**CÓDIGO DE PROCESSAMENTO DOS DADOS DE ENTRADA DO MODELO HIDROLÓGICO CAAWM**

1) **Considerações gerais**

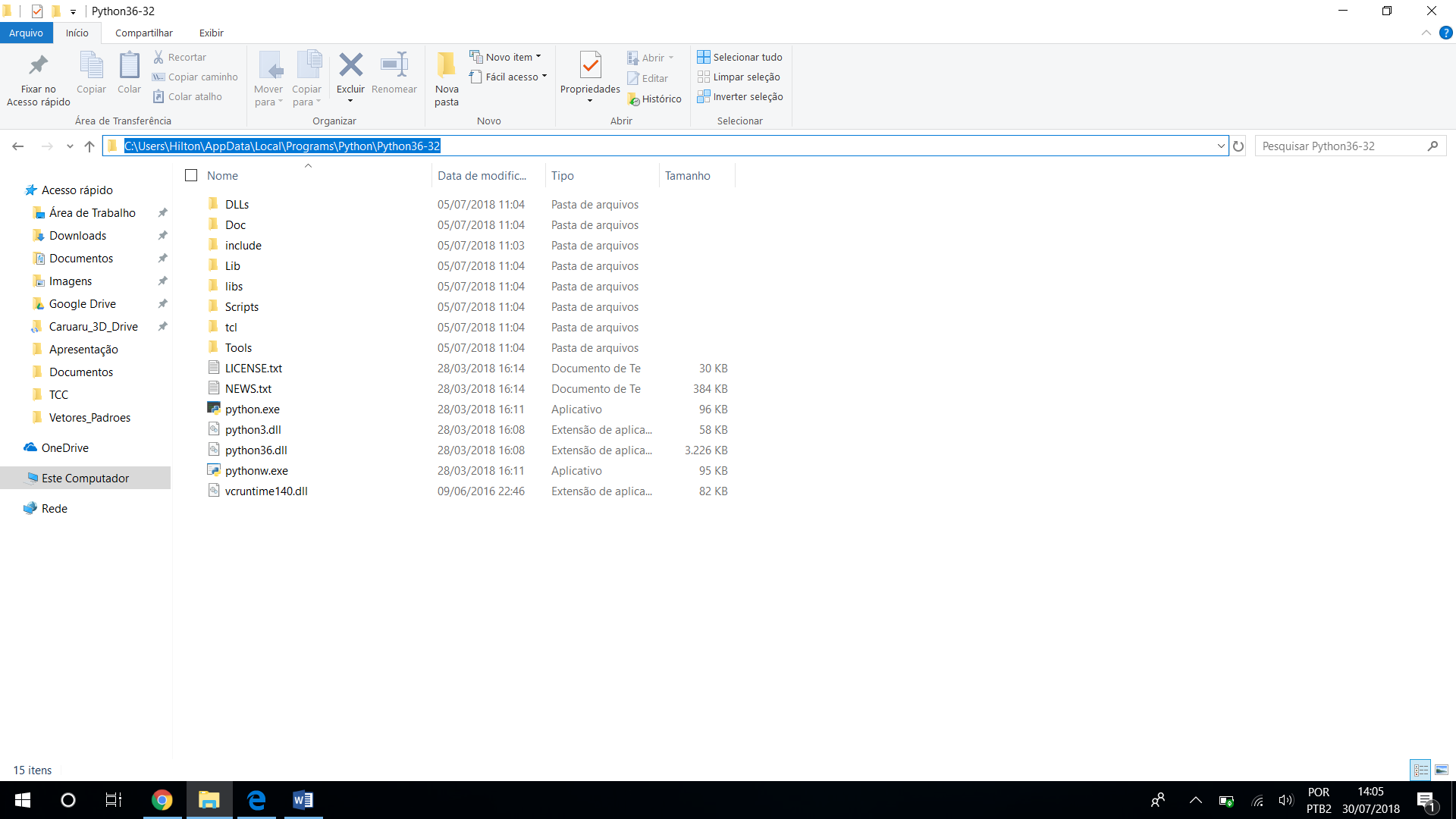
1.1) O código foi feito utilizando a versão 3.6.4 do python. Versões diferentes podem resultar em erros de sintaxe ou em não funcionamento de algumas funções, portanto observar a compatibilidade entre o código e a versão do python.

1.2) Configurar o computador para poder realizar o posterior download e instalação das bibliotecas

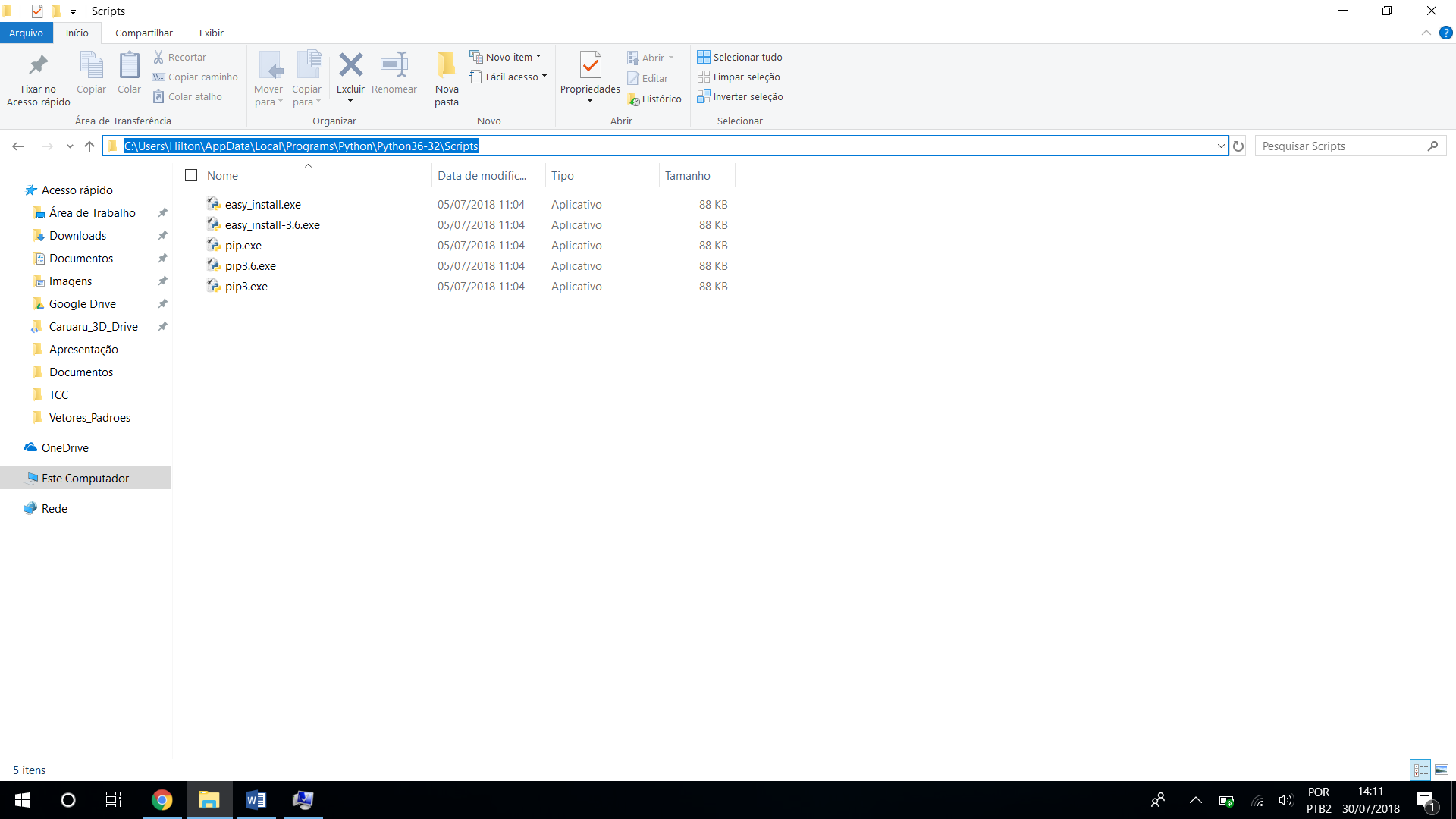
1.2.1) Pegar o endereço do local onde o python está salvo. Geralmente o python está salvo

em um endereço do tipo C:\Users\Livia\AppData\Programs\Python\Python36-32

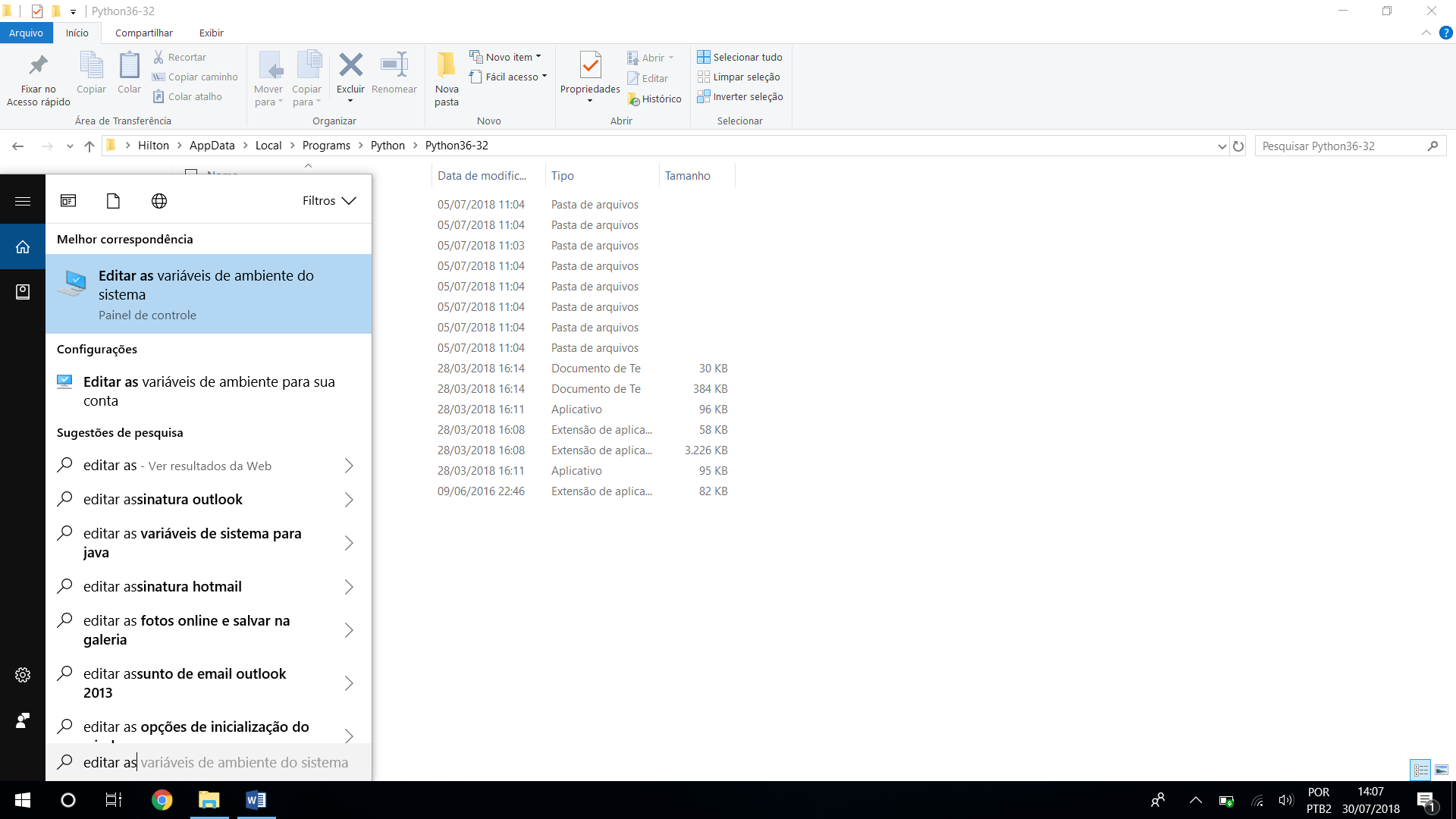
obs: O “Livia” é relativo ao meu pc, cada um tem um nome específico.



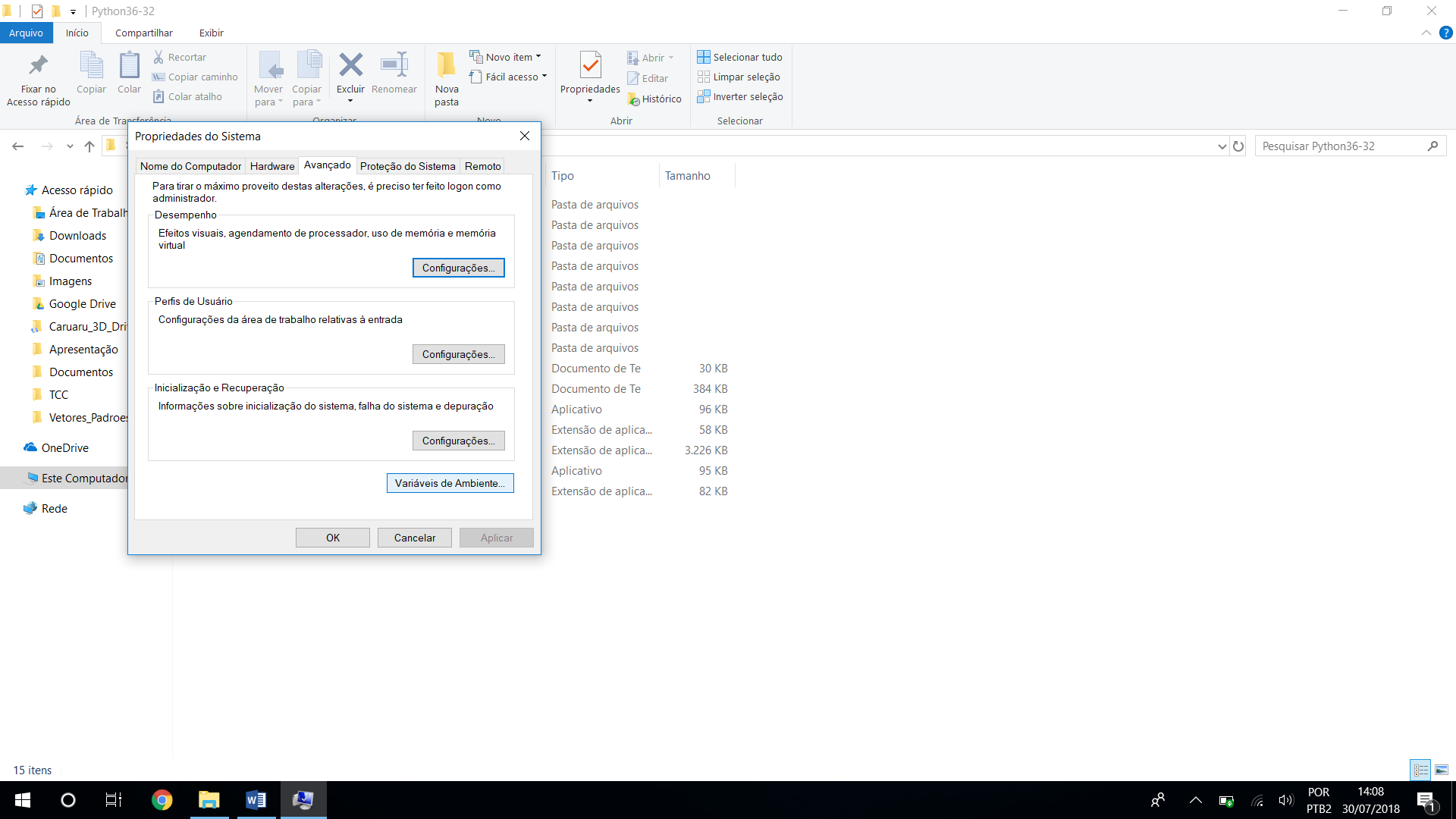
1.2.2) Pegar também o endereço da pasta “Scripts”, a qual se encontra dentro da pasta Python 36-32. Geralmente o endereço é C:\Users\Livia\AppData\Programs\Python\Python36-32\Scripts



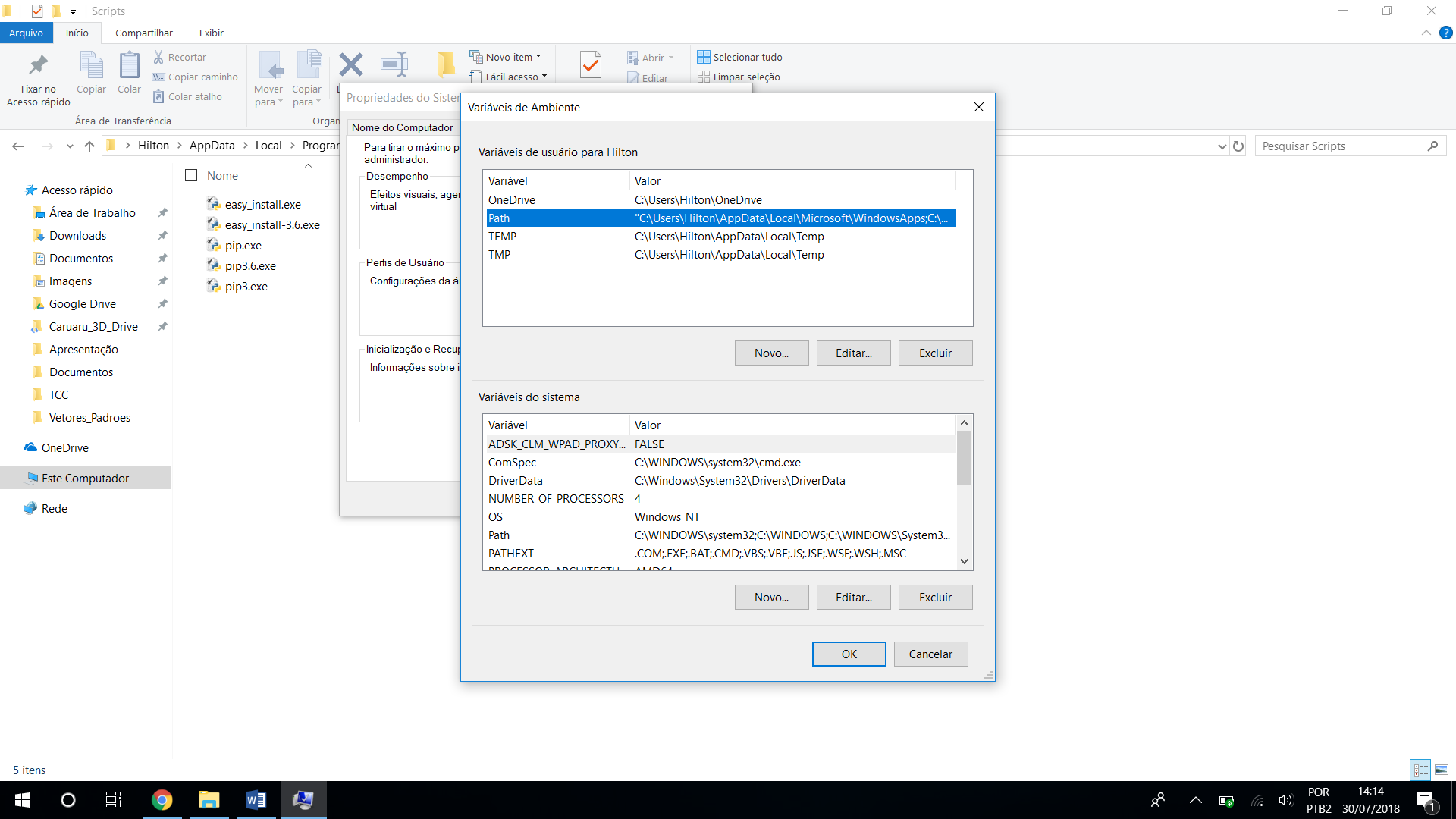
1.2.4) Ir em -> Editar as variáveis de ambiente do sistema



1.2.5) Depois ir em -> Variáveis de Ambiente

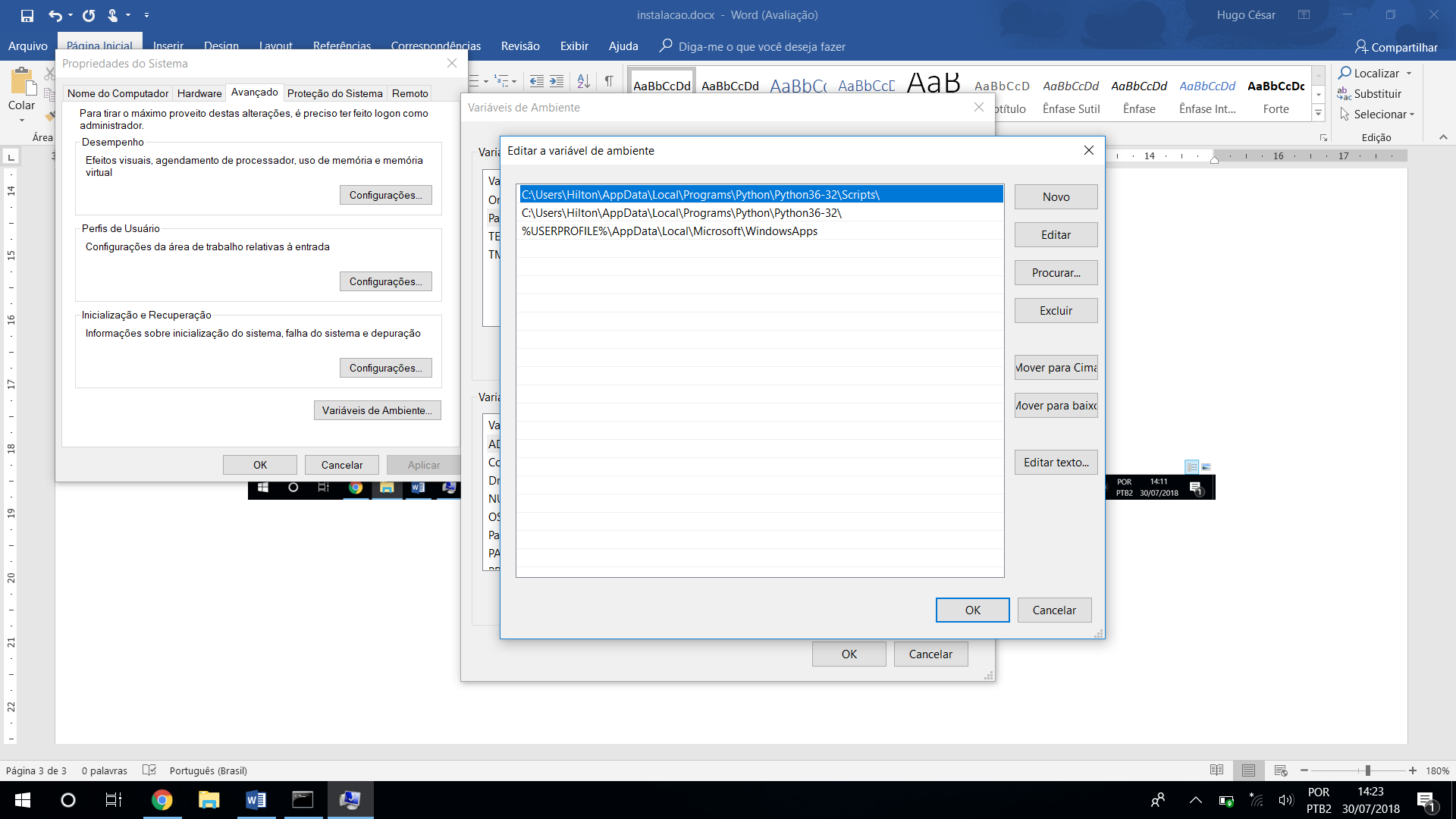


1.2.6) Depois ir em -> Variáveis de usuário -> Path -> Editar



1.2.7) Adicionar os dois caminhos (Python e Scripts)

Obs: Colocar uma \ ao final do caminho, conforme mostrado no print abaixo.

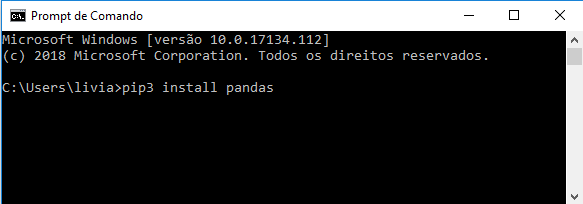


1.3) A execução do código requer a instalação das seguintes bibliotecas: pandas, utm, scipy, numpy.

1.3.1) Instalar as bibliotecas no python

1.3.1.1) Abrir o prompt de comando (digitar "cmd" na barra de pesquisa do windowns)

1. 3.1.2) Escrever pip3 install scipy/pandas/utm/numpy (o nome de biblioteca, uma de cada vez) apertar o botão “*Enter*”



1.4) Salvar o código em uma pasta. O código vai trabalhar procurando arquivos e subpastas dentro dessa pasta. Dessa forma, evita ter que colocar o caminho completo do diretório no código. Para facilitar, sugiro que crie duas pastas: dados e resultados. Prefira sempre nomes e caminhos curtos, simples e sem caracteres especiais.

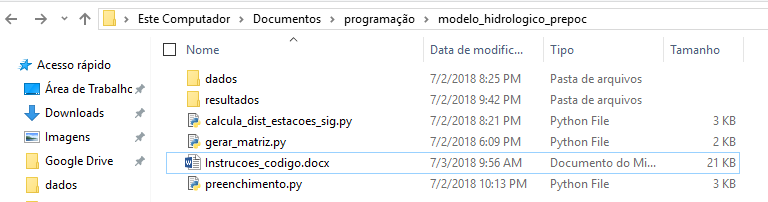


Figura 1- Diretório onde que os códigos estão salvos

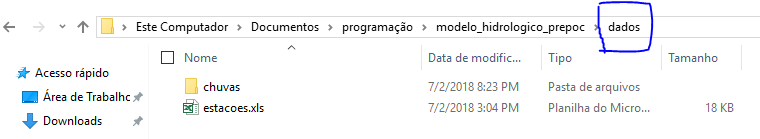


Figura 2- Arquivos dentro do subdiretório dados

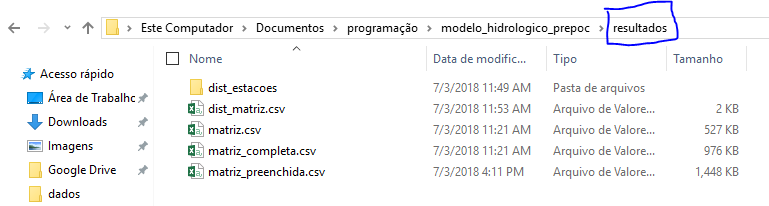


Figura 3- Figura 2- Arquivos dentro do subdiretório resultados

1.5) Fazer uma análise dos dados dos postos. Nos meus dados, eu achei diversos postos que possuíam nomes (códigos) diferentes, mas as coordenadas bem como os dados de chuvas eram os mesmos. Nesse caso, deve-se escolher apenas um posto. No caso de postos que mudaram de nome e possuem séries complementares, deve-se unir os dados em uma única estação com um único código.

2) **Código para gerar matriz com todas as estações: gerar\_matriz.py**

2.1) Baixar os dados do hidroweb no formato .cvs e extraí-los em uma subpasta dentro da pasta em que o código está salvo. Você pode colocar em outra pasta, mas aí terá que colocar o caminho completo no código.

2.2) Evite abrir o csv das estações e salvar qualquer alteração antes de rodar o código, isso pode mudar o encode do arquivo e causar erros.

2.3) Na linha 7 (diretorio\_estacoes = 'dados/chuvas'), inserir o diretório dos arquivos baixados. Se os arquivos estiverem no mesmo diretório do código, basta colocar o nome/caminho da pasta. Nesse caso, os arquivos estão na pasta chuvas, a qual está dentro da pasta dados (ver figura 1).

2.4) Na linha 9 (diretorio\_resultados = 'resultados'), inserir o diretório onde a matriz gerada será salva.

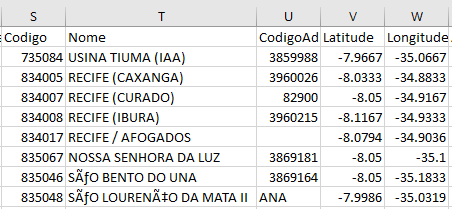
2.5) Na linha 63 (matriz\_completa.loc[ date(1968,6, 1): date(2018,1,1) ].to\_csv(diretorio\_resultados+'/matriz.csv'), definir o intervalo de tempo desejado e o nome do arquivo. A data está no formato (ano, mês, dia).

2.6) O código vai gerar como resultados uma matriz contendo os dados das estações no intervalo de tempo definido ( arquivo sob o nome de “matriz”), e uma outra matriz com todos os dados para todos os anos de operação da estação. O arquivo gerado estará salvo no diretório definido no item 2.4 sob o nome de “matriz\_completa”. O arquivo “matriz” será utilizado como dado de entrada no código de preenchimento de falhas.

3) **Código para calcular a distância entre as estações: calcula\_dist\_estacoes\_sig.py**

3.1) Esse código vai gerar um arquivo .csv contendo a matriz de distância entre as estações, gerando o arquivo “dist\_matriz” e mais um conjunto de arquivos .csv que contem pares de distâncias entre duas estações. Esses últimos arquivos terão como nome o código da estação e deverão ser salvos em um diretório. No meu caso, salvei em uma pasta chamada “dist\_estacoes” dentro da paste “resultados”. Olhar a figura 3.

3.1) Deve-se criar um arquivo .csv contendo colunas com as seguintes informações: código, a latitude e longitude da estação.



Você pode criar uma tabela no excel com essas informações ou pode extrair a tabela de atributos do qgis/arcgis. O código vai rastrear as colunas com os nomes “Codigo”, “Latitude” e “Longitude”, por isso é importante que o nome dessas colunas esteja escrito dessa maneira. As demais colunas (se houver) serão ignoradas.

Obs1: atentar para existência de duas estações com a mesma coordenada.

Obs2: serão utilizadas as coordenas geográficas no formato grau decimal (formato comumente utilizado nas tabelas de atributos). O código realizará a transformação para coordenadas planas.

3.2) Na linha 9 (nome\_arquivo = "estacoes.csv"), declarar o nome do arquivo que contém as informações mencionadas no item anterior.

3.3) Na linha 11 (diretorio\_estacoes = 'dados'), declarar o diretório onde o arquivo está salvo.

3.4) Na linha 13 (diretorio\_resultados = 'resultados'), definir o diretório onde a matriz de distância será salva.

3.5) Na linha 14 (diretorio\_distancias = 'resultados/dist\_estacoes'), definir o diretório onde será salvo o conjunto de arquivos .csv que contem pares de distâncias entre duas estações. Esse arquivo será utilizado como entrada para o código de preenchimento de falhas.

3.6) Na linha 16 (dados = pd.read\_csv(diretorio\_estacoes+'/'+nome\_arquivo, delimiter=',', decimal='.'), definir o formato no qual o seu arquivo de entrada está configurado (separador de coluna e separador decimal).

Obs1: Esse não é o formato que seu computador está configurado, é a configuração do seu arquivo. A configuração pode ser conferida abrindo o arquivo em um bloco de notas.

3.7) Na linha 37 (dist\_matrix.to\_csv(diretorio\_resultados+'/dist\_matriz.csv',decimal='.') definir qual o separador desejado para o arquivo “dist\_matriz” – arquivo que contem a matriz de distância entra todas as estações.

3.8) Na linha 55 (estacoes\_proximas.to\_csv(diretorio\_distancias +'/'+str(estacao)+'.csv', decimal='.'), definir qual o separador decimal desejado.

3.8) Na linha 56 (estacoes\_proximas.to\_csv(diretorio\_distancias +'/'+str(estacao)+'.csv', decimal='.'), definir qual o separador decimal desejado.

4) **Código para preenchimento das falhas das estações: preenchimento\_teste.py**

4.1) Esse código vai gerar três arquivos:

- matriz\_preenchida: contendo todos os dados, ou seja, tanto os que o código conseguiu preencher quanto os que o código não conseguiu preencher

- matriz\_preenchida\_válidos: contendo apenas dados de dias em que todas as estações foram preenchidas

- matriz\_preenchida\_inválidos: contendo dados de dias em que pelo menos uma estação não foi preenchida

4.2) Na linha 9 (nome\_arquivo = "matriz.csv") declarar o nome do arquivo que contém os dados das estações no intervalo de tempo definido (resultado do código do item 3). Esse arquivo é a matriz que será preenchida.

4.3) Na linha 11 (diretorio\_entradas = 'resultados'), definir o diretório em que o arquivo a ser preenchido se encontra (arquivo do item anterior).

4.4) Na linha 13 (diretorio\_resultados = 'resultados'), definir o diretório onde a matriz preenchida será salva.

4.4) Na linha 15 (diretorio\_estacoes\_proximas = 'resultados/dist\_estacoes'), definir o diretório onde está o conjunto de arquivos que contem pares de distâncias entre duas estações – gerado do código do item 3.

5) **Código para visualizar o intervalo de tempo com dados: calcula\_intervalo\_dados.py**

5.1) Esse código vai imprimir o intervalo de tempo em que todas as estações apresentam dados.

5.2) Entrar com a matriz\_preenchida\_válidos

5.3) Intervalo de tempo e número de dias consecutivos com dados

