Projeto da disciplina Programação Imperativa (COMP0334)

Período: 2019.1 **Turmas:** 02, 03, 04 e 05

Professores: Alberto Costa Neto / Kalil Araujo Bispo

Data de conclusão: 19/08/2019

Data de entrega (link via SIGAA): 18/08/2019

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

Data de demonstração/arguição: 19/08/2019 (turma 04 e 03) e 20/08/2019 (turmas 02 e 05)

1. Descrição

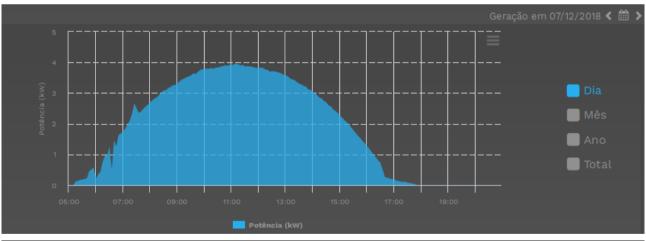
A tecnologia para geração de energia solar tem se tornado cada vez mais barata e acessível, atraindo consumidores que desejam reduzir gastos e ao mesmo tempo contribuir com o uso de uma energia limpa.

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 pegar arquivos contendo a potência geração durante o dia e produzir alguns gráficos como os apresentados a seguir.

Gráfico Diário (Gráfico de Linha)

O gráfico a ser gerado deve se assemelhar aos apresentados abaixo, que exibem, respectivamente os dados dos dias 07/12/2018 e 18/07/2019 plotados no gráfico em intervalos de 5 min.



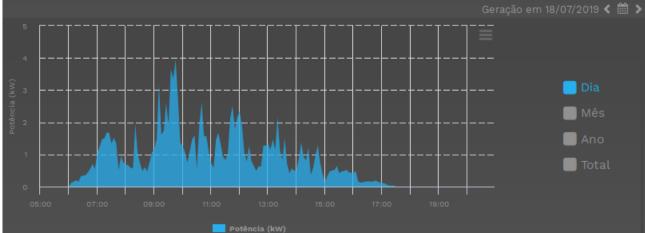


Gráfico Mensal (Gráfico de Barras)

O gráfico mensal a ser produzido deve ser semelhante ao apresentado abaixo, que exibe os dados de um mês, sendo que cada barra vertical consiste de um dia. Observe que o mês pode ter de 28 a 31 dias e que também não há garantia de que todos os dias têm arquivos de log.

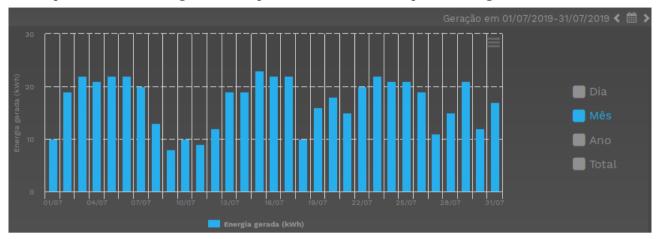
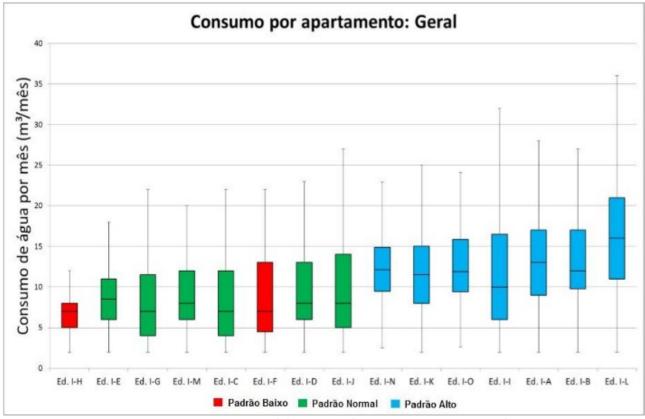


Gráfico Semestral (Boxplot)

O gráfico semestral deve ser do tipo boxplot, no qual cada caixa na vertical representa os 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 tenham maiores valores e menos variação ao longo dos dias, quando comparado com outros meses.



Fonte: http://www.ufjf.br/srhps/files/2018/09/A0002.pdf

DICA: Um ponto de partida para construção de gráficos, usando a biblioteca Matplotlib de Python, pode ser encontrado no seguinte endereço:

https://repl.it/@kalilbispo/PI-20191

2. Coleta de Dados

Os dados referentes à geração de energia solar de um dia estão armazenados em um servidor Web, disponíveis para download.

Cada arquivo segue o padrão de nomenclatura "dia_2019_01_31.json", no qual 2019 refere-se ao ano, 01 ao mês e 31 ao dia.

http://albertocn.sytes.net/2019-1/pi/projeto/dia 2019 01 31.json

Formato do arquivo:

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 um "dicionário", com as chaves mostradas na tabela abaixo:

Chave	Valor
"potência"	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\$).
"sucesso"	Indica se houve geração de energia no dia.

Uma lista com todos os dias para os quais há dados de geração de energia solar está contida em um outro arquivo, chamado "arquivos.json". Para baixar este arquivo, utilize a URL abaixo:

http://albertocn.sytes.net/2019-1/pi/projeto/arguivos.json

Formato do arquivo:

```
["dia_2019_01_31.json", "dia_2019_01_01.json", "dia_2019_01_02.json"]
```

Os dados também são gravados no formato JSON. Neste caso, contém uma lista com os nomes dos arquivos armazenados no servidor Web, sendo a única forma de se saber para quais dias há dados disponíveis.

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. 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.

4. 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.

5. Nota

Em relação à nota da 3º unidade das turmas 02, 03, 04 e 05 de Programação Imperativa, os professores Alberto e Kalil decidiram que:

- O aluno pode escolher entre fazer:
 - **Fazer apenas a 3º prova:** a nota obtida servirá também para repor uma nota baixa da 1º ou 2º unidade. É a situação prevista no plano de ensino;
 - **Fazer apenas o projeto:** neste caso a nota da 3º unidade será definida a do projeto, mas não servirá como repositiva da 1º ou 2º unidade;
 - **Fazer ambos:** esta última situação é mais interessante, tanto em termos de aprendizado como possibilidade de melhoria da nota, já que a 3º prova servirá como repositiva da 1º ou 2º nota e a nota da 3º unidade será a nota do projeto.
- 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 pofessor e explicar como tudo foi feito;
- Caso o aluno faça o projeto individualmente, terá apenas um corte de 20% da nota;
- O projeto e a prova da 3º unidade envolvem todos os conceitos vistos durante a disciplina.