ATIVIDADE AVALIADA – EXERCÍCIOS DE PROGRAMAÇÃO PYTHON 02

1. Na era da conectividade global, é possível acessar diferentes tipos de informação em qualquer lugar do mundo, desde que se disponha de um acesso à Internet. Isso serve não apenas para a visualização de pessoas, mas também para a automação de tarefas baseadas em dados do mundo exterior como, por exemplo, meteorologia. Existem diversos provedores de serviços de dados, que não provêm páginas web, mas sim dados, muitas vezes textualmente, organizados em formatos como XML ou JSON. Alguns desses serviços são gratuitos para usos limitados e não comerciais, enquanto outros são pagos.

Um webservice gratuito e que não requer autorização, criação de conta e outros detalhes de segurança é o open-meteo.com. Ele disponibiliza uma API (interface de programação de aplicações) baseada no protocolo HTTP, o que torna muito simples o acesso: basta ter uma URL com os parâmetros corretos, e o provedor retorna um conteúdo JSON com os dados solicitados. Por exemplo, para a cidade de Porto Alegre, a URL abaixo retorna os dados meteorológicos atuais:

https://api.open-meteo.com/v1/forecast?latitude=-30.033&longitude=-51.23&timezone=auto&current_weather=true

Em Python, existem vários pacotes e módulos que fornecem recursos para o acesso a webservices e posterior processamento dos dados obtidos nas requisições. Um deles é o **json**, que é parte da base do interpretador Python, que permite converter qualquer objeto em uma string no formato JSON e vice-versa. O formato do conteúdo retornado pela URL mostrada acima, por exemplo, é facilmente transformado em um dicionário Python, o que possibilita acessar qualquer valor retornado no próprio tipo de dado (string, numérico ou lógico) de forma simples.

O outro pacote de fácil uso e que facilita a requisição de dados para um servidor HTTP e a obtenção de respostas é o **requests**. Ele não é da base do interpretador Python, e pode ser necessário instalá-lo, de acordo com a plataforma que estiver utilizando (WinPython, replit.com, Anaconda ou Thonny – verifique como fazê-lo, se necessário). O requests disponibiliza diversas funções e tipos de dados para trabalhar com requisições e respostas HTTP, sendo `get()` uma das funções mais simples: ela requer que se passe uma URL (como string), estabelece a conexão com o servidor dessa URL, faz a requisição nela definida, aguarda uma resposta, que é retornada como um objeto do tipo `Response`. Entre os diferentes dados contidos nesse objeto, estão o `status_code`, que indica se a requisição foi bem-sucedida (um número), e o conteúdo, que pode ser textual ou binário. No caso da api do open-meteo.com, o conteúdo é textual e retornado no formato JSON. Aí, o método `json()` de um objeto do tipo `Response` é capaz de gerar um dicionário contendo os dados, e a partir de então é só processar os dados recebidos (ver a documentação em https://requests.readthedocs.io/en/latest/).

Um exemplo de uso de requests é apresentado abaixo, onde se obtém a cotação do Dólar em relação a Real:

```
from requests import get

grana='https://economia.awesomeapi.com.br/json/last/USD-BRL'

r=get(grana)

print (r.status_code) # Se status_code for 200, foi tudo bem

resultado = r.json()['USDBRL'] # Pega o valor da chave USDBRL (que também é um dicionário)

r.close() # Libera o resource (e bastante memória com isso)

for k in resultado:

    print (f'{k}: {resultado[k]}')
```

Com base na explicação acima, e nos exemplos disponíveis na documentação dos sites relacionados (e na própria documentação do Python), desenvolva um script que obtenha os dados meteorológicos atuais para a cidades que lhe for designada (em documento a parte). Com os dados obtidos, apresente a temperatura atual, a velocidade/direção do vento e qual é a situação do tempo (chuvoso, nublado, ensolarado, de acordo com o WMO Weather interpretation codes – ver na documentação do open-meteo.com).

Crie também um script Python que faça um estudo histórico sobre a evolução do tempo na cidade que lhe for designada. Obtenha a temperatura, umidade relativa do ar, a temperatura do ponto de orvalho, velocidade e direção do vento durante um determinado período de tempo. Isso é possível com o seguinte padrão de URL:

https://api.open-meteo.com/v1/forecast?latitude=**lat**&longitude=**lon**&hourly=temperature_2m,relativehumidity_2m,dewpoint_2m,windspeed_10m,winddirection_10m&timezone=auto&start_date=**inicial**&end_date=**final**

lat e lon são as coordenadas da cidade em questão, inicial e final são as datas inicial e final do período solicitado, em formato ISO8601 (aaaa-mm-dd). Esses parâmetros devem ser substituídos pelos valores da cidade e do período a ser avaliado.

O script deve salvar os dados obtidos para o período de um mês (especificado em documento a parte) em um arquivo no formato csv, que possa ser importado por uma planilha eletrônica como a Google Planilhas, Microsoft Excel ou LibreOffice Calc. O formato das linhas deve ser o seguinte:

<Data/Hora>;<temperatura>;<ponto de orvalho>;<umidade relativa>;<direção do vento>;<velocidade do vento>

Observe que o separador de campos deve ser o caractere ‘;’. Os dados numéricos utilizam . como separador de casas decimais (na importação, deve-se definir que o idioma da planilha é inglês. Isso fará com que a planilha reconheça corretamente o ponto decimal).

Por exemplo, uma linha de dados seria assim:

2022-09-01T00:00;14.1;13.5;13.0;93

Utilize uma planilha eletrônica para criar os seguintes gráficos de dispersão a partir do arquivo csv gerado:

Data/Hora x temperatura e ponto de orvalho

Data/Hora x umidade relativa do ar

Salve a planilha com os gráficos no formato do software utilizado (não modifique o csv original).

Crie também um script Python para criar gráficos de rosa dos ventos (WindRose) do mês. Utilize matplotlib para tanto. O gráfico pode ser criado como um gráfico polar, a partir de uma redução dos dados. Calcule a média de cada hora do mês (por exemplo, média da direção e da velocidade do vento para as 14:00). A partir desses dados, gere um novo arquivo com esses resultados de médias horárias. Com esse arquivo, gere as rosas dos ventos por turnos: manhã (06:00-11:00), tarde (12:00-17:00), noite (18:00-23:00) e madrugada (00:00-05:00).

ATENÇÃO: Reúna todos os scripts e arquivos gerados por estes em um único arquivo zip, cujo nome deve conter os quatro últimos algarismos de seu número de matrícula como sufixo (por exemplo: t01-xxxx.zip). Este arquivo deve ser submetido no AVA E enviado por e-mail para carlos.rocha@riogrande.ifrs.edu.br.

Prazo limite para entrega: 05/05/2023