

# Graficador de funciones

Henry Ccoarite Dueñas

## Descripción

Permite graficar dos funciones ingresadas por el usuario en la variable  $x$ . El programa permite graficar en un mismo sistema de coordenadas dos funciones matemáticas en la variable  $x$ , representando cada una con un color distinto para diferenciarlas visualmente y mostrando de manera clara los ejes cartesianos con sus divisiones numéricas. Además de la representación gráfica, ofrece un panel lateral donde se listan los pares ordenados  $(x, y)$  generados a partir de un valor de paso definido por el usuario, lo que facilita observar cómo cambian los resultados numéricos a medida que varía  $x$ . Gracias a esta funcionalidad, el sistema no solo permite analizar la forma de las curvas, sino también verificar valores puntuales de las funciones. Asimismo, al aceptar expresiones que incluyen operaciones básicas y funciones de la librería `math` como `sin`, `cos` o `exp`, el graficador puede trabajar con una amplia variedad de expresiones, desde ecuaciones lineales hasta trigonométricas y exponenciales.

### Entrada

- Dos expresiones en la variable  $x$  (ej.:  $2x+3$ ,  $-x+2$ ).
- Un entero que indica el paso de muestreo.

### Salida

- Gráfica de las dos funciones en el mismo sistema de coordenadas.
- Listado de los pares  $(x, y)$  calculados en el panel lateral.

### Restricciones

- El muestreo de  $x$  usado en el código está acotado al rango  $[-200, 200]$ .
- Las expresiones se evalúan con un entorno controlado que incluye `math` y bloquea `__builtins__`.

## Código Python

```
1 import tkinter as tk
2 import math
3
4 def evaluar(expr: str, x: float):
5     """Eval a una expresi n matem tica en x usando math."""
6     try:
```

```

7         return eval(expr, {"x": x, "math": math, "__builtins__":
8             {}})
9     except Exception:
10         return None
11
12 def dibujar_ejes(canvas, ancho, alto, escala):
13     """Dibuja los ejes cartesianos en el canvas."""
14     cx, cy = ancho // 2, alto // 2
15     canvas.create_line(0, cy, ancho, cy, fill="black")
16     canvas.create_line(cx, 0, cx, alto, fill="black")
17
18     for i in range(-ancho // (2 * escala), ancho // (2 * escala) +
19         1):
20         x = cx + i * escala
21         canvas.create_line(x, cy - 3, x, cy + 3, fill="black")
22         if i != 0:
23             canvas.create_text(x, cy + 12, text=str(i), font=("Arial", 7))
24
25     for j in range(-alto // (2 * escala), alto // (2 * escala) + 1):
26         y = cy - j * escala
27         canvas.create_line(cx - 3, y, cx + 3, y, fill="black")
28         if j != 0:
29             canvas.create_text(cx - 12, y, text=str(j), font=("Arial", 7))
30
31 def trazar_funcion(canvas, salida, expr, color, ancho, alto, escala,
32     paso):
33     """Traza una funci n en el lienzo y lista sus valores en el
34         cuadro de texto."""
35     cx, cy = ancho // 2, alto // 2
36     puntos = []
37     salida.insert(tk.END, f"\nValores de {expr}:\n")
38
39     for x in range(-200, 201, paso):
40         x_real = x / 10
41         y_real = evaluar(expr, x_real)
42         if y_real is not None:
43             px = cx + x_real * escala
44             py = cy - y_real * escala
45             puntos.append((px, py))
46             salida.insert(tk.END, f"x={x_real:.1f}, y={y_real:.2f}\n")
47
48     if len(puntos) > 1:
49         canvas.create_line(puntos, fill=color, width=2, smooth=True)
50
51 def graficar():

```

```
48     """Borra el canvas, dibuja ejes y grafica las funciones
        ingresadas."""
49     canvas.delete("all")
50     salida.delete("1.0", tk.END)
51
52     ancho, alto = 800, 600
53     escala = 30
54     paso = int(entrada_paso.get())
55
56     dibujar_ejes(canvas, ancho, alto, escala)
57
58     f1, f2 = entrada1.get(), entrada2.get()
59     trazar_funcion(canvas, salida, f1, "red", ancho, alto, escala,
        paso)
60     trazar_funcion(canvas, salida, f2, "blue", ancho, alto, escala,
        paso)
61
62     root = tk.Tk()
63     root.title("Graficador de Funciones")
64
65     panel = tk.Frame(root, padx=10, pady=10)
66     panel.pack()
67
68     tk.Label(panel, text="Funci n 1 (ej: 2*x+3)").grid(row=0, column=0,
        sticky="w")
69     entrada1 = tk.Entry(panel, width=25)
70     entrada1.insert(0, "2*x+3")
71     entrada1.grid(row=0, column=1, padx=5)
72
73     tk.Label(panel, text="Funci n 2 (ej: -x+2)").grid(row=1, column=0,
        sticky="w")
74     entrada2 = tk.Entry(panel, width=25)
75     entrada2.insert(0, "-x+2")
76     entrada2.grid(row=1, column=1, padx=5)
77
78     tk.Label(panel, text="Paso (ej: 5)").grid(row=2, column=0, sticky="w")
79     entrada_paso = tk.Entry(panel, width=10)
80     entrada_paso.insert(0, "5")
81     entrada_paso.grid(row=2, column=1, sticky="w")
82
83     tk.Button(panel, text="Graficar", command=graficar).grid(row=3,
        column=0, columnspan=2, pady=8)
84
85     canvas = tk.Canvas(root, width=800, height=600, bg="white")
86     canvas.pack(side="left")
87
88     salida = tk.Text(root, width=40, height=35)
```

```
89 salida.pack(side="right")
90
91 root.mainloop()
```

Listing 1: Graficador de funciones con Tkinter

## Ejemplo de uso

Funciones:  $2x+3$  ,  $-x+2$  Paso: 5

Salida esperada:

- Ambas funciones dibujadas en el plano cartesiano.
- Lista de pares  $(x, y)$  en el panel lateral para cada función.

