

SOFTWARE DE APLICACIÓN

SOLUCIÓN NUMÉRICA DE UNA ECUACIÓN DIFERENCIAL ORDINARIA

MÉTODO EULER



MANUAL DE USUARIO

SOFTWARE DE APLICACIÓN

El software de aplicación del método de Euler SoftEDOEuler v1.0 es uno de los métodos numéricos más básicos y fundamentales para resolver ecuaciones diferenciales ordinarias (EDO) de primer orden. Se usa cuando no se puede encontrar una solución analítica exacta.

AUTORES:

- LEONEL COYLA IDME
- 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.	CONFIGURACIÓN OPCIONAL	6
3.3.	ACTUALIZACIÓN	6
4.	DESCRIPCIÓN DE LA APLICACIÓN SOFTEDOEULER v1.0	7
4.1.	MENÚ PRESENTACIÓN	8
4.2.	EULER	8
4.3.	MENÚ AYUDA	11
4.4.	MENÚ ACERCA DE	11
5.	PREGUNTAS FRECUENTES (FAQS) SOBRE EL SOFTWARE	13
5.1.	¿QUÉ ES MÉTODO EULER EN ECUACIONES DIFERENCIALES ORDINARIAS Y CUÁL ES SU PROPÓSITO?	13
5.2.	¿CUÁLES SON LOS REQUISITOS MÍNIMOS DEL SISTEMA PARA EJECUTAR?	13
5.3.	¿CÓMO INSTALO O CONFIGURO SOFTEDOEULER?	13
6.	ADVERTENCIAS Y SUGERENCIAS DEL SOFTWARE DE APLICACIÓN.	14
6.1.	ADVERTENCIAS.....	14
6.2.	SUGERENCIAS	15
7.	CÓDIGO	16

1. INTRODUCCIÓN

SoftEDOEuler v1.0 es una aplicación educativa diseñada para facilitar la resolución de ecuaciones diferenciales ordinarias (EDO) de primer orden mediante el método numérico de Euler. Su enfoque práctico permite visualizar y comprender de forma sencilla cómo se aproximan las soluciones de estas ecuaciones a través de este método básico, pero fundamental en el estudio de métodos numéricos.

Pensada para estudiantes, docentes e investigadores, SoftEDOEuler ofrece una herramienta intuitiva y funcional que apoya el análisis y la enseñanza del comportamiento de soluciones aproximadas. Su diseño accesible permite explorar conceptos clave sin la necesidad de software complejo, favoreciendo así el aprendizaje activo y la experimentación matemática.

SoftEDOEuler ha sido desarrollado en Python, utilizando una interfaz simple y fácil de usar, pensada para ejecutarse en sistemas de escritorio.

PROYECTO REGISTRADO CON URL:

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

2. REQUISITOS DEL SISTEMA

Para asegurar un rendimiento óptimo de **SoftEDOEuler v1.0**, el software diseñado para resolver ecuaciones diferenciales ordinarias de primer orden es fundamental que tu equipo cumpla con los requisitos mínimos y recomendados. Antes de instalar y utilizar el programa, verifica que tanto el hardware como el software estén correctamente configurados.

2.1. Sistema Operativo

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

- Windows: 7, 8, 10, 11 (recomendado Windows 10 o superior)
- macOS: 10.14 Mojave o superior
- Linux: Ubuntu 18.04 o superior, Debian, Fedora, etc.

2.2. Hardware

- Procesador: Intel/AMD 1 GHz o superior
- Memoria RAM: 2 GB mínimo (recomendado 4 GB)
- Espacio en disco: 100 MB libres
- Pantalla: 1024x768 píxeles o superior

2.3. Software

- Python 3.8 o superior
- Librerías: matplotlib, numpy, tkinter (opcional para interfaz gráfica)
- Tkinter: Incluido por defecto en la instalación oficial de Python
- Editor sugerido: VSCode, Thonny, IDLE o cualquier IDE para Python

3. INSTALACIÓN Y CONFIGURACIÓN

Para ejecutar correctamente **SoftEDOEuler v1.0**, es necesario comprobar que los siguientes componentes estén debidamente instalados en tu equipo.

3.1. Instalar Python

- Visite el sitio oficial: <https://www.python.org>
- Descargue la versión recomendada para su sistema operativo.
- Ejecute el instalador y marque la opción “Add Python to PATH” antes de continuar.
- Verifique la instalación abriendo una terminal o símbolo del sistema y ejecutando:
 - `python --version`
- Ejecutar la Aplicación
 - Guarde el código en un archivo llamado `SoftEDOEuler.py`.

- Abre una terminal o consola
- Dirígete a la carpeta donde está el archivo
- Ejecutar el programa

`python SoftEDOEuler.py` #código fuente

`SoftEDOEuler.exe` archivo ejecutable

`mysetup.exe` archivo instalador de la aplicación

CONTACTO DEL DESARROLLADOR: +51951679658

3.2. Configuración opcional

SoftEDOEuler v1.0 ha sido desarrollado para operar con una resolución mínima de 1024x768 píxeles. Si deseas utilizar una resolución mayor, puedes modificar el tamaño de la ventana directamente desde el código fuente.

- Abre el archivo **SoftEDOEuler.py** en un editor de texto.
- Busca la línea donde se establece el tamaño de la ventana (en el código proporcionado, es `selt.geometry("1150x800")`)
- Cambia las dimensiones de acuerdo con tu necesidad.

3.3. Actualización

- Para actualizar SoftEDOEuler v.1.0
- Verificar si existe una nueva versión disponible en el sitio oficial o repositorio.
- Descargar la versión más reciente.
- Reemplazar los archivos antiguos.
- Ejecutar nuevamente la aplicación.

4. DESCRIPCIÓN DE LA APLICACIÓN SoftEDOEuler v1.0

El método de Euler fue desarrollado por el matemático suizo Leonhard Euler en el siglo XVIII, específicamente alrededor del año 1768, aunque sus trabajos sobre cálculo y ecuaciones diferenciales comenzaron mucho antes.

El método de Euler es una técnica numérica básica utilizada para aproximar soluciones de ecuaciones diferenciales ordinarias (EDO) de primer orden. Se basa en el concepto de derivada como tasa de cambio, utilizando la pendiente de la función en un punto conocido para estimar el valor de la solución en el siguiente punto. A partir de una condición inicial, el método avanza paso a paso, sumando al valor actual una corrección proporcional a la derivada y al tamaño del paso. Aunque su precisión es limitada en comparación con métodos más avanzados, como Runge-Kutta, el método de Euler es ampliamente utilizado por su simplicidad, facilidad de implementación y utilidad en la enseñanza de los fundamentos del análisis numérico.

Se aplica para resolver problemas del tipo:

$$\frac{dy}{dx} = f(x, y), \quad y(x_0) = y_0$$

Buscamos estimar la solución $y(x)$ en un intervalo cercano a x_0 , utilizando incrementos pequeños de tamaño h .

Fórmula del método de Euler

Considerando el un punto (x_n, y_n) , el valor del siguiente punto se determina mediante la siguiente fórmula:

$$y_{n+1} = y_n + h \cdot f(x_n, y_n)$$

$$x_{n+1} = x_n + h$$

Esto se repite para obtener una secuencia de puntos aproximados de la solución.

4.1. Menú Presentación

Al iniciar SoftEDOEuler v1.0, los usuarios son recibidos con una pantalla de bienvenida que incluye un mensaje introductorio, el logotipo distintivo y el nombre del programa resaltado. Esta sección inicial tiene como propósito familiarizar al usuario con la interfaz y los objetivos principales del software, proporcionando una visión general de sus funcionalidades antes de comenzar con el proceso de cálculo.



4.2. Euler

El menú de "Euler" de *SoftEDOEuler v1.0* es la sección central de la aplicación, diseñada para interactuar con las ecuaciones diferenciales ordinarias de primer orden cuyo método Euler desarrollado en SoftEDOEuler.

Para que el **método de Euler** pueda resolver un problema de ecuaciones diferenciales ordinarias (EDO), se necesitan los siguientes datos esenciales:

1. La ecuación diferencial:

$$\frac{dy}{dx} = f(x, y)$$

Es decir, la función que define la derivada de y respecto a x .

2. **Condición inicial:**

Un punto inicial conocido donde se conoce el valor de la función:

$$y(x_0) = y_0$$

3. **Intervalo de solución:**

El intervalo de valores de x sobre el cual se busca obtener una aproximación de la solución.

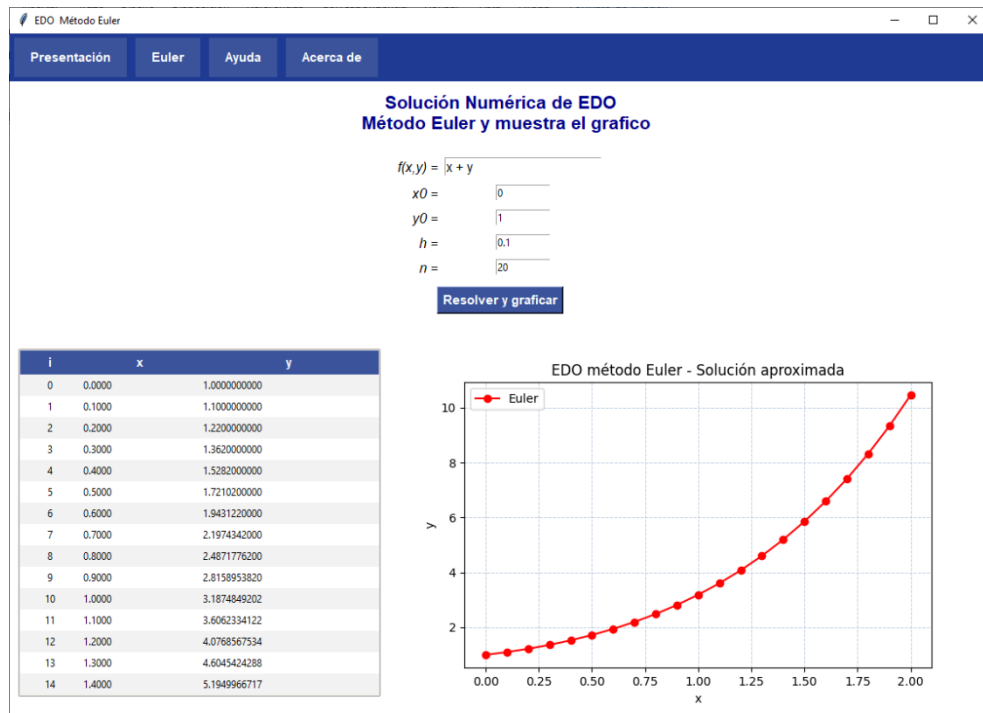
4. **Tamaño del paso h :**

El incremento o paso pequeño con el que se avanzará de x_n a x_{n+1} . Esto afecta la precisión y el número de pasos.

A continuación, mostramos la ventana de ingreso de datos:

The screenshot shows a web application titled "EDO Método Euler". It has a navigation bar with four tabs: "Presentación", "Euler", "Ayuda", and "Acerca de". The main content area is titled "Solución Numérica de EDO" and "Método Euler y muestra el grafico". Below the title, there is a form for inputting data. The form includes a text input for the function $f(x,y) = x + y$, and four labeled input fields: $x0 = 0$, $y0 = 1$, $h = 0.1$, and $n = 20$. A blue button labeled "Resolver y graficar" is positioned below the input fields.

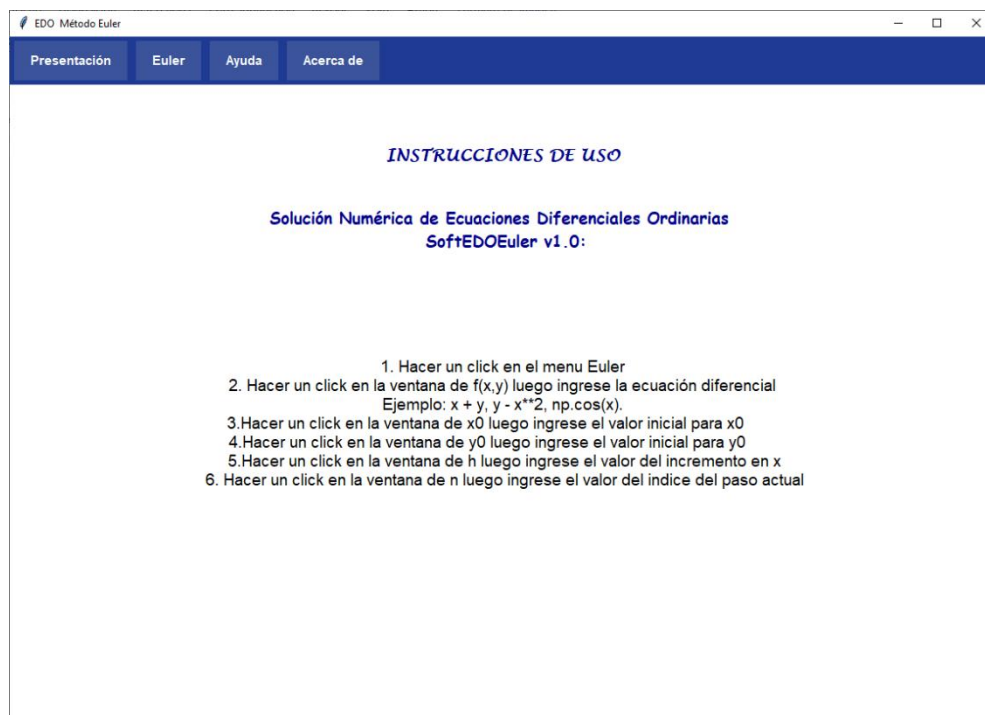
Luego de haber ingresado los datos respectivos hacemos clic en el botón Resolver y graficar, teniendo la solución respectiva en una tabla obteniéndose cada punto realizando iteraciones con el método Euler de ecuaciones diferenciales ordinarias y el grafico correspondiente.



4.3. Menú Ayuda

El menú "Ayuda" de SoftEDOEuler v1.0 ha sido creado como una herramienta accesible y funcional, especialmente pensada para apoyar a los estudiantes en la comprensión del software. Esta sección incluye explicaciones claras y precisas que facilitan tanto el aprendizaje como el uso correcto de la aplicación.

Su propósito principal es ofrecer una guía sencilla que permita a los usuarios familiarizarse rápidamente con las funciones esenciales del programa. Gracias a esto, los estudiantes pueden aprovechar al máximo las herramientas disponibles y utilizar el software de manera eficiente.



4.4. Menú Acerca de

En la sección "Acerca de", los usuarios pueden acceder a información relevante sobre SoftEDOEuler v1.0, una aplicación educativa diseñada para aproximar soluciones de ecuaciones diferenciales ordinarias (EDO) de primer orden mediante el método de Euler, especialmente útil cuando no es posible obtener una solución analítica exacta.

Esta sección del programa incluye datos importantes como el nombre completo del software, la versión actual, el desarrollador principal (identificado como Leonel Coyla Idme), la fecha de lanzamiento y los canales de contacto disponibles para brindar soporte técnico, recibir sugerencias o responder consultas.

El propósito de esta sección es ofrecer claridad sobre la autoría y el origen del programa, lo que contribuye a fortalecer la confianza de los usuarios al momento de utilizar la herramienta. Al proporcionar estos datos, se busca reforzar la transparencia del proyecto y demostrar su carácter confiable y educativo.

Además, este espacio establece un vínculo directo con el desarrollador, permitiendo que los usuarios puedan resolver inquietudes, enviar comentarios o realizar preguntas de manera sencilla. De este modo, estudiantes, docentes y familias pueden conocer con precisión el origen de la aplicación y asegurarse de que están utilizando una plataforma confiable, creada específicamente para apoyar procesos de aprendizaje.



5. PREGUNTAS FRECUENTES (FAQS) SOBRE EL SOFTWARE

5.1. ¿Qué es Método Euler en Ecuaciones diferenciales ordinarias y cuál es su propósito?

El método de Euler es una técnica numérica utilizada para aproximar soluciones de ecuaciones diferenciales ordinarias (EDO) de primer orden. Se basa en la idea de que, conociendo la pendiente de la solución en un punto (es decir, la derivada), se puede estimar el valor de la función en puntos cercanos, avanzando paso a paso.

El propósito del método de Euler es ofrecer una forma simple de aproximar soluciones de ecuaciones diferenciales ordinarias cuando no se puede encontrar una solución exacta de forma analítica. Es especialmente útil para:

- Visualizar el comportamiento de soluciones de ecuaciones diferenciales.
- Resolver problemas iniciales de manera rápida y sencilla.
- Introducir conceptos básicos de métodos numéricos y análisis de errores.

Aunque no es el método más preciso, su sencillez y facilidad de implementación lo convierten en una herramienta muy valiosa para fines educativos y para introducir a los estudiantes en los conceptos básicos del análisis numérico.

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

El aplicativo es compatible con múltiples sistemas operativos, incluyendo Windows (versiones 7, 8, 10 y 11, siendo recomendable Windows 10 o superior), macOS (a partir de la versión 10.14 Mojave) y diversas distribuciones de Linux como Ubuntu 18.04 o superior, Debian, Fedora, entre otras. En cuanto a los requisitos de hardware, se requiere un procesador Intel o AMD de al menos 1 GHz, un mínimo de 2 GB de memoria RAM (se recomienda 4 GB), al menos 100 MB de espacio libre en disco, y una pantalla con resolución mínima de 1024x768 píxeles. Para un funcionamiento óptimo, es necesario tener instalado Python 3.8 o superior, junto con Tkinter (incluido por defecto en la instalación oficial de Python). Además, se recomienda utilizar editores o entornos de desarrollo compatibles como VSCode, Thonny, IDLE u otros similares.

5.3. ¿Cómo instalo o configuro SoftEDOEuler?

El software SoftEDOEuler v1.0 está diseñado para funcionar óptimamente con una resolución mínima de 1024x768 píxeles, garantizando un rendimiento adecuado en la mayoría de las pantallas. Sin embargo, si el usuario prefiere una

resolución mayor, el tamaño de la ventana puede ajustarse de forma sencilla modificando el código fuente del programa.

Para realizar este cambio, se debe abrir el archivo `SoftEDOEuler.py` en un editor de texto y buscar la línea donde se establece el tamaño de la interfaz, que normalmente aparece como `selt.geometry("1150x800")`. A partir de allí, es posible personalizar las dimensiones de la ventana según las necesidades del usuario.

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

Si `SoftEDOEuler v1.0` presenta fallos durante su funcionamiento, lo primero que se debe verificar es que todas las bibliotecas necesarias, como `tkinter` y `pillow`, estén correctamente instaladas y que se utilice Python en su versión 3.8 o superior. Además, es esencial asegurarse de que no haya errores de sintaxis en la entrada de datos y que el archivo de imagen `SoftEDOEuler.png` se encuentre en la misma carpeta desde la cual se ejecuta el programa.

Si después de realizar estas verificaciones el problema persiste, se recomienda reiniciar tanto la aplicación como el sistema operativo. En caso de que los errores continúen, es aconsejable ponerse en contacto con el desarrollador principal a través del correo electrónico indicado en la sección "**Acerca de**" del programa (lcoyla@unap.edu.pe). Al hacerlo, es importante detallar el problema e incluir capturas de pantalla o mensajes de error que puedan facilitar la identificación y solución del inconveniente.

6. ADVERTENCIAS Y SUGERENCIAS DEL SOFTWARE DE APLICACIÓN.

6.1. Advertencias

Tener en cuenta que *softEDOEuler v1.0* ha sido desarrollado con un propósito estrictamente educativo. En este contexto, el método de Euler en los métodos numéricos tiene como objetivo proporcionar una forma aproximada y sencilla de resolver ecuaciones diferenciales ordinarias (EDO) de primer orden, especialmente cuando no es posible obtener una solución analítica.

Además, se recomienda no eliminar ni renombrar el archivo **SoftEDOEuler.png**, ya que esto podría afectar su correcta visualización en la

sección de presentación del software. Para asegurar un funcionamiento óptimo, es aconsejable ejecutar la aplicación en sistemas operativos compatibles y mantener actualizadas todas las bibliotecas necesarias.

6.2. Sugerencias

Se recomienda que los usuarios practiquen con ejemplos de ecuaciones diferenciales ordinarias de primer orden, ya que esta actividad no solo refuerza los conocimientos teóricos, sino que también promueve el desarrollo de habilidades prácticas.

La simplicidad del método de Euler lo convierte en una herramienta ideal para introducir a los usuarios en el análisis numérico, permitiendo visualizar cómo pequeñas variaciones en las condiciones iniciales o en el tamaño del paso afectan la precisión del resultado. En definitiva, *softEDOEuler v1.0* no solo permite experimentar con estos conceptos, sino también fomentar un aprendizaje activo y exploratorio.

7. Código

```
1 import tkinter as tk
2 from tkinter import ttk
3 import numpy as np
4 from matplotlib.figure import Figure
5 from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
6 import math
7
8 from PIL import ImageTk
9 from PIL import Image
10
11 class AppEuler(tk.Tk):
12     def __init__(self):
13         tk.Tk.__init__(self)
14         self.title("EDO Método Euler")
15         self.geometry("1150x800")
16
17         # Menú horizontal
18         self.menuFrame = tk.Frame(self, bg="#1F3A93", height=50)
19         self.menuFrame.pack(side="top", fill="x")
20
21         self.contenidoFrame = tk.Frame(self, bg="white")
22         self.contenidoFrame.pack(fill="both", expand=True)
23
24         for menu in ["Presentación", "Euler", "Ayuda", "Acerca de"]:
25             btn = tk.Button(self.menuFrame, text=menu, bg="#3A539B", fg="white",
26                             font=("Arial", 11, "bold"), bd=0, padx=15, pady=10,
27                             command=lambda n=menu: self.mostrarSeccion(n))
28             btn.pack(side="left", padx=5, pady=5)
29
30         self.mostrarSeccion("Presentación")
31
32     def borrarContenido(self):
33         for widget in self.contenidoFrame.winfo_children():
34             widget.destroy()
35
36     def mostrarSeccion(self, menu):
37         self.borrarContenido()
38         if menu == "Presentación":
39             tk.Label(self.contenidoFrame, text="", font=("Arial", 12, "bold"), fg="#00008B", bg="white").pack()
40             tk.Label(self.contenidoFrame, text="", font=("Arial", 12, "bold"), fg="#00008B", bg="white").pack()
41             tk.Label(self.contenidoFrame, text="BIENVENIDOS A LA APLICACIÓN",
42                     font=("Comic Sans MS", 18, "bold"), fg="#00008B", bg="white").pack()
43             tk.Label(self.contenidoFrame, text="Solución numérica de una Ecuación Diferencial Ordinaria\n"
44                     "Método Euler",
45                     font=("Arial", 14, "bold"), fg="#00008B", bg="white").pack()
46             tk.Label(self.contenidoFrame, text="", font=("Arial", 12, "bold"), fg="#00008B", bg="white").pack()
47             tk.Label(self.contenidoFrame, text="", font=("Arial", 12, "bold"), fg="#00008B", bg="white").pack()
48
49             try:
50                 image = Image.open("Euler.png")
51                 image = image.resize((500, 350))
52                 imageTk = ImageTk.PhotoImage(image)
53                 imageLabel = tk.Label(self.contenidoFrame, image=imageTk, bg="white")
54                 imageLabel.image = imageTk
55                 imageLabel.pack(pady=20)
56             except Exception as e:
57                 print(f"Error al cargar la imagen: {e}")
58
59             tk.Label(self.contenidoFrame, text="", font=("Arial", 12, "bold"), fg="#00008B", bg="white").pack()
60             tk.Label(self.contenidoFrame, text="", font=("Arial", 12, "bold"), fg="#00008B", bg="white").pack()
```



```

61 tk.Label(self.contenidoFrame, text="SoftEDO Euler v1.0", font=("Comic Sans MS", 20, "bold"),
62         fg="#00008B", bg="white").pack()
63
64
65 elif menu == "Euler":
66     self.mostrarEuler()
67
68 elif menu == "Ayuda":
69
70     tk.Label(self.contenidoFrame, text="\n\nINSTRUCCIONES DE USO", bg="white",
71             font=("Lucida Handwriting", 14, "bold"), fg="#00008B").pack(pady=20)
72     tk.Label(self.contenidoFrame, text="Solución Numérica de Ecuaciones Diferenciales Ordinarias"
73         "\n SoftEDO Euler v1.0:\n\n", bg="white",
74             font=("Comic Sans MS", 14, "bold"), fg="#00008B").pack(pady=20)
75     tk.Label(self.contenidoFrame, text="\n1. Hacer un click en el menu Euler"
76         f"\n2. Hacer un click en la ventana de f(x,y) luego ingrese la ecuación diferencial\n"
77         f"Ejemplo: x + y, y - x**2, np.cos(x).\n"
78         f"3.Hacer un clic en la ventana de x0 luego ingrese el valor inicial para x0      \n "
79         f"4.Hacer un clic en la ventana de y0 luego ingrese el valor inicial para y0      \n "
80         f"5.Hacer un clic en la ventana de h luego ingrese el valor del incremento en x\n "
81         f"6. Hacer un clic en la ventana de n luego ingrese el valor del índice del paso actual\n",
82         bg="white",
83         font=("Arial", 14)).pack(pady=20)
84
85 elif menu == "Acerca de":
86     tk.Label(self.contenidoFrame, text="\n\n\nSolución Numérica de Ecuaciones Diferenciales Ordina
87         "\nSoftEDO Euler v1.0",
88         bg="white", font=("Comic Sans MS", 14, "bold"),
89         fg="#00008B").pack(pady=(0, 10))
90     # Lista de autores
91     autores = [
92         "\nDesarrollado por: Leonel Coyla Idme",
93         "Elqui Yeye Pari Condori",
94         "Juan Reynaldo Paredes Quispe",
95         "José Pánfilo Tito Lipa",
96     ]
97
98     # Etiqueta por autor
99     for autor in autores:
100         tk.Label(self.contenidoFrame, text=autor, bg="white",
101                 font=("Comic Sans MS", 14)).pack(pady=2)
102
103     labelAcercade = tk.Label(self.contenidoFrame, text= "\n\nLanzamiento : agosto 2025",
104                             font=("Comic Sans MS", 14),fg="#003366",bg="white")
105     labelAcercade.pack(pady=(1,10))
106     labelAcercade = tk.Label(self.contenidoFrame, text= "Contacto: lcoyla@unap.edu.pe",
107                             font=("Comic Sans MS", 14),fg="#003366",bg="white")
108     labelAcercade.pack(pady=(1,10))
109
110 def mostrarEuler(self):
111     tk.Label(self.contenidoFrame, text="Solución Numérica de EDO\n Método Euler y muestra el grafico"
112         font=("Arial", 16, "bold"), bg="white",fg="#00008B").pack(pady=10)
113
114     form = tk.Frame(self.contenidoFrame, bg="white")
115     form.pack(pady=10)
116
117     # Ingreso de datos
118     tk.Label(form, text="f(x,y) =", font=("Arial", 12, "italic"),bg="white").grid(row=0, column=0, sticky="e")
119     ecuacion_entry = tk.Entry(form, width=20, font=("Arial", 12))
120     ecuacion_entry.grid(row=0, column=1, padx=5, pady=5)

```

```

121 ecuacion_entry.insert(0, "x + y")
122
123 tk.Label(form, text="x0 =", font=("Arial", 12, "italic"), bg="white").grid(row=1, column=0, sticky="e")
124 x0_entry = tk.Entry(form, width=10)
125 x0_entry.grid(row=1, column=1, padx=5, pady=5)
126 x0_entry.insert(0, "0")
127
128 tk.Label(form, text="y0 =", font=("Arial", 12, "italic"), bg="white").grid(row=2, column=0, sticky="e")
129 y0_entry = tk.Entry(form, width=10)
130 y0_entry.grid(row=2, column=1, padx=5, pady=5)
131 y0_entry.insert(0, "1")
132
133 tk.Label(form, text="h =", font=("Arial", 12, "italic"), bg="white").grid(row=3, column=0, sticky="e")
134 h_entry = tk.Entry(form, width=10)
135 h_entry.grid(row=3, column=1, padx=5, pady=5)
136 h_entry.insert(0, "0.1")
137
138 tk.Label(form, text="n =", font=("Arial", 12, "italic"), bg="white").grid(row=4, column=0, sticky="e")
139 n_entry = tk.Entry(form, width=10)
140 n_entry.grid(row=4, column=1, padx=5, pady=5)
141 n_entry.insert(0, "20")
142
143 resultadoFrame = tk.Frame(self.contenidoFrame, bg="white")
144 resultadoFrame.pack(fill="both", expand=True, pady=10)
145
146 def calcular():
147     try:
148         funcionCadena = ecuacion_entry.get()
149         x0 = float(x0_entry.get())
150         y0 = float(y0_entry.get())
151         h = float(h_entry.get())
152         n = int(n_entry.get())
153
154
155     def funcion(x, y):
156         variablesOperacion = {
157             "x": x,
158             "y": y,
159             "np": np,
160             "math": math
161         }
162         return eval(funcionCadena, variablesOperacion)
163
164     # Método de Euler
165     xsgte = [x0]
166     ysgte = [y0]
167     for _ in range(n):
168         ynuevo = ysgte[-1] + h * funcion(xsgte[-1], ysgte[-1])
169         xnuevo = xsgte[-1] + h
170         xsgte.append(xnuevo)
171         ysgte.append(ynuevo)
172
173     for w in resultadoFrame.winfo_children():
174         w.destroy()
175
176     # Mostrar tabla
177     columnas = ("i", "x", "y")
178     tree = ttk.Treeview(resultadoFrame, columns=columnas, show="headings", height=15)
179
180     # Configura el encabezado

```

```

181     for col in columnas:
182         tree.heading(col, text=col)
183
184     # Ajuste de ancho de columna (persona)
185     tree.column("i", width=70, anchor="center")
186     tree.column("x", width=140, anchor="w")
187     tree.column("y", width=210, anchor="w")
188
189
190     style = ttk.Style()
191     style.theme_use("clam") # boton estilizado
192
193     # Encabezado de la tabla
194     style.configure(
195         "Treeview.Hheading",
196         font=("Arial", 11, "bold"),
197         background="#3A539B", # azul oscuro medio
198         foreground="white", #Color de texto
199         relief="flat"
200     )
201
202     # Filas de la tabla
203     style.configure(
204         "Treeview",
205         background="#F5F5F5",
206         foreground="black",
207         rowheight=25,
208         fieldbackground="#F5F5F5"
209     )
210
211     # Efecto hover puntero en barra o en filas
212     style.map(
213         "Treeview",
214         background=[("selected", "#81C784")]
215     )
216
217     # Definir colores como variables
218     colorFilaImpar = "#f2f2f2" #Color gris claro
219     colorFilaPar = "white"
220     colorSeleccion = "#FFB347" #Color naranja de la barra
221
222     # Aplicar estilos
223     style.map("Treeview", background=[("selected", colorSeleccion)])
224     tree.tag_configure("oddrow", background=colorFilaImpar)
225     tree.tag_configure("evenrow", background=colorFilaPar)
226
227     # Encabezados
228     for col in columnas:
229         tree.heading(col, text=col)
230
231     for i in range(len(xsgte)):
232         xx = xsgte[i]
233         yy = ysgte[i]
234
235         if i % 2 == 0:
236             tag = "oddrow" # Fila impar
237         else:
238             tag = "evenrow"
239
240         valores = (i, f"{xx:.4f}", f"{yy:.10f}")

```

```

241         tree.insert("", "end", values=valores, tags=(tag,))
242
243     tree.pack(side="left", padx=10, pady=5)
244
245     # Gráfica con matplotlib
246     figura = Figure(figsize=(5, 4), dpi=100)
247     eje = figura.add_subplot(111)
248     eje.plot(xsgte, ysgte, marker='o', color="red", label="Euler")
249     eje.set_title("EDO método Euler - Solución aproximada")
250     eje.set_xlabel("x")
251     eje.set_ylabel("y")
252     eje.grid(
253         True,
254         which='major',
255         color='#B0C4DE', # Color azul suave
256         linestyle='--',
257         linewidth=0.7,
258         alpha=0.8
259     )
260     eje.legend()
261
262     canvas = FigureCanvasTkAgg(figura, master=resultadoFrame)
263     canvas.draw()
264     canvas.get_tk_widget().pack(side="right", fill="both", expand=True)
265
266     except Exception as e:
267         for w in resultadoFrame.winfo_children():
268             w.destroy()
269         tk.Label(resultadoFrame, text=f"Error: {e}", fg="red", bg="white").pack()
270
271     tk.Button(form, text="Resolver y graficar", command=calcular,
272             bg="#3A539B", fg="white", font=("Arial", 11, "bold")).grid(row=5, column=0,
273             colspan=2, pady=8)
274
275 def main():
276     app = AppEuler()
277     app.mainloop()
278 if __name__ == "__main__":
279     main()
280

```