

Relatório PBL problema 2: Sistema para o Censo Demográfico de 2020

Samuel da Costa Araújo Nunes¹

¹Engenharia 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 as estatísticas de uma pesquisa do IBGE, a partir da manipulação de três arquivos, que informam os dados sobre as regiões e suas características, os técnicos cadastrados no sistema e a própria pesquisa realizada pelos técnicos.*

1. Introdução

O Instituto Brasileiro de Geografia e Estatística (IBGE), surgiu com o propósito de avaliar, estudar e entender as principais características sobre todo o âmbito nacional. Os censos eram realizados de forma analógica, por essa razão o processo de depuração de dados era muito lento[IBGE 2019]. Com o passar do tempo, a tecnologia foi modernizando o processamento das informações. Hoje, com um pequeno programa é possível economizar verba e tempo, analisando dados brutos e gerando estatísticas quase que instantaneamente.

O IBGE foi fundado por Mário Augusto Teixeira de Freitas, para atuar no campo de geociências e estatísticas sociais, demográficas e econômicas. Assim, Assim, a principal função do IBGE é realizar censos e organizar as informações públicas obtidas em pesquisas com milhões de brasileiros.

A atual tecnologia que o IBGE utiliza permite o cálculo de estatísticas de forma mais prática e eficiente. E assim, é possível disponibilizar essas informações para a população de forma mais rápida, tornando o acesso as informações mais acessíveis.

Para o desenvolvimento do programa solicitado pelo IBGE, foram utilizadas as estruturas da linguagem *Python*, especialmente, os comandos para leitura de arquivos e manipulação de lista, além de funções, a fim de modularizar o código, utilizadas para organizar o código em partes menores e independentes que podem ser reutilizadas em momentos oportunos. O software proposto foi desenvolvido na IDE *Pycharm*.

2. Desenvolvimento

Ao iniciar o desenvolvimento do programa, foi analisado os critérios básicos para o seu funcionamento. Logo, surgiu o primeiro problema. Como coletar dados dos arquivos, usando a linguagem *Python*? Através da leitura do livro *Introdução a algoritmos e programação com Python*, [Wazlawick 2017], foi apresentado funções de manipulação de arquivo nativas, como *Open*, *Readlines*, entre outros, através de pesquisas na internet, também foi descoberto as estruturas de dados: Lista, Tupla e Dicionário[eXcript 2015], usadas para coletar e agrupar dados e as indicações para as suas aplicações.

O primeiro desafio, foi a coleta de dados dos arquivos sem criar variáveis globais, a solução foi a geração de funções, com a finalidade de gerar variáveis locais que trabalhariam com as informações e retornariam para outra função dados já processados. Seguindo este preceito, foram criadas funções para tratamento geral a todos os arquivos, como a função "Separador", que toda vez que era chamada, lia uma linha do arquivo, retirava qualquer formatação e dividia as informações e retornava essas informações contidas em uma lista. Esta função, dentre outras, foram criadas para atender casos gerais, ou seja, com esta função foi possível separar dados dos três arquivos de texto do projeto.

O problema exigia também uma validação entre as matrículas dos técnicos que realizaram a pesquisa e um cadastro prévio dos técnicos, este foi o segundo problema encontrado. Estudando a fundo as estruturas de dados do *Python*, foi encontrado a lista, a Tupla e o Dicionário. Sabendo que este programa poderia receber cargas densas de dados a ser refinado, foi utilizado o Dicionário como ferramenta para a coleta dos dados dos técnicos cadastrados, pois sua estrutura de dados funciona como uma Tabela de dispersão, ou seja, é possível associar dados e informação a uma simples chave, que pode ser acessada de forma rápida pelo sistema. Assim, foi desenvolvido a Função "verificador-interno", que vai adicionando a um dicionário as informações dos técnicos, usando suas matrículas como chave e seus dados como seus valores. Enquanto ele importa os dados fornecidos, a própria função verifica se já existe um técnico com a mesma matrícula do que ele está importando. E apurava possíveis erros, no final retornava um dicionário com todo os técnicos cadastrados.

```
T001;Abel;M;31/01/1962
```

```
T001;Abel;M;31/01/1962
```

```
T002;Abelardo;M;27/06/1958
```

Neste caso, o programa iria indicar a mensagem de erro: A Matrícula T001 foi cadastrada mais de uma vez, porem se existisse duas pessoas diferentes com mesma matrícula:

```
T001;Abel;M;31/01/1962
```

```
T001;Abelardo;M;27/06/1958
```

O console exibiria a mensagem:A Matrícula T001 possui mais de um tecnico cadastrado

Agora, com o uso da função "VerificadorInterno", foi possível verificar a procedência dos técnicos que realizaram a pesquisa. Assim, foi produzido uma nova Função. O "VerificadorPesquisa", que olha linha a linha no arquivo *exemploPesquisa.txt*, os técnicos que aplicaram a pesquisa e perguntam ao sistema se os mesmos, são chaves do dicionário. Se sim, o programa enfim segue para o processamento das estatísticas, se não, ele sinaliza com uma mensagem de erro. As funções de verificação, tiveram seus parâmetros adaptados para também, realizar a verificação das regiões cadastradas. Assim, é possível reaplicar uma função já feita.

Após a verificação dos dados, o programa pode gerar as estatísticas. Primeiro o programa irá printar um menu, que disponibilizará 8 opções de estatísticas, ao escolher uma, será chamada uma função que representa a opção escolhida.

=====

Sistema para o Censo Demografico de 2020

=====

- [1] - N° de domicílios utilizados para a coleta
- [2] - N° de domicílios particulares que já estão pagos, pagando e alugados
- [3] - Quantos domicílios por cidade possuem banheiro e quantos não possuem
- [4] - A forma mais comum de abastecimento de agua por cidade
- [5] - O percentual de domicílios por cidade que ainda não possuem energia elétrica
- [6] - O percentual de moradores que participaram da entrevista por cor ou raça
- [7] - A região com maior número de municípios pesquisados
- [8] - Apresentar todos os dados anteriores

Opção:

Cada alternativa leva a uma função, a primeira, pede a quantidade de domicílios total que participaram da coleta de dados, ao analisar, cada linha da pesquisa se refere aos dados de um domicílios, ou seja, a quantidade total de linhas do arquivo da pesquisa, menos a linha de cabeçalho representa quantidade de domicílios que participaram do censo.

A segunda estatística, chama uma função que observa as informações contidas na coluna da pesquisa referente a quantidade de domicílios pagos, alugados e ainda pagando. Incrementa uma variável para cada opção, e depois de analisar toda a pesquisa, retorna uma Tupla, contendo tais variáveis. Onde são organizadas e printadas ao usuário.

A terceira estatística, chama uma função que gera um dicionário relacionando o código da cidade, em relação ao seu Nome, depois, ele cria dois dicionários, um em relação a quantidade de casas por cidade sem banheiro e a outra com banheiro, durante a leitura do arquivo pesquisa ele divide as informações entre esses dois , relacionando o código da cidade como uma chave do dicionário das cidades, e retorna uma Tupla com os dois dicionários dentro. Após, ele chama uma função que puxa esses valores e os organiza para apresentar ao usuário.

A quarta estatística, possui um funcionamento muito próximo ao da terceira, porém, ela gera uma lista com 10 espaços para cada cidade da pesquisa, onde cada casa é uma possível resposta do questionário, no final do programa, ele pede o índice do maior valor encontrado na lista, ou seja, a forma mais comum de abastecimento desta cidade.

A quinta estatística, realiza o mesmo processo de depuração da quarta, a diferença é que ela gera um valor percentual entre as casas que não possuem energia elétrica per-

centual, é quantidade de casas na cidade "x", que não possuem energia elétrica, sobre a soma total de casas por cidade, e tudo isso multiplicado por 100.

A sexta e a sétima, utilizaram o mesmo raciocínio das questões já abordadas. A que mais se diferenciou foi a sétima, onde foi preciso a construção de 5 listas, definidas pelos estados quem compõe as regiões brasileiras. Assim, durante a varredura de dados do arquivo de pesquisa, há uma verificação da região a qual a cidade pertence, e incrementa uma variável, responsável por contabilizar os dados por região. Esta função retorna uma Tupla contendo a quantidade de cidades pesquisadas nas 5 regiões. O índice com maior valor encontrado dentro da lista, corresponde a região com mais pesquisas.

3. Conclusão

Com o proposto problema aprendi e pratiquei muito as estruturas de dados fornecidas no *Python*, o uso correto da modularização do código, o uso das eficiente das funções. Como um programa piloto, ele atende a todas as funcionalidades exigidas pelo pedido do IBGE, porém é possível melhorar consideravelmente o programa considerando as necessidades reais do órgão.

O foco principal para a construção desse projeto foi facilitar o uso do usuário e apresentar as informações de forma clara, na medida em que o problema proposto possibilitou.

Nos últimos testes, foi observado uma demora na resposta do computador, ao exibir os dados processados da quantidade de casas com acesso a banheiro em uma cidade, tais correções de processamento poderiam ser atualizadas em futuras versões.

References

- eXcript (2015). Funções dos dicionários no python. Disponível em: <http://excript.com/python/funcoes-dicionarios.html>. Acesso em: 18 Jul. 2019.
- IBGE (2019). História das estatísticas brasileiras. Disponível em: <https://memoria.ibge.gov.br/publicacoes/historia-das-estatisticas-brasileiras.html>. Acesso em: 7 Jul. 2019.
- Wazlawick, R. S. (2017). *Introdução a algoritmos e programação com Python*. Elsevier, 2017, 1th edition.