

Projeto da disciplina Programação Imperativa (COMP0334)

Período: 2019.2 **Turmas:** 05, 06, 08, 09, 12, 13 e 17

Professores: Alberto Costa Neto / Galileu Santos de Jesus / Kalil Araujo Bispo

Artefato de entrega: Programa em Python hospedado no site Repl.it

Data de entrega (link via SIGAA)	Data de demonstração/arguição	Horário	Turma
02/03/2020 até 08:00	02/03/2020 e 04/03/2020	13:00 a 14:40	05
	02/03/2020 e 04/03/2020	18:50 a 20:30	13
	03/03/2020 a 12/03/2020	07:00 a 08:40	06 e 08
	03/03/2020 a 12/03/2020	09:00 a 10:40	09 e 12
27/02/2020 até 08:00	27/02/2020	20:40 a 22:20	17

1. Descrição

A tecnologia para geração de energia solar tem se tornado cada vez mais barata e acessível, atraindo consumidores que desejam reduzir gastos e ao mesmo tempo contribuir com o uso de uma energia limpa.

Para que o consumidor possa acompanhar o funcionamento de sua usina, existem sistemas de monitoramento *on-line*, que recebem dados enviados por *data loggers* periodicamente, ou seja, o *status* de funcionamento da usina solar fica registrado.

Estes dados podem ser analisados e comparados mais facilmente por meio de gráficos. Seu trabalho é desenvolver algoritmos que analisem dados de um arquivo contendo a potência gerada durante os dias e produzir alguns gráficos, conforme modelos apresentados a seguir:

Gráfico de Linha

O gráfico a ser gerado deve se assemelhar aos apresentados abaixo, que exibe os dados do dia 26/01/2019 plotados no gráfico em intervalos de 5 min.

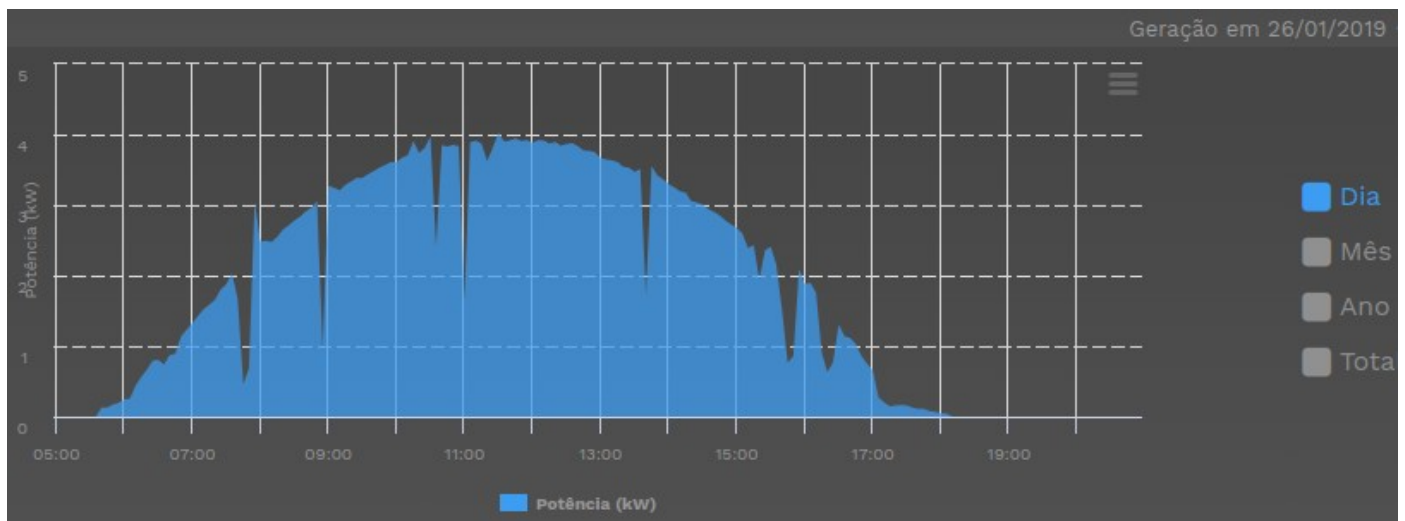


Figura 1

Outro exemplo deste gráfico encontra-se abaixo, exibindo os dados referentes ao dia 28/08/2019.

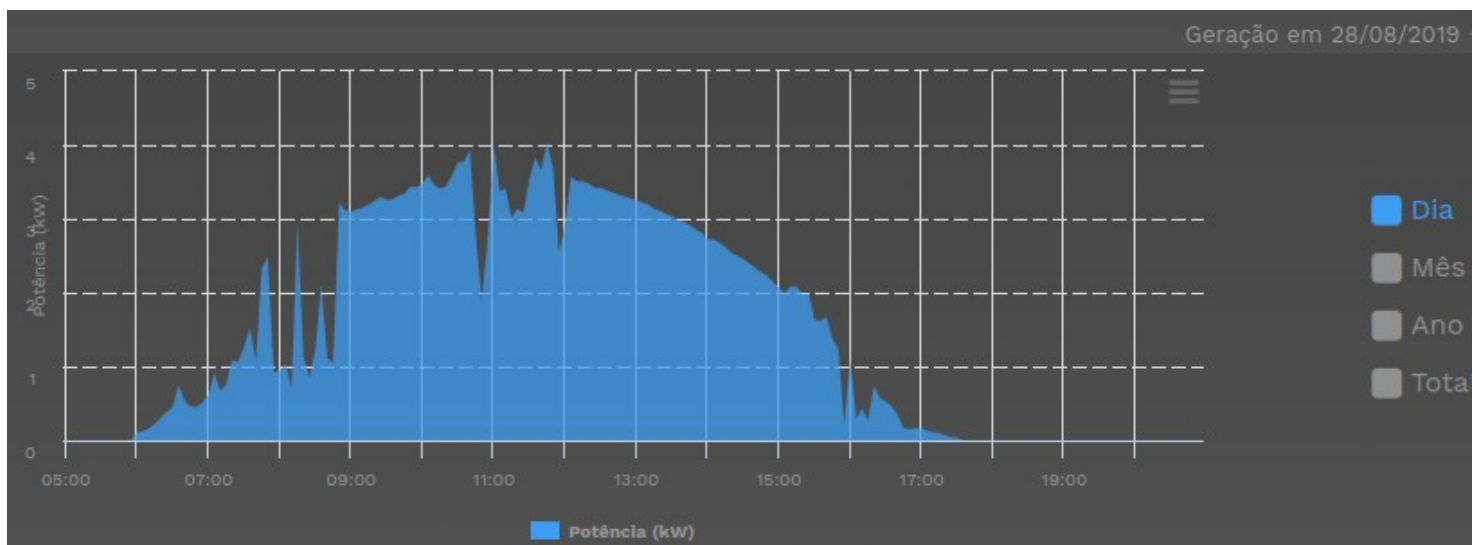


Figura 2

Gráfico de Barras

Um exemplo de gráfico de barras é o gráfico mensal, que exibe os dados de um determinado mês. Cada barra vertical consiste do total de energia gerada em um dia do mês. Observe que o mês pode ter de 28 a 31 dias e que também não há garantia de que há geração de energia todos os dias do mês.

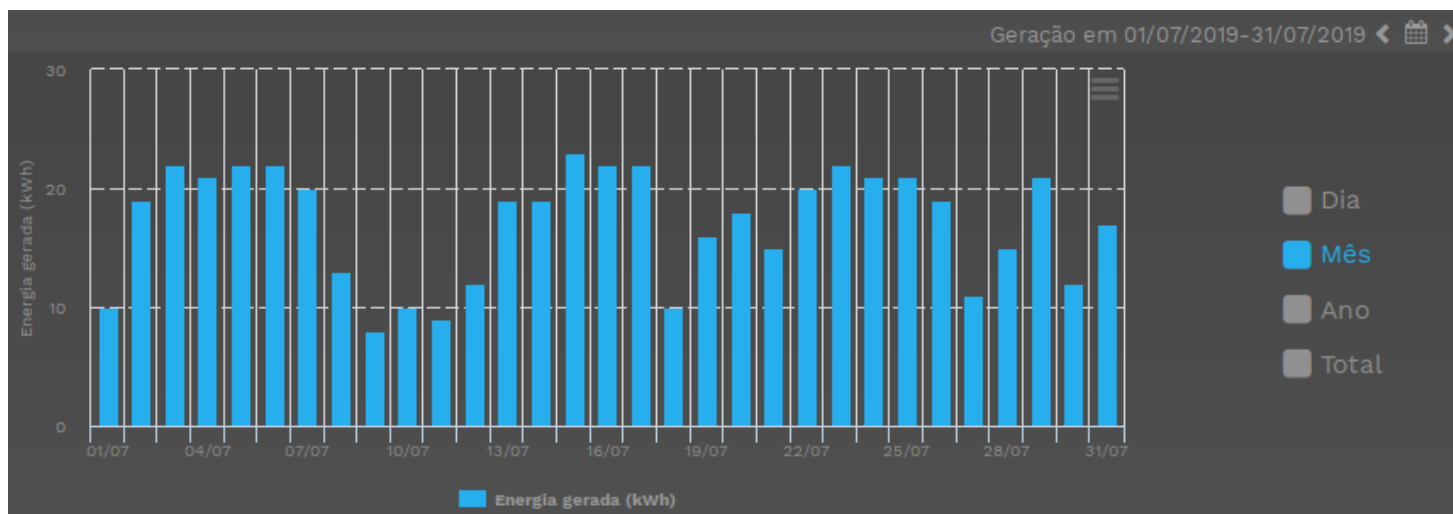


Figura 3

Outro exemplo de gráfico de barras é o gráfico de produção mensal de energia em um determinado ano. Por exemplo, o gráfico abaixo é referente ao ano de 2019.

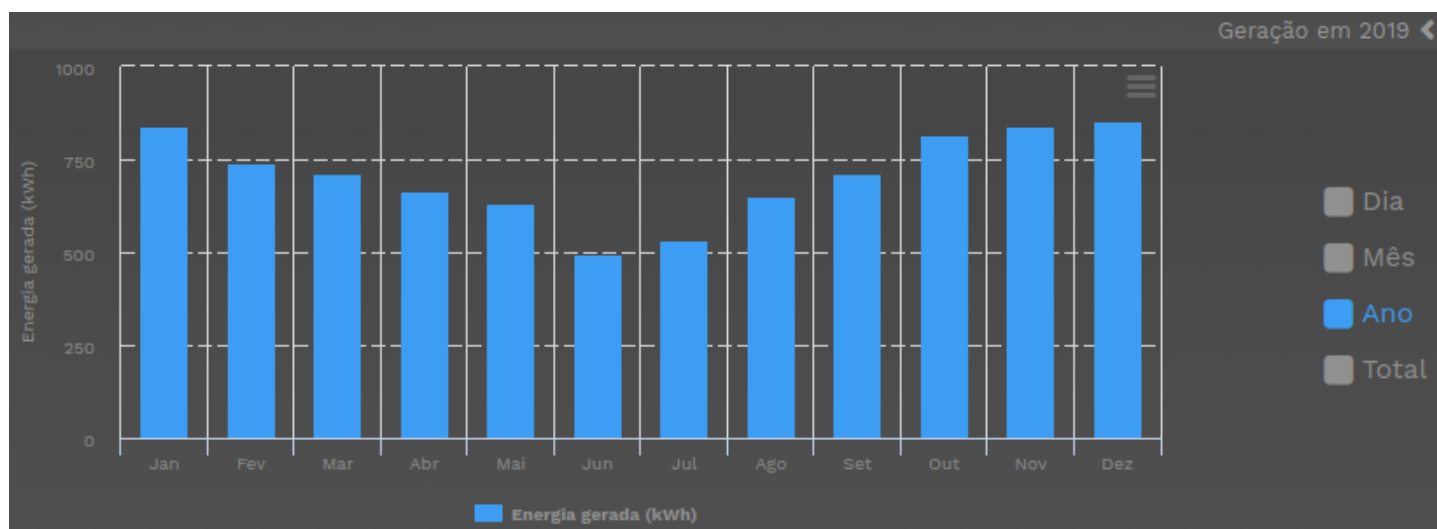


Figura 4

Gráfico Boxplot

Um exemplo de gráfico boxplot é o gráfico semestral abaixo, no qual cada caixa na vertical representa a concentração dos dados referentes aos dias de um mês. Este gráfico serve para visualizar quão estável é a geração de energia ao longo dos meses. Espera-se, por exemplo, que meses do verão (mês 1) tenham maiores valores e menos variação ao longo dos dias, quando comparado com outros meses (mês 6, por exemplo).

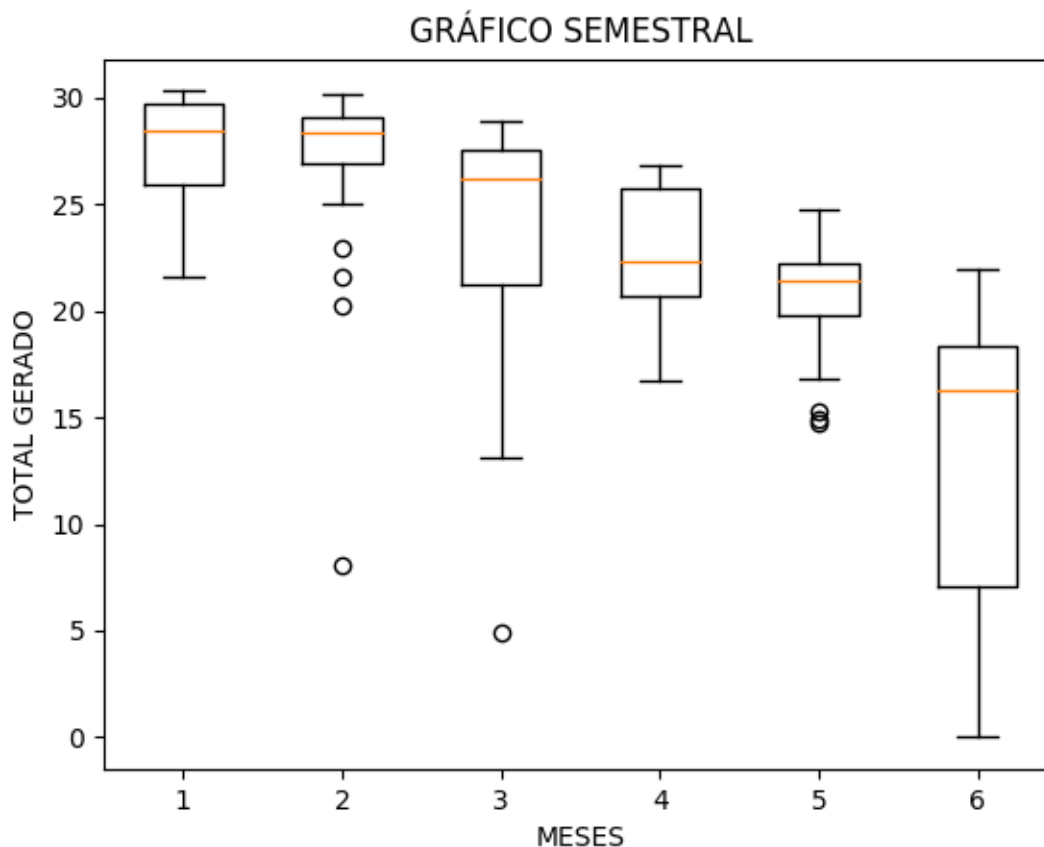


Figura 5

Para mais detalhes de como é construído este gráfico, sugerimos assistir ao seguinte vídeo:

https://youtu.be/S6x_fjofxM

DICA: Para aprender mais sobre a biblioteca Matplotlib de Python, acesso o link a seguir:

<https://matplotlib.org/>

<https://paulovasconcellos.com.br/15-comandos-de-matplotlib-que-talvez-você-não-conheça-17cf88a75119>

ATENÇÃO: O seu projeto deve seguir o template abaixo:

<https://repl.it/@kalilbispo/ProjetoPI-20192>

2. Coleta de Dados

Os dados referentes à geração de energia solar de todos os dias estão disponíveis para download em um servidor Web. O hyperlink para o arquivo encontra-se a seguir.

http://albertocn.sytes.net/2019-2/pi/projeto/geracao_energia.json

Formato do arquivo (resumido com apenas 2 dias de geração de energia):

```
[
  {
    "dia": "2019-01-26",
    "potencia": [-1, -1, -1, -1, 0, 0, 0, 0, 131, 127, 168, 197, 244, 258,
439, 562, 666, 793, 806, 739, 875, 893, 1136, 1234, 1324, 1424, 1522, 1583,
1661, 1800, 1880, 2033, 1669, 458, 680, 3051, 2484, 2493, 2476, 2550, 2655,
2716, 2783, 2834, 2910, 2963, 3055, 957, 3277, 3242, 3209, 3284, 3333, 3384,
3383, 3438, 3480, 3529, 3569, 3612, 3610, 3671, 3710, 3907, 3740, 3812, 3972,
2408, 3850, 3827, 3855, 3836, 1648, 3888, 3913, 3869, 3624, 3816, 4025, 3902,
3915, 3944, 3910, 3920, 3883, 3922, 3917, 3875, 3899, 3842, 3863, 3880, 3837,
3777, 3768, 3753, 3666, 3647, 3636, 3609, 3545, 3531, 3472, 3515, 1726, 3554,
3421, 3366, 3298, 3253, 3198, 3177, 3057, 3035, 3005, 2945, 2902, 2853, 2782,
2727, 2678, 2595, 2389, 2431, 1945, 2359, 2406, 2150, 1513, 764, 861, 2089,
1886, 1895, 1754, 907, 629, 772, 1307, 1140, 1114, 1029, 858, 750, 661, 274,
201, 149, 158, 164, 157, 131, 111, 109, 82, 67, 47, 41, 0, -1, -1, -1, -1, -1,
-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
-1, -1, -1, -1, -1, -1, -1, -1, -1],
    "energiaDia": 29,
    "economiaDia": 22.28708
  },
  {
    "dia": "2019-08-28",
    "potencia": [-1, -1, -1, -1, -1, -1, 0, 0, 0, 0, 0, 0, 110, 125, 159,
227, 305, 389, 454, 747, 533, 457, 450, 512, 612, 911, 669, 759, 1086, 1053,
1270, 1516, 1058, 2332, 2480, 923, 943, 1026, 694, 2949, 1123, 848, 1227,
2107, 1112, 1047, 3222, 3117, 3100, 3127, 3155, 3196, 3242, 3299, 3257, 3269,
3312, 3340, 3442, 3425, 3475, 3592, 3455, 3409, 3442, 3584, 3764, 3776, 3929,
2772, 1880, 2728, 4004, 3371, 3414, 3021, 3134, 3081, 3530, 3829, 3659, 4028,
3677, 2548, 2921, 3566, 3505, 3505, 3481, 3412, 3427, 3382, -1, -1, -1, -1, -1,
-1, -1, 3194, 3135, 3111, 3067, 3039, 2982, 2955, 2915, 2859, 2812, 2727, 2722,
2679, 2615, 2537, 2497, 2458, 2398, 2341, 2275, 2236, 2138, 2060, 1989, 2084,
2077, 1974, 1988, 1636, 1614, 1672, 1374, 1239, 210, 1092, 291, 427, 260, 736,
572, 524, 456, 342, 160, 151, 158, 159, 130, 110, 95, 68, 43, 37, 0, -1, -1, -1,
-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1],
    "energiaDia": 24,
    "economiaDia": 18.50328
  }
]
```

O arquivo é gravado no formato JSON, o qual se assemelha à sintaxe usada em Python para definir Listas e Dicionários, tornando fácil entender sua estrutura. Contém uma “lista”, na qual cada elemento é um “dicionário” que representa os dados de um dia. As chaves do dicionário são descritas na tabela abaixo:

Chave	Valor
“dia”	Uma String representando a data à qual estão relacionados os demais dados de geração de energia. Segue o formato ANO-MÊS-DIA, visando facilitar a ordenação e agrupamento pelo critério da data.
“potencia”	Uma lista de valores inteiros que representam a potência em watts gerada naquele instante. Os intervalos são de 5 em 5 min, iniciando às 5:00 e se encerrando às 21:00. O valor -1 indica que o data logger não enviou dados neste instante, portanto o sistema estava sem gerar energia e este valor não deve ser computado no cálculo.
“energiaDia”	Um valor inteiro que corresponde ao total de energia gerado no dia (em Kwh).
“economiaDia”	Um valor real que indica o valor economizado no dia (em R\$).

Para ver exemplo de como baixar um arquivo JSON de um servidor Web via HTTP e fazer um processamento simples sobre o mesmo, veja o seguinte exemplo:

<https://repl.it/@albertocn/BaixarHTTPProcessarJSON>

Para mais detalhes sobre a API JSON de Python, acesse:

<https://docs.python.org/3/library/json.html>

3. Opções de Gráficos

Cada dupla deverá montar um gráfico de cada tipo (linha, barra e boxplot), conforme a escolha feita pelo(s) professor(es) da disciplina.

Tipo do Gráfico	Descrição
Linha (L1)	Mostra a potência de geração (watts) ao longo do dia, em intervalos de 5 min (Figuras 1 e 2)
Linha (L2)	Mostra a quantidade de energia gerada por dia (kwh) ao longo de um mês completo.
Linha (L3)	Mostra a quantidade de energia gerada por dia (kwh) ao longo de um ano completo.
Barra (B1)	Mostra os totais de geração diária de energia de um mês. Cada barra mostra o total de energia gerada (kwh) em um dia do mês (Figura 3).
Barra (B2)	Mostra o total de energia gerada em cada mês (kwh) de um certo ano. Para isso será preciso fazer o somatório do total de energia gerada em cada dia do mês correspondente a cada barra do gráfico (Figura 4).
Barra (B3)	Mostra a potência média de geração (watts) em intervalos de 60 min para um certo dia. Para isso precisa tirar a média das potências médias a cada 5 min em cada intervalo de 60 min. Por exemplo, a primeira barra representa a média entre 05:00 e 06:00. A segunda das 06:00 às 07:00 e assim por diante até finalizar com a barra de 20:00 a 21:00.

Boxplot (X1)	Mostra a quantidade de energia gerada diariamente ao longo de um semestre, organizando os valores diários de um mês em um boxplot, totalizando 6 boxplots (um para cada mês do semestre, como visto na Figura 5)
Boxplot (X2)	Mostra a quantidade de energia gerada diariamente ao longo de um ano, organizando os valores diários de um mês em um boxplot, totalizando 12 boxplots (um para cada mês do semestre)
Boxplot (X3)	Mostra a quantidade de energia gerada diariamente ao longo de cada estação do ano, organizando os valores diários de uma estação em um boxplot, totalizando 4 boxplots (um para cada estação do ano)

4. Gravação de Dados

Os gráficos devem ser gerados e gravados no próprio Repl.it, conforme foi exemplificado. Qualquer interação do programa com o usuário deverá ser através da entrada e saída padrão do Python. Portanto, procure dar mensagens e orientação de o uso no seu programa.

5. Critérios de Avaliação

O projeto irá ser avaliado considerando vários critérios, dentro os quais relacionamos:

- 1) **Eficácia:** Se atinge os objetivos propostos, ou seja, gera os gráficos corretamente;
- 2) **Eficiência:** Se atinge os objetivos propostos de uma maneira que utilize poucos recursos computacionais e funcione de forma rápida;
- 3) **Qualidade do código:** O código fonte produzido utiliza nomes adequados para variáveis, funções e outros elementos, se foram utilizadas funções com parâmetros adequados para modularizar o código e promover reuso. Além disso, demonstrar domínio no conhecimento que foi adquirido durante o curso (por exemplo, usar Listas e Dicionários de forma adequada).
- 4) **Comentários:** Contém comentários que facilitam o entendimento do código fonte, evitando ser prolixo com comentários óbvios e desnecessários.

6. Nota

O projeto vale 10,0 (dez) pontos, será em dupla, mas a nota será individual. Para receber a nota, a dupla terá que apresentar o projeto ao(s) professor(es) e explicar como tudo foi feito;

- Caso o aluno faça o projeto individualmente, terá apenas um corte de 20% da nota;