



Codigo:

```
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 Zurich, Suiza (últimas 24 horas).
    Devuelve dos listas: horas y temperaturas.
    """
    try:
        url = (
            "https://api.open-meteo.com/v1/forecast"
            "?latitude=47.37617306663388&longitude=8.544114338689061"
            "&hourly=temperature_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"]

        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 personalizada."""
    fig, ax = plt.subplots(figsize=(10, 4))
    ax.plot(
        horas,
        temps,
        linestyle="--",
        marker="o",
        markevery=4, # cada 4 puntos
        markersize=6,
        markerfacecolor="white",
        markeredgewidth=1.5,
        color="#1f77b4", # azul moderno
        linewidth=2
    )
    ax.set_title("Temperatura en Zurich (últimas 24 horas)",
fontsize=14, fontweight="bold")
    ax.set_xlabel("Hora", fontsize=12)
    ax.set_ylabel("Temperatura (°C)", fontsize=12)
    ax.grid(True, linestyle="--", alpha=0.5)
    ax.tick_params(axis="x", rotation=45)
    ax.set_facecolor("#f9f9f9") # fondo claro
    fig.tight_layout()
    return fig

```

```

def create_bar_chart(horas, temps):
    """Gráfica de barras personalizada."""
    fig, ax = plt.subplots(figsize=(10, 4))
    bars = ax.bar(horas, temps, color="#ff7f0e", edgecolor="black")

    # Agregar etiquetas sobre cada barra (cada 4 barras)
    for i, bar in enumerate(bars):

```

```

        if i % 4 == 0:
            height = bar.get_height()
            ax.text(
                bar.get_x() + bar.get_width() / 2,
                height + 0.5,
                f"{height:.1f}",
                ha="center",
                va="bottom",
                fontsize=8,
                rotation=90
            )

    ax.set_title("Temperatura en Zurich (últimas 24 horas)",
fontsize=14, fontweight="bold")
    ax.set_xlabel("Hora", fontsize=12)
    ax.set_ylabel("Temperatura (°C)", fontsize=12)
    ax.tick_params(axis="x", rotation=45)
    ax.set_facecolor("#f9f9f9")
    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()

```

Cambios Realizados:

Se mejoraron las gráficas cambiando colores, aumentando el tamaño de la figura y de las letras, usando marcadores más visibles (cada 4 puntos), agregando cuadrículas, fondo claro y etiquetas sobre algunas barras para mostrar los valores.