



Sesión 7 (dirigida): API pública + persistencia + gráficas

José Emilio Ibarra Aguirre- A00573821

Camilo Duque

Grupo 401

ITESM León

3 de Octubre del 2025

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 velocidad del viento horaria
    de Hong Kong (últimas 24 horas).
    Devuelve dos listas: horas y velocidades.
    """
    try:
        url = (
            "https://api.open-meteo.com/v1/forecast"
            "?latitude=70.3193&longitude=114.1694"
            "&hourly=windspeed_10m&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"]["windspeed_10m"] # reutilizamos el nombre de variable

        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="o", markersize=3)
    ax.set_title("Viento en Hong Kong (línea)")
    ax.set_xlabel("Hora")
    ax.set_ylabel("m/s")
    ax.tick_params(axis="x", rotation=45)
    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("Viento en Hong Kong (barras)")
    ax.set_xlabel("Hora")
    ax.set_ylabel("m/s")
    ax.tick_params(axis="x", rotation=45)
    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)

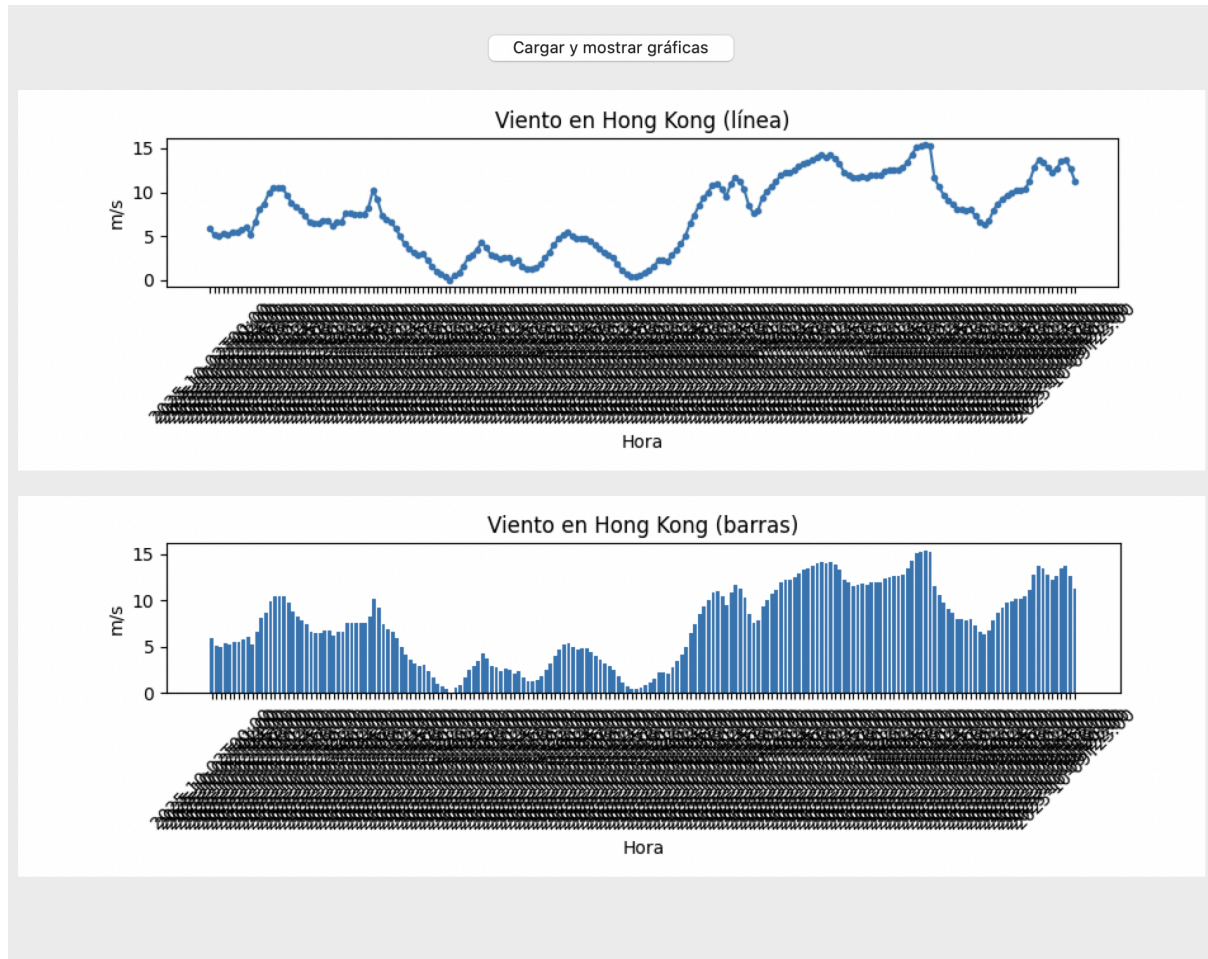
    # Botón para cargar datos y graficar
    def cargar():
        horas, temps = fetch_data()
        if horas and temps:
            mostrar_graficas(frm, horas, temps)

    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()

```

Captura de pantalla



Explicación

Cambié el código para pasar de temperatura en León a viento en Hong Kong. Primero, ajusté las coordenadas en la URL: de 21.12,-101.68 (León) a 22.3193,114.1694 (Hong Kong). Segundo, modifiqué el parámetro de la API: ya no pedimos temperature_2m, ahora pedimos windspeed_10m. Tercero, actualicé el docstring: antes hablaba de “temperaturas de León”; ahora dice “velocidad del viento de Hong Kong”. Cuarto, en el JSON ya no leo ["temperature_2m"] sino ["windspeed_10m"]. (Mantuve los nombres de variables temperaturas y temps para no romper la estructura; ahora guardan velocidad del viento).

Quinto, cambié los títulos de las gráficas: de “Temperatura en León (línea/barras)” a “Viento en Hong Kong (línea/barras)”. Sexto, ajusté la unidad del eje Y: de °C a m/s. Todo lo demás quedó igual (funciones, formato, ventana y botón).