFTWARE DE APLICACIÓN.

6.1. Advertencias

Es importante tener en cuenta que RootGraph v1.0 está diseñado exclusivamente con fines educativos, por lo que no debe utilizarse para aplicaciones técnicas o profesionales que requieran alta precisión en cálculos numéricos. El uso incorrecto del software, como el ingreso de funciones mal escritas o fuera del dominio permitido, puede generar errores o resultados inesperados. Asimismo, modificar el código fuente sin conocimientos adecuados puede afectar su funcionamiento. Se recomienda no eliminar ni cambiar el nombre del archivo de imagen Graph.png, ya que esto podría impedir la correcta visualización en la sección de presentación. Por último, se sugiere utilizar la aplicación en sistemas compatibles y mantener las bibliotecas actualizadas para evitar problemas de ejecución.

6.2. Sugerencias

Se sugiere a los usuarios explorar y practicar con diversas funciones matemáticas dentro de la aplicación RootGraph v1.0 para familiarizarse con el comportamiento gráfico de las ecuaciones no lineales y el proceso de búsqueda de raíces. Esta interacción no solo refuerza el aprendizaje teórico, sino que también permite desarrollar habilidades analíticas al interpretar los resultados visuales obtenidos. Además, se recomienda utilizar funciones reales aplicadas a contextos cotidianos o problemas universitarios, lo que facilitará una comprensión más significativa y motivadora del tema.

7. Código

```
1 import tkinter as tk
2 import sympy as sp
3 import numpy as np
4 import matplotlib.pyplot as plt
5 from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
6
7 from tkinter import PhotoImage
8 from PIL import Image, ImageTk
9
10 def encontrarRaices():
11     limpiarFrame()
12     for widget in canvas_frame.winfo_children():
13         widget.destroy()
14
15     tk.Label(frameContenido, text="", font=("Arial", 12, "bold"), fg="#00008B", bg="lightblue").pack()
16     tk.Label(frameContenido, text="RAÍCES DE UNA ECUACIÓN POLINÓMICA NO LINEAL", font=("Arial", 12, "bold"), fg="#00008B", bg="lightblue").pack()
17     tk.Label(frameContenido, text="", font=("Arial", 12, "bold"), fg="#00008B", bg="lightblue").pack()
18     tk.Label(frameContenido, text="Ingrese la función f(x):", font=("Arial", 12, "bold"), fg="#00008B", bg="lightblue").pack()
19
20     entry_function = tk.Entry(frameContenido, font=("Arial", 12, "bold"), fg="#00008B", bg="lightblue")
21     entry_function.pack()
22
23     def calcular():
24         try:
25             expr_str = entry_function.get()
26             x = sp.symbols('x')
27             expr = sp.sympify(expr_str)
28
29             f_lambdified = sp.lambdify(x, expr, 'numpy')
30
31             x_vals = np.linspace(-10, 10, 400)
32             y_vals = f_lambdified(x_vals)
33
34             roots = []
35             for i in range(len(x_vals) - 1):
36                 if y_vals[i] * y_vals[i + 1] < 0:
37                     root_approx = (x_vals[i] + x_vals[i + 1]) / 2
38                     roots.append(root_approx)
39
40             if roots:
41                 root_label.config(text=f"Raíces encontradas: {', '.join(map(lambda r: f'{r:.4f}', roots))}", font=("Arial", 12, "bold"))
42             else:
43                 root_label.config(text="No se encontraron raíces en el intervalo.")
44
45             functionPlot(expr, roots)
46
47         except Exception as e:
48             root_label.config(text=f"Error: Ingrese la ecuación")
49
50     def limpiarVentana():
51         entry_function.delete(0, tk.END)
52         root_label.config(text="")
53         info_label.config(text="")
54         for widget in canvas_frame.winfo_children():
55             widget.destroy()
56
57     tk.Button(frameContenido, text="Encontrar Raíces", command=calcular, fg="white", bg="green", font=("Arial", 12, "bold"), height=2, width=20).pack(side="left", padx=10, pady=(10, 3))
58     tk.Button(frameContenido, text="Limpiar Ventana", command=limpiarVentana, fg="white", bg="red", font=("Arial", 12, "bold"), height=2, width=20).pack(side="left", padx=10, pady=(10, 3))
59
60     def functionPlot(expr, roots):
61         x = sp.symbols('x')
62         f_lambdified = sp.lambdify(x, expr, 'numpy')
63
64         x_vals = np.linspace(-10, 10, 400)
65         y_vals = f_lambdified(x_vals)
66
67         fig, ax = plt.subplots(figsize=(8, 6))
68         ax.plot(x_vals, y_vals, label=f"f(x) = {expr}")
69
70         ax.axhline(0, color='darkgray', linewidth=1.5)
71         ax.axvline(0, color='darkgray', linewidth=1.5)
72
73         ax.xaxis.label.set_color('blue')
74         ax.yaxis.label.set_color('blue')
75         ax.tick_params(axis='x', colors='blue')
76         ax.tick_params(axis='y', colors='blue')
77
78         for root in roots:
79             ax.axvline(root, color='red', linestyle='--')
80             ax.plot(root, 0, 'ro', markersize=8)
```



```

80     ax.plot(root, 0, 'ro', markersize=8)
81
82     ax.set_xlim([-10, 10])
83     ax.set_ylim([-10, 10])
84     ax.grid(True, linestyle='--', linewidth=1)
85     ax.legend()
86
87     for widget in canvas_frame.winfo_children():
88         widget.destroy()
89
90     canvas = FigureCanvasTkAgg(fig, master=canvas_frame)
91     canvas.draw()
92     canvas.get_tk_widget().pack()
93
94     def limpiarFrame():
95         for widget in frameContenido.winfo_children():
96             widget.destroy()
97
98     def mostrarPresentacion():
99         limpiarFrame()
100         for widget in canvas_frame.winfo_children():
101             widget.destroy()
102
103         root_label.config(text="")
104         info_label.config(text="")
105
106         tk.Label(frameContenido, text="", font=("Arial", 12, "bold"), fg="#00008B", bg="lightblue").pack()
107         tk.Label(frameContenido, text="", font=("Arial", 12, "bold"), fg="#00008B", bg="lightblue").pack()
108         tk.Label(frameContenido, text="BIENVENIDOS A LA APLICACIÓN", font=("Comic Sans MS", 14, "bold"), fg="#00008B", bg="lightblue").pack()
109         tk.Label(frameContenido, text="RAÍCES DE UNA ECUACIÓN POLINOMICA NO LINEAL", font=("Arial", 12, "bold"), fg="#00008B", bg="lightblue").pack()
110         tk.Label(frameContenido, text="", font=("Arial", 12, "bold"), fg="#00008B", bg="lightblue").pack()
111         tk.Label(frameContenido, text="", font=("Arial", 12, "bold"), fg="#00008B", bg="lightblue").pack()
112
113         try:
114             image = Image.open("Graph.png")
115             image = image.resize((400, 400))
116             imageTk = ImageTk.PhotoImage(image)
117             image_label = tk.Label(frameContenido, image=imageTk, bg="#FFD700")
118             image_label.image = imageTk
119             image_label.pack(pady=20)
120         except Exception as e:
121             print(f"Error al cargar la imagen: {e}")
122
123         tk.Label(frameContenido, text="", font=("Arial", 12, "bold"), fg="#00008B", bg="lightblue").pack()
124         tk.Label(frameContenido, text="", font=("Arial", 12, "bold"), fg="#00008B", bg="lightblue").pack()
125         tk.Label(frameContenido, text="RootGraph", font=("Comic Sans MS", 20, "bold"), fg="#00008B", bg="lightblue").pack()
126
127     def mostrarAyuda():
128         limpiarFrame()
129         for widget in canvas_frame.winfo_children():
130             widget.destroy()
131
132         root_label.config(text="")
133         info_label.config(text="")
134
135         tk.Label(frameContenido, text="\n\nINSTRUCCIONES DE USO:\n\n", bg="lightblue", font=("Comic Sans MS", 14, "bold"), fg="#00008B").pack(pady=20)
136         tk.Label(frameContenido, text="\n1. Ingresar la función en el cuadro correspondiente"
137             f"\nEjemplos de Funciones validas\nx**3 - 2*x + 1\nsin(x) - x/2\n"
138             f"2. Hacer un click en el boton Encontrar las raíces"
139             f"3. Visualización gráfica\n4. Hacer un click en el botón Limpiar ventana", bg="lightblue", font=("Comic Sans MS", 14), pack(pady=20))
140
141     def mostrarAcerca():
142         limpiarFrame()
143         for widget in canvas_frame.winfo_children():
144             widget.destroy()
145
146         root_label.config(text="")
147         info_label.config(text="")
148
149         tk.Label(frameContenido, text="\n\n\nCálculo de raíces de ecuaciones no lineales."
150             f"\nRootGraph v1.0", bg="lightblue", font=("Comic Sans MS", 14, "bold"), fg="#00008B").pack(pady=20)
151
152         textos = [
153             "\nDesarrollado por: Leonel Coyla Idme",
154             "Alfredo Mamani Canqui",
155             "Elqui Yaya Pari Condoni",
156             "Juan Reynaldo Paredes Quispe",
157             "José Pánfilo Tito Lipa",
158         ]
159
160         for texto in textos:

```

```

161     tk.Label(frameContenido, text=texto, bg="lightblue", font=("Comic Sans MS", 14)).pack(pady=4)
162
163     label_Acerca_de = tk.Label(frameContenido, text= "\n\nLanzamiento : 11 de abril 2025", font=("Comic Sans MS", 14), fg="#003366", bg="lightblue")
164     label_Acerca_de.pack(pady=(1,10))
165     label_Acerca_de = tk.Label(frameContenido, text= "Contacto: lcoyla@unap.edu.pe", font=("Comic Sans MS", 14), fg="#003366", bg="lightblue")
166     label_Acerca_de.pack(pady=(1,10))
167
168 |
169 def crearInterfaz():
170     global frameContenido, root_label, info_label, canvas_frame
171
172     root = tk.Tk()
173     root.title("Buscador de Raíces")
174     root.geometry("900x900")
175     root.config(bg="lightblue")
176
177     menu_bar = tk.Menu(root)
178     root.config(menu=menu_bar)
179
180     menu_bar.add_command(label="Presentación", command=mostrarPresentacion)
181     menu_bar.add_command(label="Gráfico", command=encontrarRaices)
182     menu_bar.add_command(label="Ayuda", command=mostrarAyuda)
183     menu_bar.add_command(label="Acerca de", command=mostrarAcerca)
184
185     main_frame = tk.Frame(root, bg="lightblue")
186     main_frame.pack(pady=10)
187
188     frameContenido = tk.Frame(main_frame, bg="lightblue")
189     frameContenido.pack()
190
191     root_label = tk.Label(main_frame, text="", bg="lightblue")
192     root_label.pack()
193
194     info_label = tk.Label(main_frame, text="", fg="blue", bg="lightblue")
195     info_label.pack()
196
197     canvas_frame = tk.Frame(main_frame, bg="lightblue")
198     canvas_frame.pack()
199
200     root.mainloop()
201
202
203 def main():
204     crearInterfaz()
205
206 if __name__ == "__main__":
207     main()

```