**Programa meteorológico - Desarrollado en Python**

**Desarrollado por: Alvarez, Benjamin**

**Lenguaje: Python  
Idioma: Español  
Fecha: 14/05/2023**

**Introducción:**

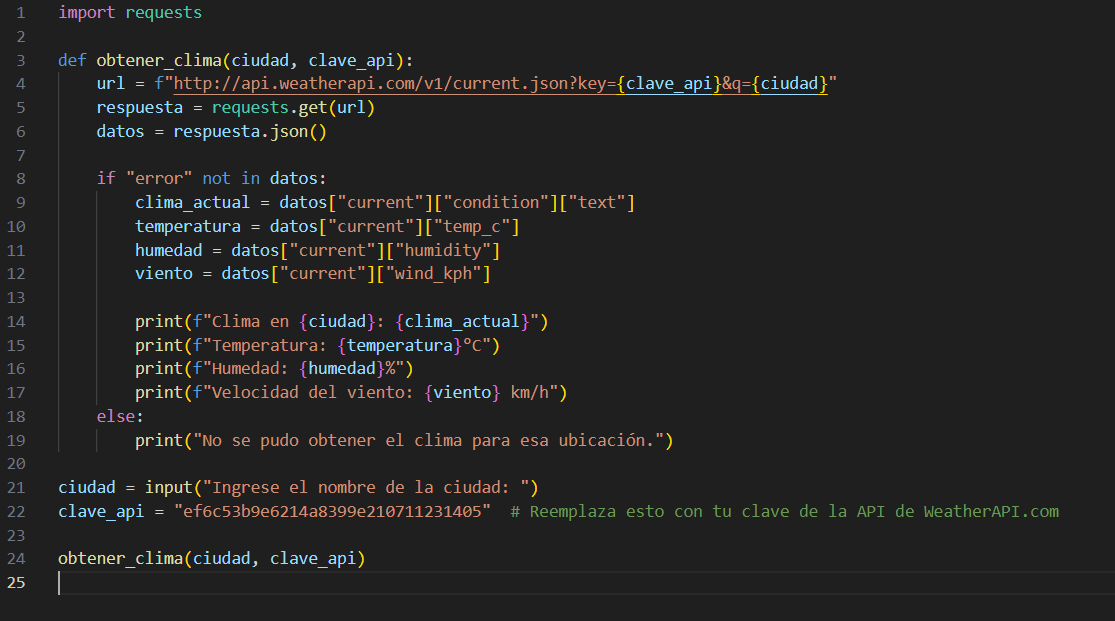
Un programa meteorológico es una herramienta poderosa que permite recopilar, analizar y visualizar datos relacionados con el clima y las condiciones atmosféricas. Con el avance de la tecnología y el acceso a una gran cantidad de información meteorológica en tiempo real, estos programas se han vuelto cada vez más sofisticados y precisos.

El objetivo principal de un programa meteorológico es proporcionar pronósticos precisos y actualizados, así como información detallada sobre el clima actual. Esto es esencial en diversas áreas, desde la agricultura y la aviación hasta la planificación urbana y la gestión de desastres. Al utilizar modelos matemáticos complejos y algoritmos sofisticados, estos programas pueden estimar las condiciones futuras del clima basándose en datos históricos y observaciones actuales.

El desarrollo de un programa meteorológico implica varios pasos. En primer lugar, se recopilan datos meteorológicos de diversas fuentes, como estaciones meteorológicas, satélites y radares. Estos datos incluyen variables como temperatura, presión atmosférica, humedad, velocidad y dirección del viento, entre otros. Luego, estos datos se procesan y se utilizan para alimentar modelos numéricos que simulan y predicen el comportamiento atmosférico.

La programación juega un papel fundamental en el desarrollo de un programa meteorológico. Los lenguajes de programación, como Python, se utilizan para crear algoritmos que procesan los datos meteorológicos, generan pronósticos y visualizan la información de manera comprensible. Python, en particular, es ampliamente utilizado en el ámbito de la meteorología debido a su facilidad de uso, su amplia disponibilidad de bibliotecas científicas y su capacidad para manipular datos de manera eficiente.

En este contexto, en el siguiente código de Python, se muestra un ejemplo básico de cómo se podría desarrollar un programa meteorológico utilizando Python. Este código es solo una muestra simple y se pueden realizar muchas mejoras y agregados para adaptarlo a necesidades específicas

**CÓDIGO:  
**

**EXPLICACIÓN:**

1. **Importación de la biblioteca requests:**

import requests

Esta línea importa el módulo requests, que nos permite enviar solicitudes HTTP

una API para obtener datos.

1. **Definición de la función obtener clima:**

def obtener clima(ciudad, clave api):

# Código de la función

Aquí se define la función obtener clima que toma dos argumentos: ciudad y clave api. Esta función se encargará de obtener y mostrar el clima actual de una determinada ciudad utilizando la API de WeatherAPI.com.

1. **Construcción de la URL de la API:**

url = f"[http://api.weatherapi.com/v1/current.json?key={clave\_api}&q={ciudad](http://api.weatherapi.com/v1/current.json?key=%7Bclave_api%7D&q=%7Bciudad)}"

Esta línea construye la URL de la API concatenando la clave de API y el nombre de la ciudad proporcionados como argumentos a la función. Esta URL se utilizará para enviar una solicitud GET a la API y obtener los datos del clima.

1. **Envío de la solicitud GET a la API:**

respuesta = requests.get(url)

Aquí se envía una solicitud GET a la URL de la API utilizando requests.get(). La respuesta de la API se almacena en la variable respuesta.

1. **Extracción de los datos del clima de la respuesta:**

datos = respuesta.json()

Esta línea convierte la respuesta de la API en formato JSON en un diccionario de Python utilizando el método .json() proporcionado por requisitos. Los datos del clima se almacenan en la variable datos.

1. **Verificación de errores en la respuesta:**

if "error" not in datos:

# Código para mostrar los datos del clima

else: print("No se pudo obtener el clima para esa ubicación.")

Aquí se verifica si hay algún error en la respuesta de la API. Si no se encuentra la clave "error" en el diccionario datos, significa que no hubo errores y los datos del clima están disponibles para mostrar. De lo contrario, se imprime un mensaje indicando que no se pudo obtener el clima para esa ubicación.

1. **Extracción y visualización de los datos del clima:**

clima actual = datos["current"]["condition"]["text"]

temperatura = datos["current"]["temp c"]

humedad = datos["current"]["humidity"]

viento = datos["current"]["wind km h"]

printf"Clima en {ciudad}: {clima actual}")

printf"Temperatura: {temperatura}°C")

print(f"Humedad: {humedad}%")

printf"Velocidad del viento: {viento} km/h")

Aquí se extraen los datos del clima del diccionario datos y se almacenan en variables individuales. Luego, se imprimen en la consola para mostrar el clima actual, la temperatura, la humedad y la velocidad del viento.

1. **Solicitud de entrada al usuario y llamada a la función obtener clima:**

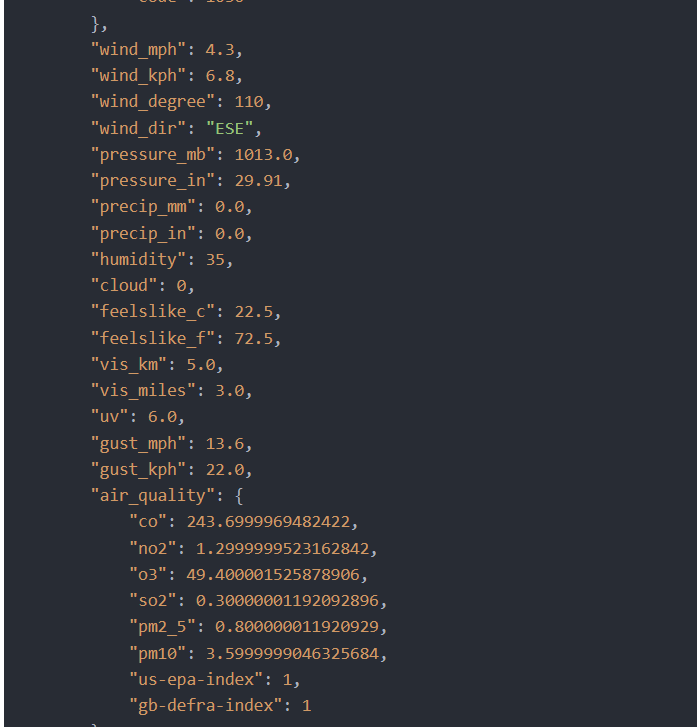
ciudad = input("Ingrese el nombre de la ciudad: ")

clave\_api = "ef6c53b9e6214a8399e210711231405" # Reemplaza

**NOTAS**





****