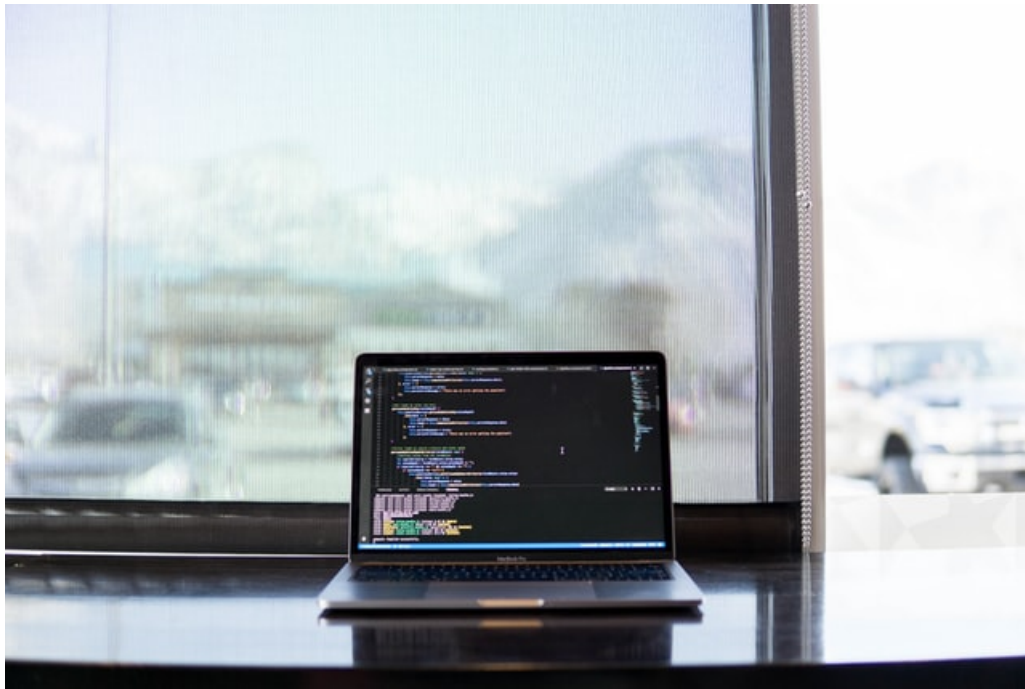


Obtén información meteorológica con Python

Publicado el [El Informatico](#) - 25 de agosto de 2021 -



Fotografía por [Maxwell Nelson](#) ([unsplash.com](#))

Gracias al lenguaje de script Python podemos crear pequeñas aplicaciones que nos permiten hacer cosas como, por ejemplo, automatización de tareas, interfaces web, o incluso recopilación y analítica de datos, entre muchas otras aplicaciones (incluido Inteligencia Artificial con tensorflow).

En éste artículo explicaré como crear un sencillo script para recopilar datos meteorológicos de la página web de la AEMET (Agencia Estatal de Meteorología española) que después podrás tratar y procesar como desees.

¿Qué es la AEMET?

La AEMET es la Agencia Estatal de Meteorología, que se encarga de realizar estudios meteorológicos en España. Son ellos los que se encargan de analizar la situación meteorológica para después ofrecer una previsión meteorológica para los próximos días, así como realizar alertas sobre posibles efectos adversos como olas de calor, fuerte oleaje en las costas, lluvias torrenciales, etcétera.

O lo que es lo mismo, si quieres una previsión del tiempo para los próximos días sobre cualquier lugar de España, debes acudir a ellos.

La página web de la AEMET (www.aemet.es) nos ofrece numerosos servicios. Podemos acceder a todos esos servicios a través de nuestro navegador, pero también ofrecen la información, pero también se ofrecen los datos en formato **xml** o **CSV**.

En concreto, para obtener la predicción del tiempo en nuestra comunidad, podemos acceder desde el portal web a la sección El Tiempo > Predicción > Municipios. A través de dicha sección, podemos seleccionar la comunidad en el mapa, y después nuestra ciudad.



Predicción para la ciudad de Madrid en los próximos días

Para poder usar estos datos en nuestro script, debemos descargar los datos en formato XML. Podemos obtener la dirección sobre el correspondiente botón XML.

[Descargar XML de la predicción detallada de Madrid](#) XML

Cómo tratar la información con Python

La información en XML está dividida en “nodos”. Cada nodo puede contener una serie de subnodos, así como contener parámetros. En el caso de la predicción por municipios que nos ofrece la AEMET, la información que queremos se encuentra en

el nodo “predicción”, con distintos subnodos llamados “dia” para cada uno de los 7 días. Esos elementos “dia” tienen un parámetro “fecha” especificando la fecha de la predicción.

```
https://www.aemet.es/es/eltiempo/prediccion/municipios/madrid-id28079
</enlace>
<language>es</language>
<copyright>
  © AEMET. Autorizado el uso de la información y su reproducción citando a AEMET como autora de la misma.
</copyright>
<nota_legal>https://www.aemet.es/es/nota_legal</nota_legal>
</origen>
<elaborado>2021-08-25T12:19:17</elaborado>
<nombre>Madrid</nombre>
<provincia>Madrid</provincia>
<prediccion>
  <dia fecha="2021-08-25">
    <prob_precipitacion periodo="00-24"/>
    <prob_precipitacion periodo="00-12"/>
    <prob_precipitacion periodo="12-24">50</prob_precipitacion>
    <prob_precipitacion periodo="00-06"/>
    <prob_precipitacion periodo="06-12">0</prob_precipitacion>
    <prob_precipitacion periodo="12-18">50</prob_precipitacion>
    <prob_precipitacion periodo="18-24">0</prob_precipitacion>
    <cota_nieve_prov periodo="00-24"/>
    <cota_nieve_prov periodo="00-12"/>
    <cota_nieve_prov periodo="12-24"/>
    <cota_nieve_prov periodo="00-06"/>
    <cota_nieve_prov periodo="06-12"/>
    <cota_nieve_prov periodo="12-18"/>
    <cota_nieve_prov periodo="18-24"/>
    <estado_cielo periodo="00-24" descripcion=""/>
    <estado_cielo periodo="00-12" descripcion=""/>
    <estado_cielo periodo="12-24" descripcion="Intervalos nubosos con lluvia escasa">43</estado_cielo>
    <estado_cielo periodo="00-06" descripcion=""/>
    <estado_cielo periodo="06-12" descripcion="Despejado">11</estado_cielo>
    <estado_cielo periodo="12-18" descripcion="Intervalos nubosos con lluvia escasa">43</estado_cielo>
    <estado_cielo periodo="18-24" descripcion="Poco nuboso">12</estado_cielo>
  </dia>
  <viento periodo="00-24">
    <direccion/>
    <velocidad/>
  </viento>
  <viento periodo="00-12">
    <direccion/>
    <velocidad/>
  </viento>
  <viento periodo="12-24">
    <direccion>SO</direccion>
    <velocidad>15</velocidad>
  </viento>
  <viento periodo="00-06">
    <direccion>NE</direccion>
    <velocidad>5</velocidad>
  </viento>
  <viento periodo="06-12">
    <direccion>S</direccion>
  </viento>
</prediccion>
```

El día actual es siempre el que más información contiene, pudiendo obtener datos sobre la predicción en distintas franjas horarias incluyendo la probabilidad de predicción, dirección y fuerza del viento, temperatura, etc. Los siguientes 3 días ofrecen una información bastante completa. Los últimos 3 días no contienen apenas información, sólo temperatura, probabilidad de precipitación y viento.

Para descargar estos datos desde la página de la AEMET, primero crea un nuevo script con el nombre que quieras (como por ejemplo, “meteo.py”) e importa el modulo **requests** y **elementtree** para poder descargar el archivo xml de la página de la aemet.

```
import requests
import xml.etree.ElementTree as ET

# La URL se obtiene de la página de la AEMET, que se corresponde con el
# código postal del municipio.
data =
requests.get("https://www.aemet.es/xml/municipios/localidad_28079.xml").c
ontent
tree = ET.ElementTree(ET.fromstring(data))
```

Con ésto habremos generado un árbol de contenidos en la variable “tree”. Ahora podemos navegar a través de los distintos nodos a través de dicha variable, que contiene un objeto ElementTree con los distintos nodos del documento XML.

Por ejemplo, si quisiéramos mostrar en la consola la temperatura, probabilidad de precipitación, y viento de los primeros 4 días (los más certeros), lo podríamos hacer de la siguiente manera:

```
# Obtener el nodo "predicción"
for forecasts in tree.findall('prediccion'):
    i = 0
    # Obtener e iterar a través de los distintos nodos "dia".
    for forecast in forecasts.findall('dia'):
        # Finaliza el programa si ya hemos mostrado 4 días (índice 0)
        if i > 3:
            break
        # Muestra la fecha de la predicción, contenida en el atributo "fecha" del
        # nodo día.
        print('\n' + forecast.attrib['fecha'] + ':')
        # Obtener las probabilidades de precipitación.
        for rainprob in forecast.findall('prob_precipitacion'):
            # En el primer día, la información se divide por periodos de tiempo. Pero
            # no en el resto.
            if 'periodo' in rainprob.attrib and rainprob.text:
                # Muestra la predicción
                print((rainprob.attrib['periodo'] + 'h: ' + rainprob.text + '% \t' +
                forecast.find('..estado_cielo[@periodo="' + rainprob.attrib['periodo'] +
                "']").attrib['descripcion'] + '\t\t' +
                forecast.find('..viento[@periodo="' + rainprob.attrib['periodo'] +
                "']").find('velocidad').text + 'Km/h\t' +
                forecast.find('temperatura').find('minima').text + '°C\t' +
                forecast.find('temperatura').find('maxima').text + '°C').ljust(30))
                i += 1
```

Para ejecutar el script, desde la consola usamos el comando:

```
$ python meteo.py
```

El cuál nos muestra la siguiente información:

2021-08-25:					
12-24h: 40%	Intervalos nubosos		15Km/h	19°C	31°C
06-12h: 0%	Despejado	10Km/h	19°C	31°C	
12-18h: 40%	Intervalos nubosos		15Km/h	19°C	31°C
18-24h: 0%	Poco nuboso	15Km/h	19°C	31°C	
2021-08-26:					
00-24h: 0%	Nubes altas	10Km/h	20°C	33°C	
00-12h: 0%	Poco nuboso	15Km/h	20°C	33°C	
12-24h: 0%	Nubes altas	10Km/h	20°C	33°C	
00-06h: 0%	Poco nuboso	0Km/h	20°C	33°C	
06-12h: 0%	Poco nuboso	10Km/h	20°C	33°C	
12-18h: 0%	Nubes altas	5Km/h	20°C	33°C	
18-24h: 0%	Poco nuboso	0Km/h	20°C	33°C	
2021-08-27:					
00-24h: 0%	Despejado	10Km/h	20°C	35°C	
00-12h: 0%	Despejado	10Km/h	20°C	35°C	
12-24h: 0%	Despejado	10Km/h	20°C	35°C	
2021-08-28:					
00-24h: 0%	Despejado	10Km/h	16°C	34°C	
00-12h: 0%	Despejado	10Km/h	16°C	34°C	
12-24h: 0%	Despejado	10Km/h	16°C	34°C	

Almacenar la información en una base de datos

Si quisiéramos almacenar ésta información para poder consultarla más tarde, por ejemplo, para hacer un histórico, podemos hacerlo de manera simple conectando Python a nuestra base de datos. Python incluye un motor para SQLite por defecto que podemos usar para crear una base de datos simple.

Por ejemplo, para almacenar la información correspondiente al día actual, podríamos usar el siguiente script:

```

import xml.etree.ElementTree as ET
import requests
import sqlite3

data =
requests.get("https://www.aemet.es/xml/municipios/localidad_28079.xml").c
ontent
tree = ET.ElementTree(ET.fromstring(data))

# Crear/abrir una base de datos llamada "meteo.db" en el directorio desde
el que
# se ejecuta el script
con = sqlite3.connect("meteo.db")
# Obtener el cursor con el que realizaremos las distintas operaciones
cur = con.cursor()
# Crear una nueva tabla de datos si es necesario
res = cur.execute("CREATE TABLE IF NOT EXISTS prediccion (id INTEGER
PRIMARY KEY AUTOINCREMENT,region TEXT NOT NULL, fecha TEXT NOT NULL,
periodo TEXT NOT NULL, nubosidad TEXT NOT NULL, viento TEXT,
precipitacion INTEGER, minima INTEGER, maxima INTEGER)")
# Realizar los cambios
con.commit()

# Obtener el nodo de predicción
forecasts = tree.find('prediccion')
# Obtener el primer nodo de día
forecast = forecasts.find('dia')
# Obtener la region
region = tree.find('nombre').text + " (" + tree.find('provincia').text +
")"
# Obtener todas las probabilidades de precipitación
for rainprob in forecast.findall('prob_precipitacion'):
    if 'periodo' in rainprob.attrib and rainprob.text:
        # Introducir los datos en la base de datos
        cur.executemany("INSERT INTO prediccion
(fecha,region,periodo,nubosidad,viento,precipitacion,minima,maxima)
VALUES (?, ?, ?, ?, ?, ?, ?, ?)",
        [(region,
        str(forecast.attrib['fecha']),
        str(rainprob.attrib['periodo']),
        str(forecast.find('./estado_cielo[@periodo="' +
rainprob.attrib['periodo'] + '"']).attrib['descripcion']),
        int(forecast.find('./viento[@periodo="' + rainprob.attrib['periodo']
+ '"']).find('velocidad').text),
        int(rainprob.text),
        int(forecast.find('temperatura').find('minima').text),
        int(forecast.find('temperatura').find('maxima').text)
        )])

# Realizar los cambios
con.commit()

```

Esto almacenará la predicción para el día de hoy en un archivo de base de datos SQLite3 llamado "meteo.db", con la correspondiente fecha y por periodos dentro del mismo día.

Si exploramos la base de datos con SQLite, podemos observar los datos obtenidos

en la base de datos.

Database Structure Browse Data Edit Pragma Execute SQL									
Table: predicion									
	id	region	fecha	periodo	nubosidad	viento	precipitacion	minima	maxima
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	2021-08-25	Madrid (Madrid)	12-24	Intervalos nublados con lluvia escasa	15	50	21	30
2	2	2021-08-25	Madrid (Madrid)	06-12	Despejado	10	0	21	30
3	3	2021-08-25	Madrid (Madrid)	12-18	Intervalos nublados con lluvia escasa	15	50	21	30
4	4	2021-08-25	Madrid (Madrid)	18-24	Poco nublado	5	0	21	30

Si ejecutáramos este script cada día por la mañana, obtendríamos en el tiempo un historial de la información meteorológica en el tiempo. Es decir, a partir de ahora, podemos realizar analíticas sobre los datos obtenidos en el tiempo por cada región usando ésta base de datos.



ANTERIOR

Encriptación LUKS con CRYPTSETUP

SIGUIENTE

Mantén tus contraseñas seguras

Buscar ...



Entradas Recientes

- [El Metaverso: Nada nuevo en el horizonte](#)
- [Estafas telefónicas: ¡No caigas en la trampa!](#)
- [Jueves de buenas noticias](#)
- [Facebook, una vez más, en problemas. Y es su propia culpa.](#)
- [Si usas Twitch, cambia tu contraseña de inmediato](#)
- [Diferencias entre Internet y La Web. ¿Qué es cada uno?](#)

Categorías

[Actualidad](#)

[Android](#)

[Básicos](#)

[Ciberseguridad](#)

[Criptografía](#)

[Emulación / Virtualización](#)[FOSS](#)[Hacking](#)[Informática](#)[Internet](#)[Juegos](#)[Opinion](#)[Otros](#)[Personal](#)[Privacidad](#)[Programación](#)[Tecnología](#)[Time Machine](#)[Tutoriales](#)

RSS

[Subscribirse al feed RSS](#)

[Inicio](#)[Catálogo](#)[Tutoriales](#)[Política de privacidad](#)[Política de Cookies](#)[Acerca de mi](#)[Acerca de ElInformati.co](#)