import tkinter as tk

from tkinter import ttk, messagebox

import requests

import matplotlib.pyplot as plt

from matplotlib.backends.backend\_tkagg import FigureCanvasTkAgg

**def** fetch\_data():

"""

Conecta con la API de Open-Meteo y obtiene temperaturas horarias

de Bayamón, Puerto Rico (últimas 24 horas).

Devuelve dos listas: horas y temperaturas.

"""

try:

url = (

"https://api.open-meteo.com/v1/forecast"

"?latitude=18.40&longitude=-66.18"

"&hourly=relativehumidity\_2m&past\_days=1"

"&timezone=auto"

)

response = requests.get(url, timeout=15)

response.raise\_for\_status()

data = response.json()

horas = data["hourly"]["time"]

temperaturas = data["hourly"]["relativehumidity\_2m"]

return horas, temperaturas

except Exception as e:

messagebox.showerror("Error", **f**"No se pudieron obtener los datos:\\n{e}")

return [], []

**def** create\_line\_chart(horas, temps):

"""Gráfica de línea."""

fig, ax = plt.subplots(figsize=(4, 2))

ax.plot(horas, temps, linestyle="-", marker="\*", markersize=5)

ax.set\_title("Humedad Relativa en Bayamón, Puerto Rico (línea)")

ax.set\_xlabel("Hora")

ax.set\_ylabel("°C")

ax.tick\_params(axis="x", rotation=90)

fig.tight\_layout()

return fig

**def** create\_bar\_chart(horas, temps):

"""Gráfica de barras."""

fig, ax = plt.subplots(figsize=(4, 2))

ax.bar(horas, temps)

ax.set\_title("Humedad Relativa en Bayamón, Puerto Rico (barras)")

ax.set\_xlabel("Hora")

ax.set\_ylabel("°C")

ax.tick\_params(axis="x", rotation=90)

fig.tight\_layout()

return fig

**def** mostrar\_graficas(frm, horas, temps):

"""Inserta las tres gráficas en el frame de la ventana tkinter."""

*# Línea*

fig1 = create\_line\_chart(horas, temps)

canvas1 = FigureCanvasTkAgg(fig1, master=frm)

canvas1.draw()

canvas1.get\_tk\_widget().pack(pady=15, fill="x")

*# Barras*

fig2 = create\_bar\_chart(horas, temps)

canvas2 = FigureCanvasTkAgg(fig2, master=frm)

canvas2.draw()

canvas2.get\_tk\_widget().pack(pady=15, fill="x")

**def** open\_win\_canvas(parent: tk.Tk):

"""

Crea la ventana secundaria con gráficas de la API.

"""

win = tk.Toplevel(parent)

win.title("Canvas con API (Open-Meteo) y gráficas")

win.geometry("960x1500")

frm = ttk.Frame(win, padding=22)

frm.pack(fill="both", expand=True)

*# Botón para cargar datos y graficar*

**def** cargar():

horas, temps = fetch\_data()

if horas and temps:

mostrar\_graficas(frm, horas, temps)

ttk.Button(frm, text="Cargar y mostrar gráficas", command=cargar).pack(pady=15)

*# Para pruebas independientes (opcional)*

if \_\_name\_\_ == "\_\_main\_\_":

root = tk.Tk()

root.title("Prueba win\_canvas")

ttk.Button(root, text="Abrir ventana Canvas", command=**lambda**: open\_win\_canvas(root)).pack(pady=20)

root.mainloop()