



## Gráficas

## Código:

```
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 humedad relativa horaria
    de Chicago, IL (últimas 24 horas).
    Devuelve dos listas: horas y humedad (%).
    """
    try:
        url = (
            "https://api.open-meteo.com/v1/forecast"
            "?latitude=41.85&longitude=-87.65"
            "&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=(6, 3))
    ax.plot(horas, temps, linestyle="-", marker="s", markersize=4,
linewidth=2, alpha=0.7)

    ax.set_title("Humedad relativa en Chicago (línea)")
    ax.set_xlabel("Hora")
    ax.set_ylabel("%")
    ax.tick_params(axis="x", rotation=45)
    ax.grid(True, linestyle="--", alpha=0.5)
    fig.tight_layout()
    return fig

def create_bar_chart(horas, temps):
    """Gráfica de barras."""
    fig, ax = plt.subplots(figsize=(6, 3))
    ax.bar(horas, temps)
    ax.set_title("Humedad relativa en Chicago (barras)")
    ax.set_xlabel("Hora")
    ax.set_ylabel("%")
    ax.tick_params(axis="x", rotation=45)
    ax.grid(True, linestyle="--", alpha=0.5)
    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=10, fill="x")

# Barras
fig2 = create_bar_chart(horas, temps)
canvas2 = FigureCanvasTkAgg(fig2, master=frm)
canvas2.draw()
canvas2.get_tk_widget().pack(pady=10, 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("960x1000")

    frm = ttk.Frame(win, padding=12)
    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=10)

# 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()
```

Luis Gabriel Aranda Torres

Explicación: Lo que hice fue cambiar la ciudad de León a Chicago, poniendo sus coordenadas (41.85, -87.65) en la URL de la API. Así ya no me trae los datos de León, sino los de Chicago.

También cambié el parámetro hourly: antes estaba usando temperature\_2m (temperatura en °C), y ahora puse relativehumidity\_2m, que es la humedad relativa en %. Por eso tuve que ajustar el código en la parte donde lee los datos, para que en lugar de temperature\_2m ahora agarre relativehumidity\_2m.

En las gráficas hice algunos detalles para que se vean mejor: le puse marcadores cuadrados a la línea, le subí un poco el grosor y le agregué transparencia. También activé la rejilla para que sea más fácil ver los valores.

Y por último, cambié los títulos y el eje Y: antes decían "Temperatura en León (°C)", ahora ya dicen "Humedad relativa en Chicago (%)", que es lo que en realidad estoy graficando.