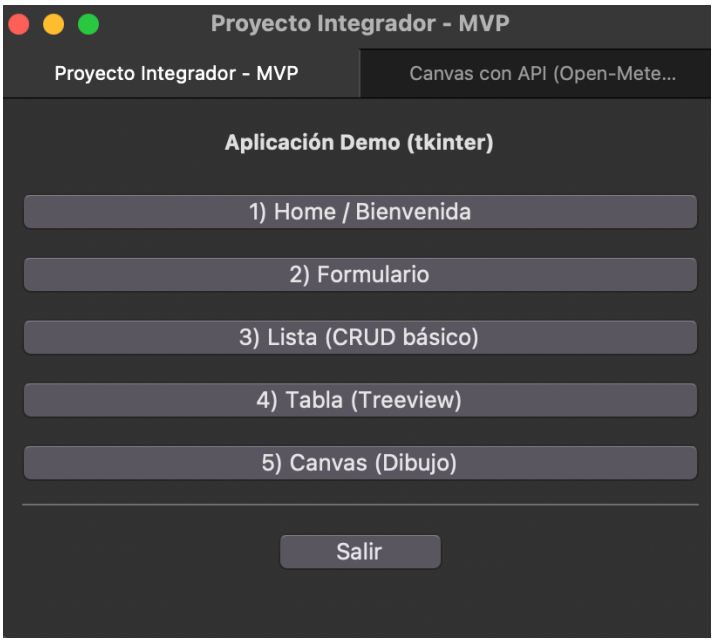
Tecnológico de Monterrey

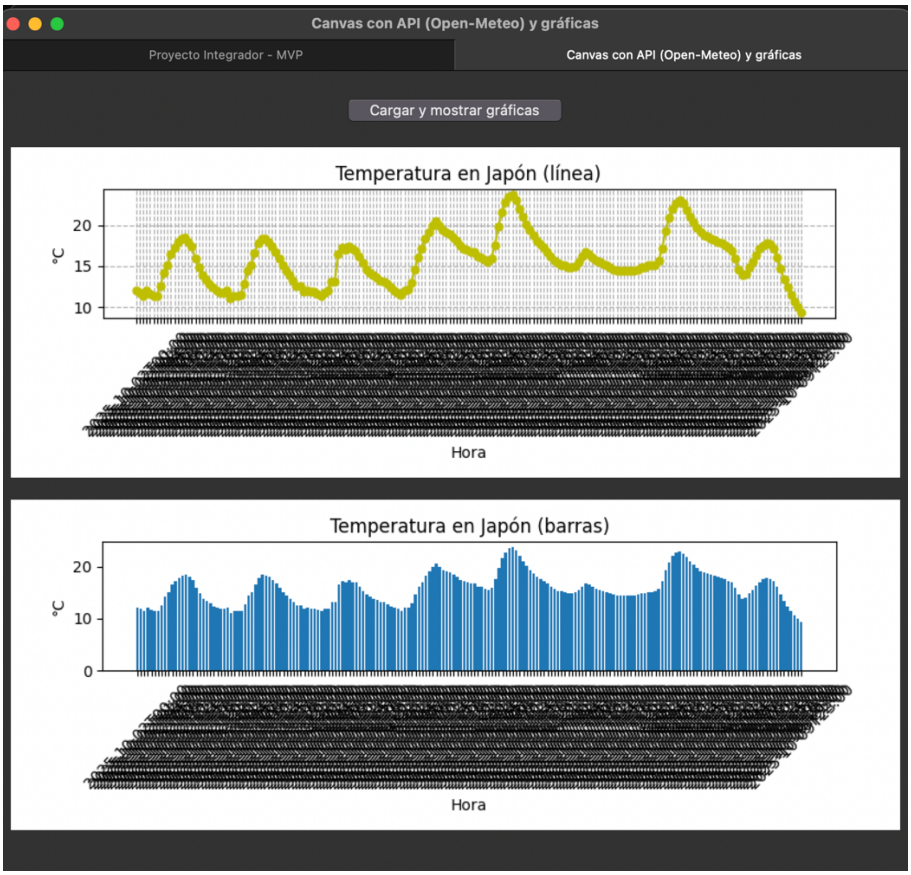
SESIÓN 7
Actividad en clase

Carlos David Martínez Rocha
2 de octubre de 2025

Tecnológico de Monterrey
Departamento de ingeniería y ciencias
Pensamiento Computacional para Ingeniería
Camilo Duque

SESIÓN 7 – CAMBIOS EN MI.





Código con cambios:

```
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:
        url = (
            "https://api.open-meteo.com/v1/forecast"
            "?latitude=36.20&longitude=138.25"
            "&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."""
    fig, ax = plt.subplots(figsize=(6, 3))
    ax.plot(horas, temps,'y', linestyle="-", marker="o", markersize=5)
    ax.grid(True, linestyle="--", alpha=1)
    ax.set_title("Temperatura en Japón (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, temps):
    """Gráfica de barras."""
    fig, ax = plt.subplots(figsize=(6, 3))
    ax.bar(horas, temps)
    ax.set_title("Temperatura en Japón (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, 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()
```

REPORTE:

Lista de cambios aplicados a la ventana

1. **Cambio de ubicación en la API (datos climáticos)**
    - **Antes:** La API consultaba la ciudad de León, Guanajuato (latitud 21.12, longitud -101.68).
    - **Ahora:** Se modificó la latitud y longitud para obtener datos de **Japón (latitud 36.20, longitud 138.25)**.
    - Este cambio permite visualizar el comportamiento de temperaturas en otra región del mundo, haciendo más versátil la aplicación.
2. **Ajuste en la gráfica de línea**
    - **Antes:** Línea azul simple, sin cuadrícula, con marcadores pequeños (tamaño 3).
    - **Ahora:** Línea de color amarillo ('y'), marcadores más grandes (tamaño 5), y se agregó **rejilla (grid)** con estilo punteado --.
    - Estos cambios hacen que la gráfica sea más clara, fácil de leer y visualmente atractiva para el usuario.
3. **Actualización de títulos en gráficas**
    - **Antes:** “Temperatura en León (línea/barras)”.
    - **Ahora:** “Temperatura en Japón (línea/barras)”.
    - Esto refuerza la coherencia con el cambio de ubicación en la API.
4. **Mantenimiento de estructura y funciones**

- Se conservaron los métodos principales: fetch_data, create_line_chart, create_bar_chart, mostrar_graficas y open_win_canvas.
- Se mantuvo la interfaz con Tkinter, asegurando que los cambios solo impactaran en la **visualización de datos** y no en la estructura base del programa.

Breve reflexión

Los cambios realizados apoyan la idea del proyecto porque enriquecen la experiencia de aprendizaje al mostrar **información real y actualizada de diferentes lugares del mundo**. Cambiar de León a Japón demuestra que la app no está limitada a un solo contexto, sino que puede adaptarse a múltiples regiones, lo cual fomenta la curiosidad de los programadores.

Además, las mejoras visuales (línea amarilla, marcadores más grandes y cuadrícula) hacen que las gráficas sean **más claras y comprensibles**, algo fundamental cuando se trabaja con niños o usuarios que están desarrollando su capacidad de interpretar datos.

En conjunto, estas modificaciones logran que la ventana no solo cumpla con mostrar gráficas, sino que también sea **atractiva, educativa y adaptable**, alineándose con el objetivo de un aprendizaje interactivo y accesible.