



Ingeniería en Sistemas Computacionales
PROGRAMACION ORIENTADA A OBJETOS
Oscar David Galvan Alvarez
TUE0096

INVESTIGACION #3 GRAFICOS

En este documento se verá información acerca del tema de gráficos más precisamente de 2 el cual es el spider web que pertenece a la librería matplotlib que genera un gráfico que muestra hacia donde tienden ciertos tipos de datos, y el segundo que es tkinter que se utiliza para interfaces de usuario.

GRAFICO SPIDER WEB (LIBRERÍA: MATPLOTLIB)

Un gráfico spider web es una representación visual de datos multivariados en forma de polígono radial donde cada eje representa una variable y los datos se conectan formando una figura cerrada esto puede ser muy útil a la hora de comparar múltiples atributos de alguna zona o área de trabajo ya que esto puede ayudar a ver si hay debilidades y en donde se tendría que poner atención para mejorar, un ejemplo mas de esto puede ser el rendimiento de los jugadores de algún deporte.

En el caso de este grafico la librería que se utiliza es la de matplotlib y sus características principales son:

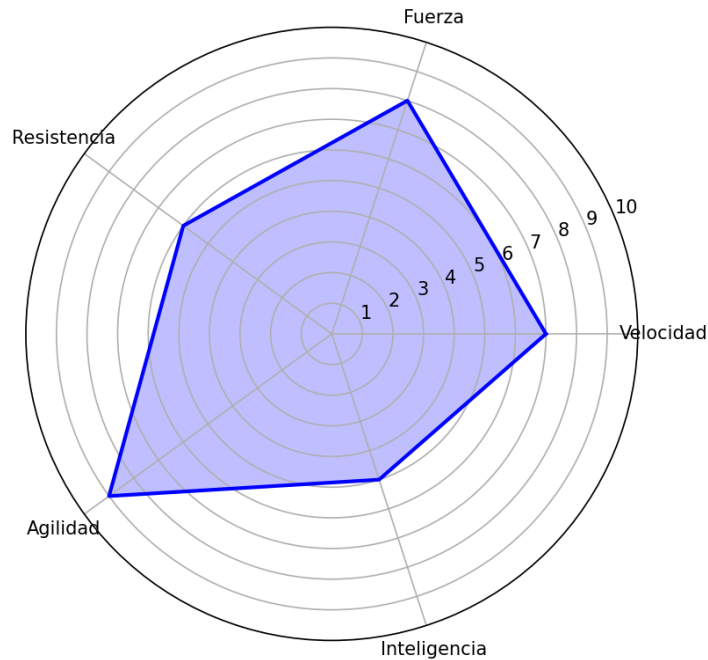
- **Ejes radiales:** Cada eje representa una métrica o categoría.
- **Escala:** Los valores se trazan desde el centro hacia el borde del gráfico.
- **Comparación:** Es útil para comparar múltiples conjuntos de datos (por ejemplo, el rendimiento de diferentes productos o personas).

```
CLASES (ATRIBUTOS Y METODOS) HERENCIA Y MODULARIDAD_DAVID_ALVAREZ - import numpy as np.py
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # Datos de ejemplo
5 labels = ['Velocidad', 'Fuerza', 'Resistencia', 'Agilidad', 'Inteligencia']
6 valores = [7, 8, 6, 9, 5]
7
8 # Configuración del gráfico
9 angulos = np.linspace(0, 2 * np.pi, len(labels), endpoint=False).tolist()
10 valores += valores[:1] # Cerrar el polígono
11 angulos += angulos[:1]
12
13 # Crear el gráfico
14 fig, ax = plt.subplots(figsize=(6, 6), subplot_kw=dict(polar=True))
15 ax.fill(angulos, valores, color='blue', alpha=0.25)
16 ax.plot(angulos, valores, color='blue', linewidth=2)
17 ax.set_yticks(range(1, 11)) # Escala
18 ax.set_xticks(angulos[:-1])
19 ax.set_xticklabels(labels)
20
21 plt.show()
```

Aquí un ejemplo en código de un grafico spider web en matplotlib en total ocupamos 2 librerías las cuales son matplotlib y numpy, después definimos los datos de ejemplo en listas para utilizarlos directamente para así graficarlos.

Luego de esto empezaremos a definir los rangos del gráfico, como lo son los ángulos, para luego de esto ahora si llamar las funciones para graficarlo.

El resultado es el siguiente:



TKINTER INTERFACES DE USUARIO

Tkinter es la librería estándar de Python para la creación de interfaces de usuario conocidas como GUI que es por graphical user interfaz esta misma en varias ocasiones viene instalada con Python mismo por lo que no se necesitaría instalarla.

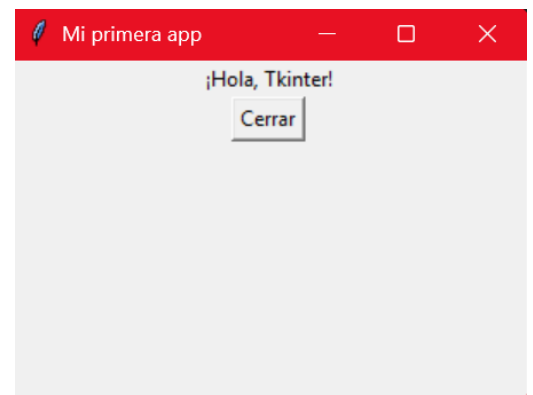
Ahora bien, ¿que puedes hacer con esta librería?, como tal puedes construir aplicaciones visuales y estas mismas pueden tener elementos como lo son:

- Mostrar texto.
- Ejecutar acciones al hacer clic.
- Campos de texto para entrada de usuario.
- Área de texto multilínea.
- Contenedor para organizar widgets.
- Etc.

Esta librería funciona como enlace el cual esta creando objetos en Python que genera comandos Tcl/Tk, estos comandos se ejecutan en un interprete Tcl que maneja la interfaz gráfica.

A continuación, un ejemplo de Tkinter:

```
CLASES (ATRIBUTOS Y METODOS) HERENCIA Y MODULARIDAD_DAVID_ALVAREZ - Untitled-1.ipynb
1 import tkinter as tk
2
3 ventana = tk.Tk()
4 ventana.title("Mi primera app")
5 ventana.geometry("300x200")
6
7 etiqueta = tk.Label(ventana, text="¡Hola, Tkinter!")
8 etiqueta.pack()
9
10 boton = tk.Button(ventana, text="Cerrar", command=ventana.destroy)
11 boton.pack()
12
13 ventana.mainloop()
```



En el ejemplo anterior se muestra una ventana en la cual muestra un texto, y un botón que dice cerrar el cual al dar clic cerrara la ventana y el código terminara de ejecutarse. Y con esto podremos hacer muchas más interfaces y aplicaciones probando con la configuración que nos ofrece.

CONCLUSION:

En esta investigación descubrí que hay una forma mas interesante de utilizar gráficos en Python como el spiderweb que puede ser utilizado en videojuegos para estadísticas de personajes u otro tipo de valores para que así el jugador pueda verlas y elegir con que clase de personaje le interesa más jugar así como también en el caso de tkinter el como utilizar la forma de graficar para el diseño de aplicaciones o programas interactivos, aprendiendo cada vez mas sobre las opciones que están disponibles en el mercado hoy en día y que están a disposición de cualquier persona que esté interesada en descubrirlas. Espero ver estos temas en las siguientes clases para así poder profundizar un poco mas y practicar estos temas que es lo importante.

GITHUB: <https://github.com/D4V1D216/PROGRAMACION-ORIENTADA-A-OBJETOS.git>

BIBLIOGRAFIAS:

[Radar chart \(aka spider or star chart\) — Matplotlib 3.10.3 documentation](#)

<https://www.areasyvolumenes.net/radio-de-un-poligono/>

[matplotlib - How to make a polygon radar \(spider\) chart in python - Stack Overflow](#)