

DEMOSTRACIÓN DE LA APP:

Esta es la GUI de nuestro widget meteorológico. Podemos ver esto en la esquina superior izquierda de la pantalla.

Nombre de la ciudad

Buscar clima

Tenemos que ingresar cualquier nombre de la ciudad en el cuadro de entrada. De esta manera:

Nombre de la ciudad

Toronto

Buscar clima

Y hacemos clic en el botón "Buscar clima".

Toronto

Clima: Clear

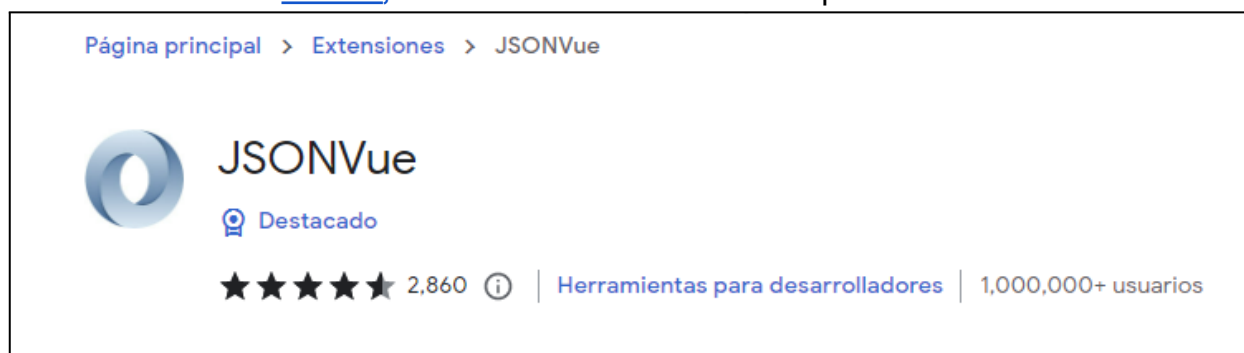
Humedad: 62%

Podemos ver que obtenemos el informe de clima y humedad de Toronto, que es una ciudad en Canadá. Y si observamos bien, podemos ver que el widget de entrada y el botón se vuelven invisibles al mostrar solo el informe del clima y la humedad. De la misma manera, también podemos buscar diferentes lugares en todo el mundo.

Comprender y obtener los datos del formato JSON:

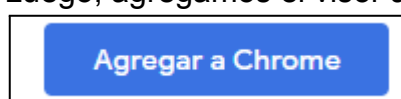
Paso 1.

- Primero abrimos el [enlace](#), tiene un enlace de referencia para el visor JSON.



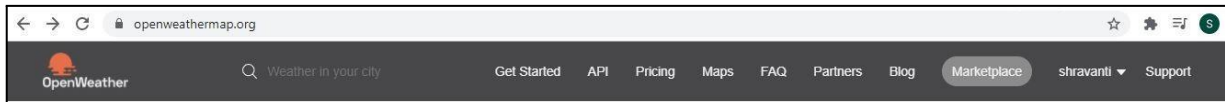
Paso 2:

- Luego, agregamos el visor JSON a Chrome simplemente haciendo clic en el botón **Agregar a Chrome**.



Paso 3.

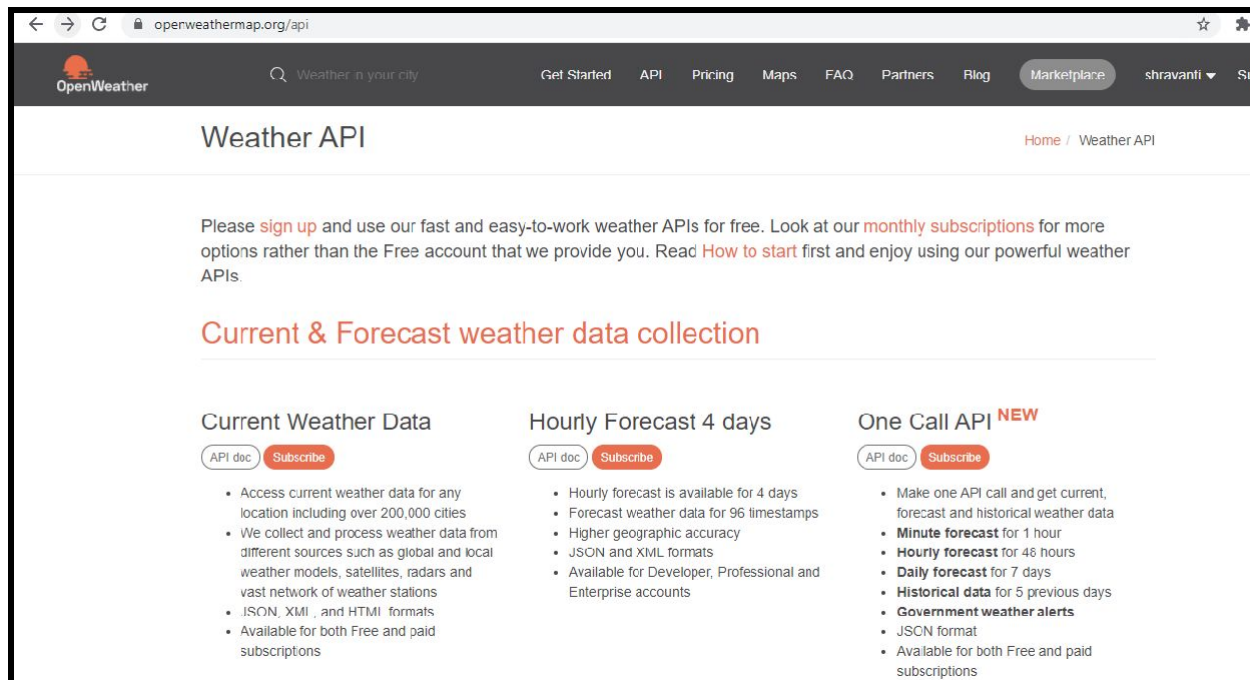
- Vamos al sitio web <https://openweathermap.org/> y hacemos clic en la pestaña API.



Paso 4.

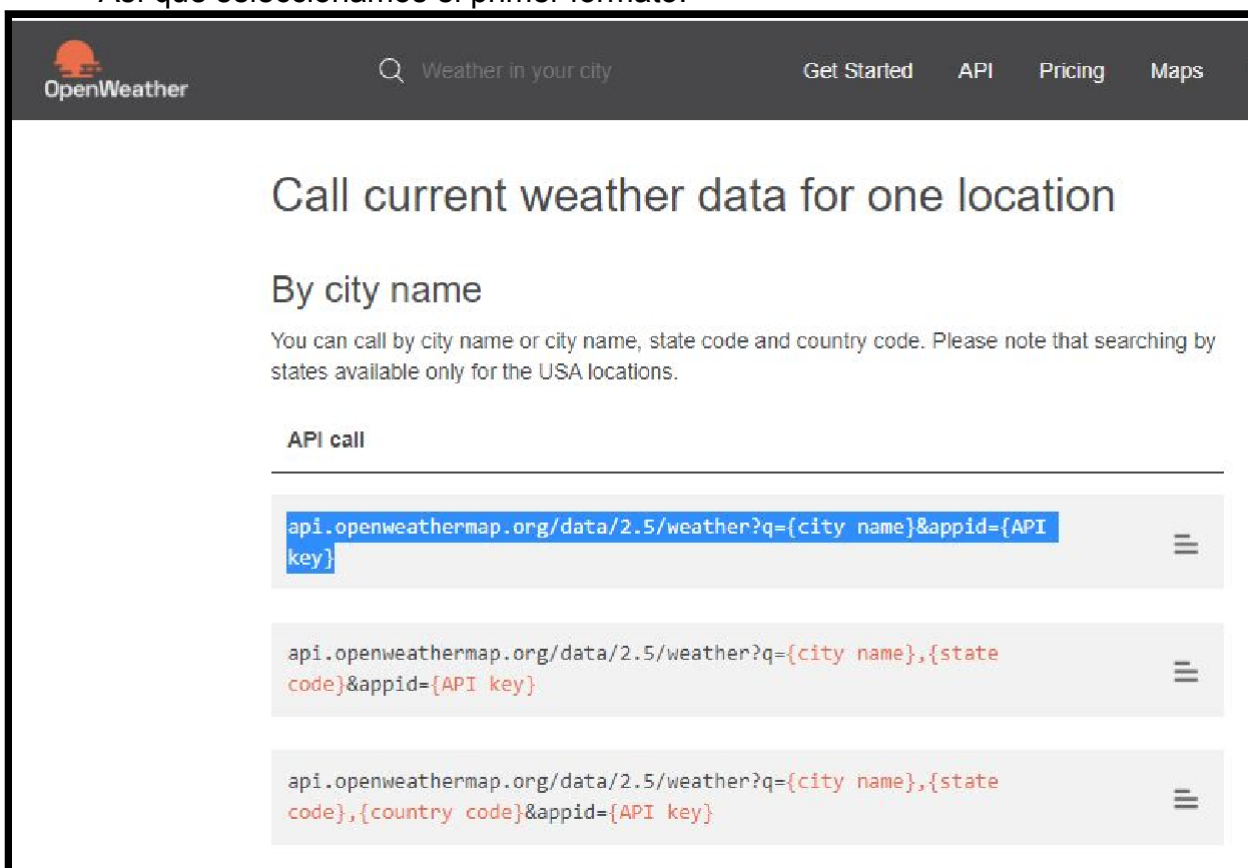


- Luego presionamos el botón API doc debajo de los datos meteorológicos actuales.



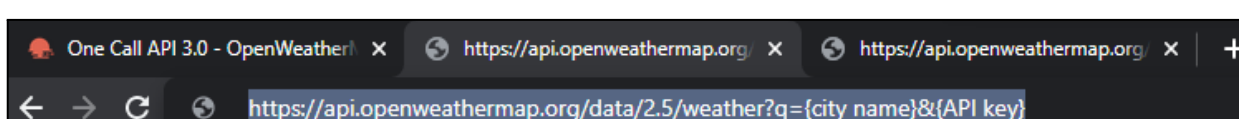
Paso 5.

- En esta página, nos desplazamos hacia abajo hasta las API call, donde obtendremos ciertos formatos en los que podemos llamar a la API usando nombres de ciudades. Así que seleccionamos el primer formato.



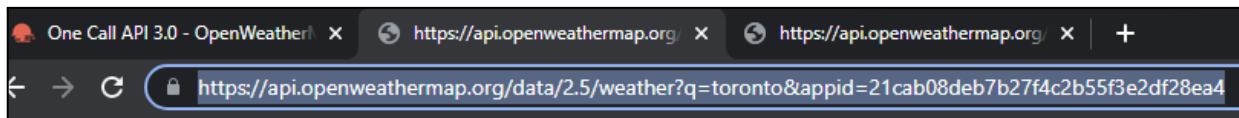
Paso 6.

- Ahora abrimos una nueva pestaña y pasamos el enlace que copiamos.



Paso 7.

- Ahora reemplazamos donde está escrito **{city_name}** con cualquier nombre de ciudad. Por ejemplo, Toronto y también reemplazamos {api key} con la clave API real sin incluir las llaves. Y presionamos enter.



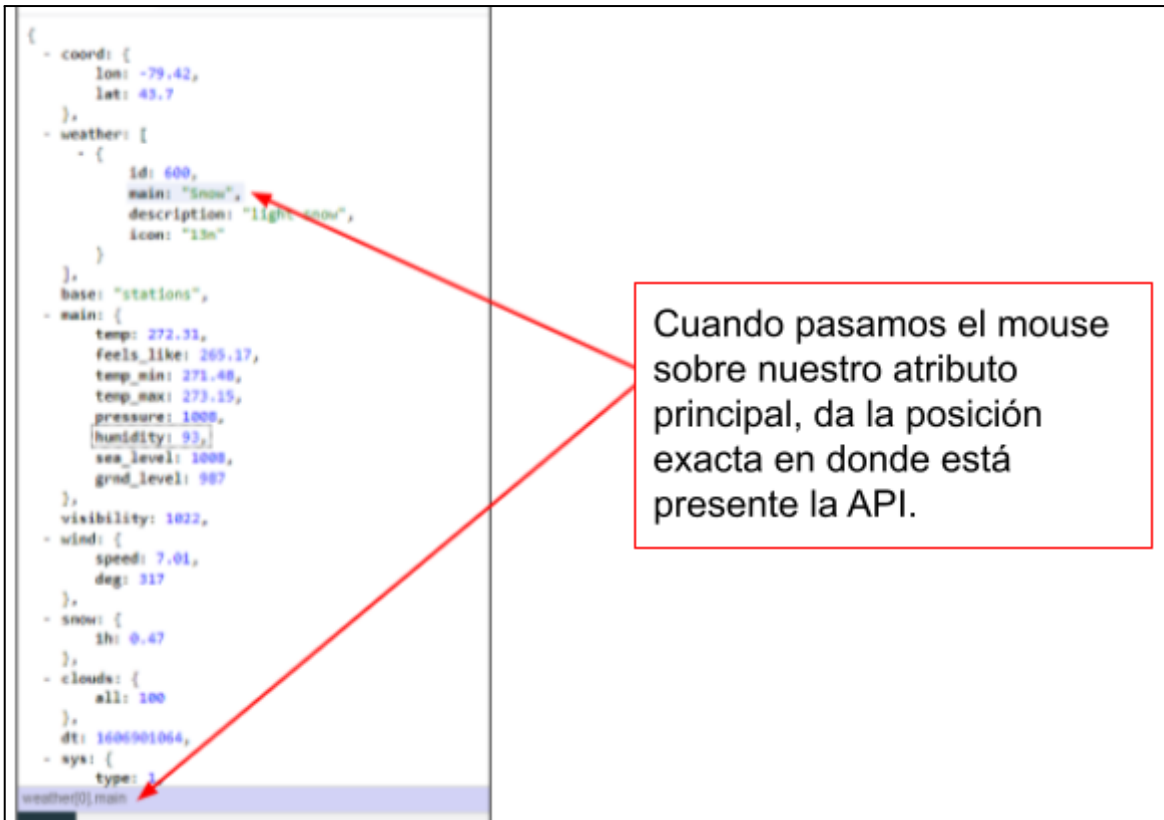
Paso 8.

- Encontraremos la página que muestra los datos en algún diccionario y formato de lista, por lo que estos son básicamente los datos del clima en formato JSON en la ciudad de Toronto.

```
{
  - coord: {
    lon: -79.42,
    lat: 43.7
  },
  - weather: [
    - {
      id: 600,
      main: "Snow",
      description: "light snow",
      icon: "13n"
    }
  ],
  base: "stations",
  - main: {
    temp: 272.31,
    feels_like: 265.17,
    temp_min: 271.48,
    temp_max: 273.15,
    pressure: 1008,
    humidity: 93,
    sea_level: 1008,
    grnd_level: 987
  },
  visibility: 1022,
  - wind: {
    speed: 7.01,
    deg: 317
  },
  - snow: {
    1h: 0.47
  },
  - clouds: {
    all: 100
  },
  dt: 1606901064,
  - sys: {
    type: 1,
    id: 718,
```

Paso 8.

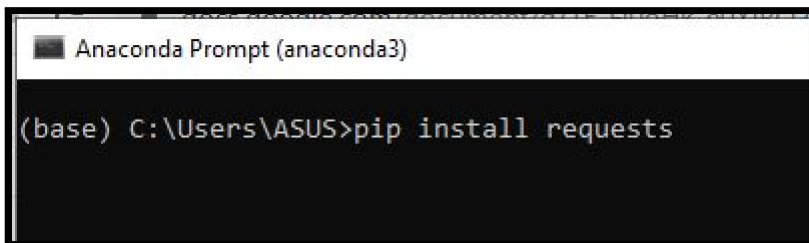
- Pasamos el mouse sobre los datos que deseamos, supongamos que pasamos el mouse sobre la API principal dentro del objeto meteorológico. Luego, abajo, en la esquina del lado izquierdo, encontraremos que se muestran algunos datos que no son más que la posición de la API principal dentro del objeto meteorológico. Entonces, al usar esto, podemos obtener las posiciones de las API y usarlas en nuestro programa.



Instalación de **requests** en Anaconda prompt.

Para usuarios de Windows:

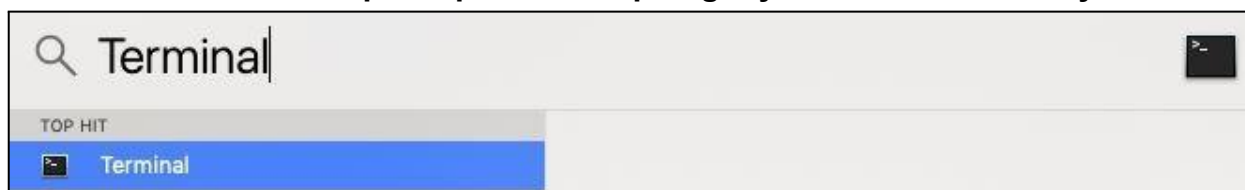
Ve a Inicio >> Anaconda prompt >> escribe **pip install requests** >> enter.



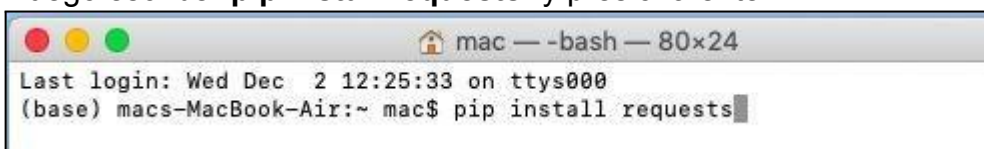
Recibirás el mensaje "**successfully installed**" -
(*instalado exitosamente*).

Para usuarios de Mac:

Presiona comando + espacio para abrir Spotlight y escribe "Terminal" y abre la Terminal.



Y luego escribe "**pip install requests**" y presiona enter



Código predefinido:

```
1 from tkinter import *
2 root=Tk()
3 root.title("Mi app meteorológica")
4 root.geometry("350x300")
5
6 root.configure(background="white")
7 #Etiquetas de configuración
8 city_name_label=Label(root, text="Nombre de la ciudad",font=("Helvetica", 12,'bold'),bg="white")
9 city_name_label.place(relx=0.28,relx=0.15,anchor=CENTER)
10
11 city_entry=Entry(root)
12 city_entry.place(relx=0.28,relx=0.35,anchor=CENTER)
13
14 weather_info_label = Label(root, bg="white", font=("bold", 10))
15 weather_info_label.place(relx=0.23,relx=0.6,anchor=CENTER)
16
17 humidity_info_label = Label(roo, bg="white", font=( "bold",10))
18 humidity_info_label.place(relx=0.22,relx=0.7,anchor=CENTER)
19
20 root.mainloop()
```

Diagrama de código:

1) Primero escribimos la estructura básica de tkinter.

2) Configuramos el fondo de la ventana root `root.configure(background="white")`.

3) Creamos una etiqueta en la ventana root.

```
city_name_label=Label(root, text="Nombre de la ciudad",font=("Helvetica", 12,'bold'),bg="white")
```

4) Luego, colocamos el elemento de etiqueta en la ventana root.

```
city_name_label.place(relx=0.5,relx=0.15,anchor=CENTER)
```

5) Creamos el elemento de entrada en la ventana root usando la clase Entry().

```
city_entry=Entry(root)
```

6) Luego colocamos el elemento de entrada en la ventana root `city_entry.place(relx=0.5,relx=0.35,anchor=CENTER)`.

7) Creamos una etiqueta weather_info en la ventana root.

```
weather_info_label = Label(root,text="Clima: ", bg="white", font=("bold", 10))
```

8) Luego colocamos la etiqueta weather_info en la ventana root.

```
weather_info_label.place(relx=0.5,relx=0.6,anchor=CENTER)
```

9) Creamos una etiqueta humidity_info_label en la ventana root.

```
humidity_info_label = Label(root,text="Humedad: ", bg="white", font=( "bold",10))
```

10) Luego colocamos el elemento de etiqueta en la ventana root.

```
humidity_info_label.place(relx=0.5,relx=0.7,anchor=CENTER)
```

CÓDIGO COMPLETO:


```

1  from tkinter import *
2  import requests
3  import json
4
5  root=Tk()
6  root.overrideredirect(True)
7  root.title("Mi app meteorológica")
8
9  root.geometry("200x200")
10 root.configure(background="white")
11 #Etiquetas de configuración
12 city_name_label=Label(root, text="Nombre de la ciudad",font=("Helvetica", 12,'bold'),bg="white")
13 city_name_label.place(relx=0.5,relly=0.15,anchor=CENTER)
14
15 city_entry=Entry(root)
16 city_entry.place(relx=0.5,relly=0.35,anchor=CENTER)
17
18 weather_info_label = Label(root, bg="white", font=("bold", 10))
19 weather_info_label.place(relx=0.5,relly=0.38,anchor=CENTER)
20
21 humidity_info_label = Label(root, bg="white", font=( "bold",10))
22 humidity_info_label.place(relx=0.5,relly=0.5,anchor=CENTER)
23
24 #Función principal
25 def city_name():
26     api_request = requests.get("https://api.openweathermap.org/data/2.5/weather?q=" +
27     city_entry.get() + "&appid="+ "21cab08deb7b27f4c2b55f3e2df28ea4")
28
29     api_output_json = json.loads(api_request.content)
30
31     weather_info=api_output_json['weather'][0]["main"]
32     print(weather_info)
33
34     humidity=api_output_json['main']['humidity']
35     print(str(humidity) + "%")
36
37     weather_info_label["text"]= "Clima: " + str(weather_info)
38     humidity_info_label["text"]="Humedad: " + str(humidity) + "%"
39
40     city_name_label["text"]= city_entry.get()
41     city_entry.destroy()
42     search_btn.destroy()
43
44 search_btn=Button(root, text="Buscar clima", command=city_name, relief=FLAT)
45 search_btn.place(relx=0.5,relly=0.48,anchor=CENTER)
46 root.mainloop()
47

```

Desglose del código:

1) Primero importamos los dos paquetes:

- `import requests` : para enviar la solicitud al sitio web <https://openweathermap.org/> para acceder a las APIs.
- `import json` : para obtener los datos con formato JSON del sitio web.

2) Creamos la función `def city_name():`.

- 3) Obtuvimos el nombre de la ciudad ingresada por el usuario desde el elemento de entrada y la concatenamos con la URL de la API junto con la clave de la API.
- 4) La clave de la API se te proporciona en la [Actividad del alumno 4](#).

```
api_request = requests.get("https://api.openweathermap.org/data/2.5/weather?q=" + city_entry.get() +
"&appid="+ "21cab08deb7b27f4c2b55f3e2df28ea4")
```

- 5) Usando la función `json.loads()` cargamos los contenidos obtenidos del sitio web que se almacenan dentro de la variable `api_request`.

```
api_output_json = json.loads(api_request.content)
```

- 6) Usando la variable `api_output_json`, obtuvimos el atributo principal que está presente en el objeto `weather`.

```
weather_info=api_output_json['weather'][0]["main"]
```

- 7) También imprimiremos el valor del clima en la consola usando la función `print()`, pasando la variable `weather_info` dentro de la función `print()`, de esta manera: `print(weather_info)`

```
api_output_json['main']['humidity']
```

- 8) Usando la variable `api_output_json`, accedimos al objeto `main`, desde el cual recuperamos la API de humedad.
- 9) Y almacenamos este valor de humedad obtenido en la variable `humidity`.

```
humidity=api_output_json['main']['humidity']
```

- 10) También imprimimos la humedad en la consola usando la función `print ()` y concatenamos la humedad obtenida con el signo de porcentaje `"%"`, así:

```
print(str(humidity) + "%")
```

- 11) A continuación, actualizamos el parámetro `"text"` de la etiqueta `weather_info_label` con la cadena de caracteres `"Clima: "`, lo concatenamos con `str(weather_info)` y también imprimimos la variable `weather_info` en la consola usando la función `print()`.

```
weather_info_label["text"] = "Clima: " + str(weather_info)
print(weather_info)
```

- 12) A continuación, actualizamos el parámetro de `"text"` de la etiqueta `label_info_humidity` con la cadena de caracteres `"Humedad:"`, lo concatenamos con `str(humidity)` y lo concatenamos con `"%"`; también imprimimos `weather_info` en la consola usando la función `print()`.

```
humidity_info_label["text"] = "Humedad: " + str(humidity) + "%"
print(str(humidity) + "%")
```

- 13) Actualizamos el parámetro de texto de la etiqueta `city_name_label` con el nombre de la ciudad ingresado en el campo de entrada.

```
city_name_label["text"] = city_entry.get()
```

- 14). Luego, usando la función `destroy()`, eliminamos el campo de entrada de esta manera

```
city_entry.destroy()
```

.

- 15) Luego, nuevamente usando la función `destroy()`, eliminamos el elemento del botón de esta manera

```
search_btn.destroy()
```

- 16) Luego creamos el elemento de botón para llamar a la función `city_name()` para ejecutar todas las funcionalidades de la app.

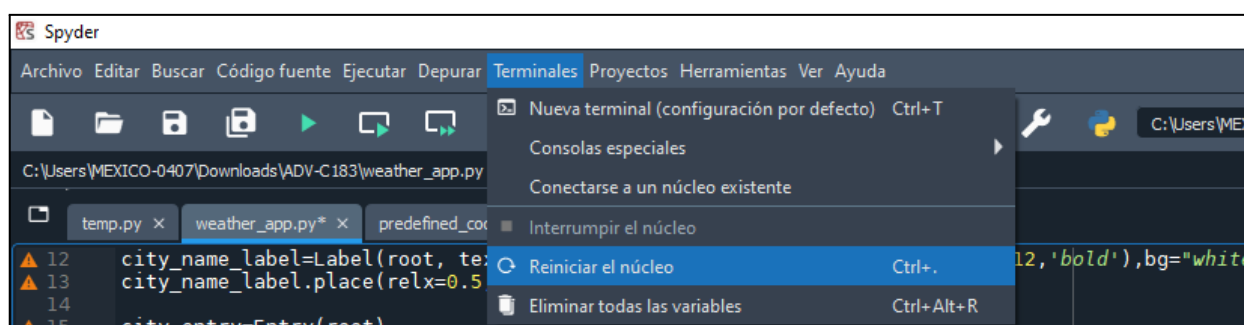
```
search_btn=Button(root, text="Buscar clima", command=city_name, relief=FLAT)
```

- 17) Colocamos el elemento de botón en la ventana `root`, configurando las posiciones de `relx` y `rely` del elemento del botón.

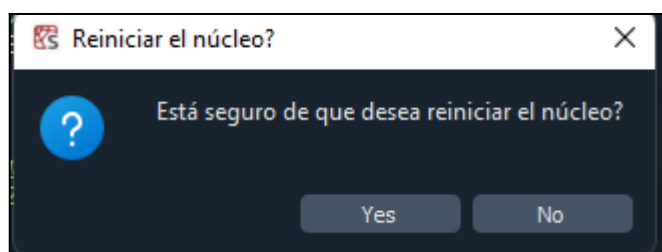
```
search_btn.place(relx=0.5, rely=0.48, anchor=CENTER)
```

Cerrando el widget.

Luego, simplemente vamos a **Terminales >> Reiniciar el núcleo.**



Hacemos clic en **yes**:



Se cerrará el widget.