

Osmar Alejandro López Gómez

A00574991

```
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 Cancún, Quintana Roo (últimas 24 horas).
    Devuelve dos listas: horas y temperaturas.
    """
    try:
        url = (
            "https://api.open-meteo.com/v1/forecast"
            "?latitude=21.15&longitude=-86.85"
            "&hourly=temperature_2m,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"]["temperature_2m"]
        humedades = data["hourly"]["relativehumidity_2m"]

        return horas, temperaturas, humedades
    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="o", markersize=3, color="green")
    ax.set_title("Temperatura en Cancún (línea)")
    ax.set_xlabel("Hora")
    ax.set_ylabel("°C")
    ax.tick_params(axis="x", rotation=45)
    fig.tight_layout()
    return fig

```

```

def create_bar_chart(horas, humedades):
    """Gráfica de barras de Humedad relativa."""
    fig, ax = plt.subplots(figsize=(6, 3))
    ax.bar(horas, humedades, color="orange")
    ax.set_title("Humedad en Cancún (barras)")
    ax.set_xlabel("Hora")
    ax.set_ylabel("% de Humedad")
    ax.tick_params(axis="x", rotation=45)
    fig.tight_layout()
    return fig

```

```

def mostrar_graficas(frm, horas, temps, humedades):
    """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, humedades)
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)

```

```

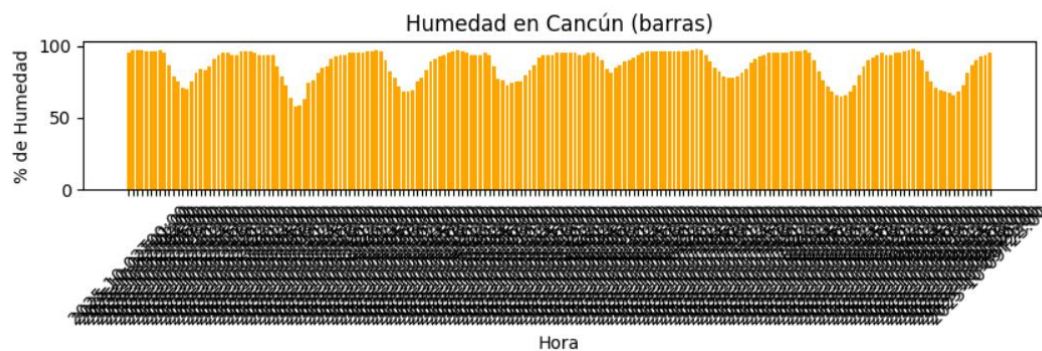
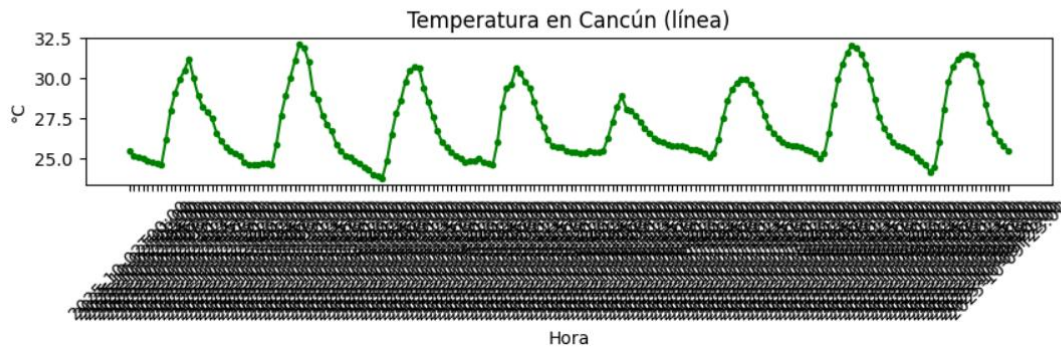
# Botón para cargar datos y graficar
def cargar():
    horas, temps, humedades = fetch_data()
    if horas and temps and humedades:
        mostrar_graficas(frm, horas, temps, humedades)

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()

```

Cargar y mostrar gráficas



Explicación de los cambios en el código:

Primero se modificó un poco la URL para que se tomaran los valores de la latitud y longitud de Cancún Quintana Roo, también se tomara en cuenta los valores de la humedad y se le agregó una variable llamada humedad.

Nueva latitud: `latitude=21.15&longitude=-86.85`

Después se modificó la gráfica de barras para usar la humedad en lugar de temperatura y se ajustó la fusión para que mostrara los datos de la gráfica de barras de color naranja.

Finalmente se cambió el título de las gráficas.