

# Relatório PBL problema 1: Segue o seco

Samuel da Costa Araújo Nunes<sup>1</sup>

<sup>1</sup>Engenharia de Computação – Universidade Estadual de Feira de Santana (UEFS)  
Feira de Santana, 06 de abril de 2019

samuelnunes1920@outlook.com

**Resumo.** *Este relatório descreve a confecção de um programa que calcula e mostra pro usuário o consumo em kWh e o gasto em Reais, de uma instituição, considerando os setores, os aparelhos de cada setor e as informações principais desses aparelhos, e todos os problemas e soluções que apareceram com o seu desenvolvimento.*

## 1. Introdução

A seca é um fenômeno que se caracteriza pela ausência, quantidade limitada ou uma gestão ineficaz das fontes hídricas, por sua vez, causa fortes impactos na sociedade, na política e na economia do país.

Em 2014, O estado de São Paulo passou pela crise hídrica mais intensa já registrada na região. De acordo com a ONU (Organização das Nações Unidas), a principal responsabilidade sobre a crise, é da ANA (Agência Nacional de Águas) e do Estado de São Paulo por não realizarem investimentos o bastante para que todos os habitantes tivessem acesso a água potável e ao saneamento básico.[Dias 2016]

A má administração dos recursos hídricos nacionais, recai diretamente sobre o valor da energia elétrica. Ao analisar esses dados, o Reitor da UEFS (Universidade Estadual de Feira de Santana), solicitou aos alunos matriculados no MI de Algoritmos que desenvolvessem um simulador de consumo de energia para a Universidade.

Para o desenvolvimento do programa solicitado pelo reitor, foram utilizadas as estruturas da linguagem Python, especialmente, os comandos para entrada e saída de dados para armazenar as informações de cada Setor, além de estruturas condicionais e laços de repetição, utilizadas para repetir as perguntas em função da quantidade de setores avaliados. O software proposto foi desenvolvido na IDE Pycharm.

## 2. Desenvolvimento

Ao iniciar o desenvolvimento do simulador, foi analisado os critérios básicos para o funcionamento do programa. Logo, surgiu o primeiro problema. Como coletar dados do Usuário, usando a linguagem *Python*? Através da leitura do livro Curso Básico de Python 3 [Álan Crístopher e Sousa 2017] e das primeiras aulas de Algoritmos I, foi apresentado as funções *Print* e *Input*, usadas para imprimir e coletar os dados, respectivamente.

Foram implementadas todas as perguntas a respeito do Setor, sempre associando as informações coletadas as variáveis. O segundo problema surgiu. Como coletar as informações de todos os setores, sem copiar e colar as linhas de código, obedecendo a quantidade de setores definida pelo Usuário? A solução achada inicialmente foi a

utilização do laço de repetição *for*, que atende a esses critérios, porém o usuário não tinha como encerrar o programa quando quisesse, contrariando a solicitação do reitor.

O laço *for*, quando compilado, gera uma lista com toda range. Assim, as funções e variáveis dentro do laço não poderiam influenciar as condições do *for*. Sendo substituído pelo laço *While*. Que consegue atender ao problema das duas condições do reitor.

Ao incluir o laço de repetição, o programa já estava coletando os dados de todos os setores, porém, gerou um novo problema, a cada *loop*, o *input* sobrescrevia as informações do setor atual pelo anterior, e no final. Ele printava apenas as informações do último setor. Foi desenvolvida duas soluções ao problema, a primeira(Ex1) era incrementar uma variável com o valor do setor atual, e a cada loop a variável seria igual a soma das informações de todos os setores, atendendo perfeitamente ao pedido do reitor. Mas, afim de printar as informações de cada setor individualmente ao final do programa, foi utilizado as listas, que a cada loop, adicionava a informação dada na casa seguinte.

Ex1:

```
consumo_total = consumo_total + consumosetor
```

Dentro do laço `while(i < setores)` : foi utilizado as funções de operações matemáticas do Python, para calcular o Custo de cada setor(Ex2), considerando a quantidade de cada aparelho, a potencia deles, quantidade de horas que funcionam e por quantos dias.

Ex2:

```
ConsAR.append(arQHD[i] * arQDM[i] * arP[i] * ar[i])
```

A equação acima, associa em uma lista, o valor do consumo do ar condicionado por setor. Sendo *arQHD*, a quantidade de horas por dia, *arQDM*, a quantidade de dias por mês, *arP*, a potência, *ar*, a quantidade de aparelhos e *i*, o setor referente.

Por fim, o programa lê um novo *While*, que varia de 0 até a quantidade de setores lidos, e dentro deste laço, ele imprime o consumo e o valor sobre cada tipo de aparelho de cada setor. Após o final do laço, o programa imprime o consumo total da UEFS, o valor total e o valor total com os impostos.

Para imprimir o valor total com os impostos, criei uma constante que é o resultado final da multiplicação de todos os impostos(Ex3).

Ex3:

- ICMS(27%)
- PIS(1,65%)
- CONFINS(7,61%)

$$1,27*1,0165*1,0761=1,38919$$

### 3. Conclusão

Com o proposto problema aprendi e pratiquei muito os laços de repetição, as estruturas condicionais e os comandos de *input* e *print*. O programa atende a todas as funcionalidades exigidas pelo pedido do reitor, porém é possível melhorar consideravelmente o programa considerando as necessidades reais das pessoas que iriam utilizar esse serviço.

Inclui no projeto algumas funcionalidades extras, como a possibilidade de mostrar ao usuário os custos e a potência gasta por setor, no final do programa.

Apesar de não ser necessário, queria mostrar ao usuário os valores gastos, apenas com duas casas decimais após a vírgula. Porém, mesmo pesquisando e usando o `%.xf`, não consegui o resultado esperado em nenhuma tentativa.

## References

- Dias, N. (2016). Crise hídrica do sistema cantareira e os direitos de acesso à Água e à informação. In artigo19, editor, *O SISTEMA CANTAREIRA E A CRISE DA ÁGUA EM SÃO PAULO falta de transparência, um problema que persiste*, pages 40–41. artigo19.
- Álan Crístofffer e Sousa (2017). *Curso Básico de Python 3*. PIBIC 2014 FAPEMIG, 1th edition.