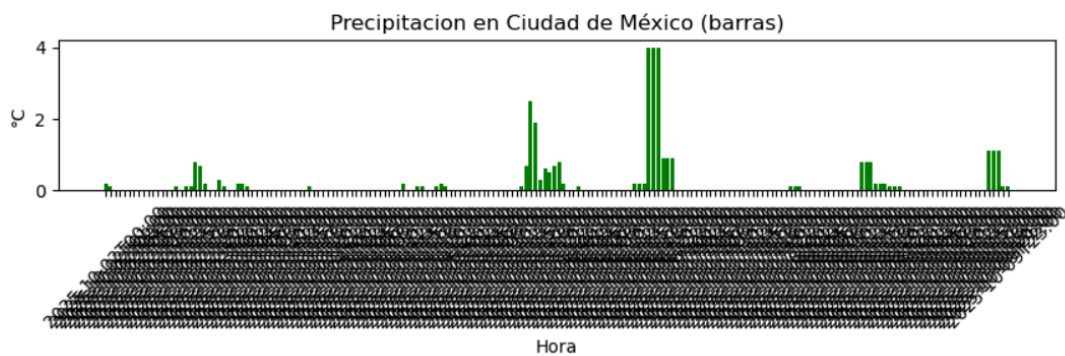
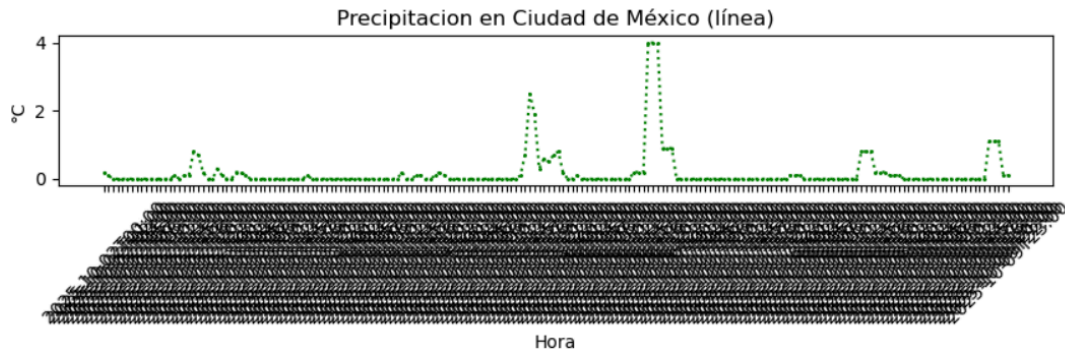


Julián Díaz Mojica

A00575197

Canvas con API (Open-Meteo) y gráficas

Cargar y mostrar gráficas



```
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 León, Gto (últimas 24 horas).
    Devuelve dos listas: horas y temperaturas.
    """
    try:
        #En esta URL cambié los valores de latitud y longitud por los
        valores de Ciudad de México
        url = (
```

```

"https://api.open-meteo.com/v1/forecast?latitude=19.43&longitude=-99.13
&hourly=precipitation&past_days=1&timezone=auto"
    )
    response = requests.get(url, timeout=15)
    response.raise_for_status()
    data = response.json()

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

    return horas, precipitation
except Exception as e:
    messagebox.showerror("Error", f"No se pudieron obtener los
datos:\n{e}")
    return [], []

def create_line_chart(horas, prec):
    """Gráfica de línea."""
    fig, ax = plt.subplots(figsize=(6, 3))
    ax.plot(horas, prec, linestyle=":", marker="D", markersize=1,
color= "green")
    ax.set_title("Precipitacion en Ciudad de México (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, prec):
    """Gráfica de barras."""
    fig, ax = plt.subplots(figsize=(6, 3))
    ax.bar(horas, prec, color = "green")
    ax.set_title("Precipitacion en Ciudad de México (barras)")
    ax.set_xlabel("Hora")
    ax.set_ylabel("°C")
    ax.tick_params(axis="x", rotation=45)
    fig.tight_layout()
    return fig

```

```

def mostrar_graficas(frm, horas, prec):
    """Inserta las tres gráficas en el frame de la ventana tkinter."""
    # Línea
    fig1 = create_line_chart(horas, prec)
    canvas1 = FigureCanvasTkAgg(fig1, master=frm)
    canvas1.draw()
    canvas1.get_tk_widget().pack(pady=10, fill="x")

    # Barras
    fig2 = create_bar_chart(horas, prec)
    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, prec = fetch_data()
        if horas and prec:
            mostrar_graficas(frm, horas, prec)

    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()

```

Explicación de principales cambios en el código:

El primer cambio se realizó en la URL de la API, se cambiaron los valores de latitud y longitud para tomar los valores de Ciudad de México en lugar de León.

Ciudad original (León): `latitude=21.12&longitude=-101.68`

Nueva ciudad (Ciudad de México): `latitude=19.43&longitude=-99.13`

Además, se cambió la importación de temperaturas, para en cambio, utilizar y graficar los valores de precipitación de esta ciudad.

También se cambió el estilo de las gráficas, añadiendo `color = "green"` para cambiar el color de ambos gráficos, se cambió por línea punteada y diamantes como marcadores.

Finalmente, se cambió el nombre de las variables que guardaban anteriormente los valores de temperatura, por nombres apropiados para la lista de precipitación.