



Sesión 7

Nicolás Urbina Zermeño-A00575077

Pensamiento computacional para ingeniería

Camilo René Duque Becerra

Grupo 101

ITESM León

03 de del 2025

```

1  import tkinter as tk
2  from tkinter import ttk, messagebox
3  import requests
4  import matplotlib.pyplot as plt
5  from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
6
7
8  def fetch_data():
9      """
10     Conecta con la API de Open-Meteo y obtiene temperaturas horarias
11     de Berna, Suiza (últimas 24 horas).
12     Devuelve dos listas: horas y temperaturas.
13     """
14     try:
15         url = (
16             "https://api.open-meteo.com/v1/forecast"
17             "?latitude=46.94800&longitude=7.44780"
18             "&hourly=windspeed_10m&past_days=1"
19             "&timezone=auto"
20         )
21         response = requests.get(url, timeout=15)
22         response.raise_for_status()
23         data = response.json()

```

```

24
25         horas = data["hourly"]["time"]
26         temperaturas = data["hourly"]["windspeed_10m"]
27
28         return horas, temperaturas
29     except Exception as e:
30         messagebox.showerror("Error", f"No se pudieron obtener los datos:\\n{e}")
31         return [], []
32
33
34 def create_line_chart(horas, temps):
35     """Gráfica de línea."""
36     fig, ax = plt.subplots(figsize=(6, 3))
37     ax.plot(horas, temps, linestyle="-", marker="o", markersize=3)
38     ax.set_title("Velocidad del viento de Berna, Suiza(línea)")
39     ax.set_xlabel("Hora")
40     ax.set_ylabel("m/s")
41     ax.tick_params(axis="x", rotation=45)
42     fig.tight_layout()
43     return fig
44
45

```

```

46 def create_bar_chart(horas, temps):
47     """Gráfica de barras."""
48     fig, ax = plt.subplots(figsize=(6, 3))
49     ax.bar(horas, temps)
50     ax.set_title("Velocidad del viento en Berna, Suiza (barras)")
51     ax.set_xlabel("Hora")
52     ax.set_ylabel("m/s")
53     ax.tick_params(axis="x", rotation=45)
54     fig.tight_layout()
55     return fig
56
57
58 def mostrar_graficas(frm, horas, temps):
59     """Inserta las tres gráficas en el frame de la ventana tkinter."""
60     # Línea
61     fig1 = create_line_chart(horas, temps)
62     canvas1 = FigureCanvasTkAgg(fig1, master=frm)
63     canvas1.draw()
64     canvas1.get_tk_widget().pack(side="top", fill="y")

```

```

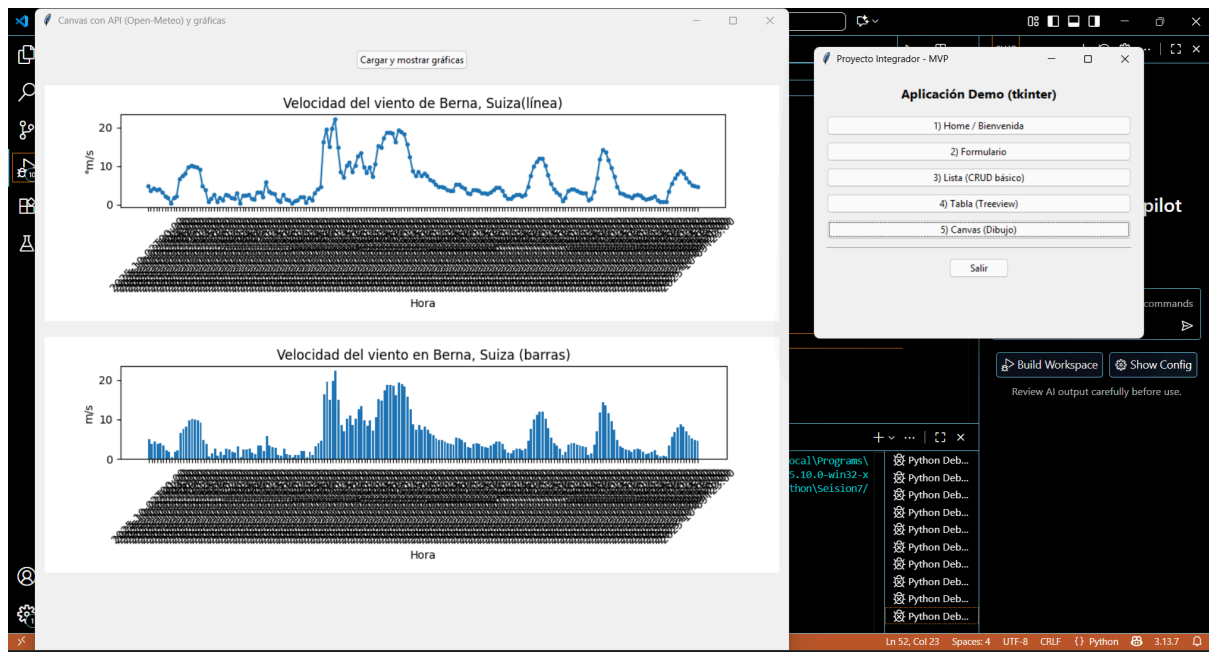
63     canvas1.draw()
64     canvas1.get_tk_widget().pack(pady=10, fill="x")
65
66     # Barras
67     fig2 = create_bar_chart(horas, temps)
68     canvas2 = FigureCanvasTkAgg(fig2, master=frm)
69     canvas2.draw()
70     canvas2.get_tk_widget().pack(pady=10, fill="x")
71
72
73 def open_win_canvas(parent: tk.Tk):
74     """
75     Crea la ventana secundaria con gráficas de la API.
76     """
77     win = tk.Toplevel(parent)
78     win.title("Canvas con API (Open-Meteo) y gráficas")
79     win.geometry("960x1000")
80
81     frm = ttk.Frame(win, padding=12)
82     frm.pack(fill="both", expand=True)
83
84     # Botón para cargar datos y graficar
85     def cargar():

```

```

82         frm.pack(fill="both", expand=True)
83
84     # Botón para cargar datos y graficar
85     def cargar():
86         horas, temps = fetch_data()
87         if horas and temps:
88             mostrar_graficas(frm, horas, temps)
89
90     ttk.Button(frm, text="Cargar y mostrar gráficas", command=cargar).pack(pady=10)
91
92
93 # Para pruebas independientes (opcional)
94 if __name__ == "__main__":
95     root = tk.Tk()
96     root.title("Prueba win_canvas")
97     ttk.Button(root, text="Abrir ventana Canvas", command=lambda: open_win_canvas(root)).pack()
98     root.mainloop()

```



- Cambios que se realizaron en mi código

Se actualizó el código para trabajar con la ciudad de Berna, Suiza, modificando las coordenadas de latitud y longitud a 46.94894 y 7.44540 en la URL; además de la temperatura, ahora se incluyen los parámetros de la velocidad del viento en la consulta, y se incorporó una rejilla punteada y semitransparente para mejorar la lectura del gráfico.