

# Censo Demográfico: Um Sistema Desenvolvido Em Python

Everton Bruno Silva dos Santos<sup>1</sup>

<sup>1</sup>Universidade Estadual de Feira de Santana (UEFS)  
Caixa Postal 44036-900 – Feira de Santana – BA – Brazil  
Av. Transnordestina, s/n, Novo Horizonte

evertonbrunosds@gmail.com

**Resumo.** A reitoria da UEFS, visando auxiliar o IBGE no processo de exibição de dados coletados ao longo de suas pesquisas de campo, solicitou aos alunos do curso de engenharia da computação que desenvolvessem o projeto piloto descrito nesse relatório. Implementada na linguagem de programação Python, essa solução importa os dados contidos em três diferentes arquivos visando processar e exibir os dados requeridos pelo usuário a medida em que desejar. Assim, quando atendidas todas as condições de uso propostas no manual, o software apresenta na tela um menu contendo uma série de opções que podem ser utilizadas para exibir estatísticas de forma precisa e coerente, sendo uma solução eficaz para o problema proposto.

## 1. Introdução

Criado no ano de 1934, ainda no período correspondente a era Vargas, o IBGE (Instituto Brasileiro de Geografia e Estatística) desempenha um papel importante no que se refere a divulgação de informações sobre as condições de vida da população em todo território nacional [IBGE 2017]. Para isso, são feitas coletas de informações em todas as regiões, estados e municípios do Brasil. Levando isso em conta, a reitoria da UEFS solicitou aos alunos do curso de engenharia da computação que implementassem um software para auxiliar o instituto nesse sentido.

Este relatório tem por objetivo descrever o sistema desenvolvido para efetuar a exibição de informações coletadas pelos técnicos do IBGE em suas pesquisas de campo. De todo modo, levando em conta que esse projeto é apenas um piloto para fins de demonstração, mesmo que tenham sido levantadas diversas informações durante as pesquisas, nem todos os dados obtidos serão considerados.

Devido ao fato do levantamento de dados já ter sido feito e armazenado em três diferentes arquivos, cabe ao software permitir importa-los. Sendo que, dois dos deles são usados em síntese para validar os dados contidos em um dos arquivos. Assim com esse processo, o software descrito permite ao usuário utilizador dessa ferramenta visualizar estatísticas de tudo o que foi pesquisado, de uma só vez ou separadamente a medida em que desejar.

Para desenvolver esse sistema, alguns questionamentos foram sanados: 1) Como importar arquivos?; 2) Qual a importância da codificação dos arquivos para o projeto?; 3) Como validar “exemploPesquisa.txt”?; 4) O que é e para que serve uma lista?; 5) O que é e para que serve uma matriz?; 6) O que é e para que serve um dicionário? e 7) O que é e para que serve a modularização?.

A solução é organizada da seguinte forma:

1. A biblioteca *os* é importada, então funções e procedimentos são criados;
2. O procedimento principal é chamado e posteriormente variáveis locais, matrizes, listas e dicionários são criados a medida em que for necessário;
3. Os arquivos situados dentro da pasta “FileBase” como “exemploPesquisa.txt”, “regioes.txt” e “tecnicosIBGE.txt” são inseridos na memória *RAM* do computador para logo se tornam matrizes;
4. Após os arquivos terem se tornado matrizes, ocorre o processo de validação de “exemploPesquisa.txt”;
5. O laço principal é iniciado contendo um menu com nove opções, assim a medida em que o usuário solicita as opções, dados estatísticos são mostrados na tela.

## 2. Desenvolvimento

Ao longo dessa sessão de desenvolvimento serão aprestados os modos como foram tratadas as questões contidas na introdução durante a solução do problema.

**Como importar arquivos?** Levado em conta que, os únicos dados que devem ser inseridos no software são oriundos dos arquivos fornecidos pelo próprio IBGE, é importante descobrir como pode ser feito o processo de importação para a memória temporária do computador.

Desse modo, na linguagem de programação *Python*, a importação de arquivos se dá por meio da função *open*, onde há a possibilidade de inserir três argumentos: o primeiro deles refere-se ao nome do arquivo a ser importado para a memória; o segundo refere-se ao modo como o arquivo será importado; e o terceiro diz respeito a codificação que será levada em conta [python.org 2016].

No sistema desenvolvido, a função *open* foi usada para importar os três arquivos a serem manipulados na memória da seguinte forma: no modo somente leitura, tendo em vista que não há a necessidade de editar os mesmos; e na codificação *UTF-8*, para que a acentuação dos dados coletados venha a funcionar.

**Qual a importância da codificação dos arquivos para o projeto?** A codificação em síntese diz respeito ao modo como são armazenados os dados de um texto no computador, desse modo, todos os elementos que compõem um texto são armazenados na forma de números. Ao levar isso em conta, devido ao fato das codificações possuírem conjuntos numéricos usados de formas divergentes, é natural que o modo como são utilizadas as codificações pode influenciar em como o computador exibe determinada informação.

Assim, ainda que na linguagem *Python* a função *open* tenha como argumento a disponibilidade de informar uma codificação, caso a mesma venha a divergir com a do arquivo a ser importado, certamente essa incoerência irá provocar um erro no programa, fazendo que o mesmo seja fechado de maneira forçada e inesperada.

Tendo em vista que, nos testes realizados a ausência de unanimidade no que se refere ao tipo codificação usada nos arquivos passou a provocar erros, houve a necessidade de mudá-los para apenas uma codificação. Nesse sentido, devido o fato de haverem caracteres acentuados, a codificação adotada em todos os arquivos passou a ser a *UTF-8*.

**Como validar “exemploPesquisa.txt”?** Por meio do cruzamento de informações disponíveis nos arquivos fornecidos pelo IBGE, um dos requisitos desse projeto piloto consiste em exibir ao usuário apenas dados válidos. Assim, houve a necessidade de observar melhor o que eles poderiam possuir em comum, logo notou-se que “regioes.txt” tem em comum a posição das cidades pesquisadas em “exemploPesquisa.txt”, bem como, esse tem em comum a matrícula dos técnicos disponíveis em “tecnicosIBGE.txt”. Vale ressaltar, que antes do fim das sessões ocorreu uma mudança nas informações contidas nesses arquivos citados, porém, nessa versão do projeto o mesmo utiliza as versões anteriores a mudança, implicando em falha caso isso não venha a ser considerado durante eventuais testes.

A princípio, o processo de validação ocorre com a importação dos arquivos para a memória *RAM* do computador, para só então, por meio do comando *split(‘;’)* transformar cada linha do arquivo em uma lista que é inserida em uma outra lista. Assim, criando uma estrutura que simula o que vem a ser uma matriz ou tabela, tendo em vista que no *python* não há em sua apresentação mais básica a estrutura de matriz como em outras linguagens, de todo modo, pode se assumir por convenção que uma matriz no *python* é uma lista de listas [Wazlawick 2017].

Assim que encontrados dados em comum entre determinada linha das matrizes de “tecnicosIBGE.txt” e “exemploPesquisa.txt”, essa linha de “exemploPesquisa.txt” é salva em uma lista, dando origem a uma nova matriz que também é comparada de igual modo com a matriz de “regioes.txt”, concluindo sequencialmente o processo de validação de dados.

**O que é e para que serve uma lista?** O conceito de lista no *Python* pode ser entendido como uma estrutura de dados capaz de conter valores de diversos tipos em diferentes posições simultaneamente [Wazlawick 2017]. Desse modo, listas desempenham o mesmo papel que as variáveis, porém, podendo conter em si dados que podem ser armazenados e obtidos por índice.

No *Python*, listas podem ser criadas vazias ou com algum elemento, ficando a critério do desenvolvedor escolher o método mais adequado para sua aplicação. Desse modo, caso o mesmo queira criar uma lista vazia o processo pode ser feito da seguinte maneira: *lista = []*, ou caso queira contendo algum elemento: *lista = [‘elemento’]*.

No *software* descrito nesse relatório, o conceito de lista é usado tanto para criar matrizes oriundas de listas de listas, quanto para armazenar múltiplos valores em diferentes índices, solucionando de maneira eficiente algumas questões que naturalmente exigiriam o uso de determinada quantidade de variáveis.

**O que é e para que serve uma matriz?** A situação onde há a necessidade de trabalhar com dados organizados na forma de tabela, pode no *Python* ser solucionada através do que vem a ser considerada uma matriz.

Ademais, na linguagem essa estrutura tem por característica a capacidade de apresentar dois tipos de índice: um tratado como coluna e outro como linha. Sendo em outras palavras, dois índices que apresentam-se como verticais e horizontais respectivamente.

Nesse sentido, a vantagem se apresenta como a possibilidade de acessar dados de modo bilateral, seja como coluna (vertical) e como linha (horizontal). Assim pode-se

determinar que, se necessário serão trabalhados de alguma forma, todos os dados contidos no ponto em que as linhas cruzam a coluna de índice 2 de uma matriz.

**O que é e para que serve um dicionário?** Na situação onde as possibilidades contidas na utilização de listas e matrizes se tornam menos interessantes, apresentam-se as formas de uso do que vem a ser um dicionário. Sendo essa, uma estrutura de memória associativa, onde dado valor fica associado a uma chave. Nesse contexto, a mesma é mais eficiente que listas e matrizes por não ser necessário sair em busca de determinada informação de índice em índice.

Desse modo, a linguagem *Python* oferece ao desenvolvedor a possibilidade de trabalhar com essa estrutura de memória associativa chamada dicionário. Ademais, por atuar com um sistema de chaves, ela nos permite usar não apenas números inteiros não negativos para acessar dados, mas também outros caracteres como letras e números, sejam eles quais forem [Wazlawick 2017].

Assim, visando ganho de desempenho e praticidade, o dicionário foi implementado na construção de determinados trechos, onde seria mais conveniente associar dado valor a determinada chave. À exemplo do problema de número sete, onde solicita que seja informado ao usuário a região com maior número de municípios pesquisados. Nesse sentido, caso fosse feita uma busca apenas em lista ou matriz o processo se tornaria menos eficiente em termos de processamento.

**O que é e para que serve a modularização?** A modularização em síntese diz respeito a possibilidade de decompor grandes problemas em outros menores, desse modo facilitando a compreensão do que é implementado na construção do algoritmo [Forbellone 2000]. Assim, a medida em que são sanadas as demandas pouco a pouco, estruturas contendo uma sequência de instruções são criadas.

Na programação, o conceito de problema é resolvido por meio da implementação de algoritmos, esses que ao serem complexos demais para o entendimento, podem ser divididos em outros algoritmos menores e com menos complexidade. Desse modo, todo o processo descrito pode ocorrer através de funções e procedimentos.

Nesse sentido, ao longo da implementação dos algoritmos usados na solução do problema, funções e procedimentos foram criados para atender as mais diversas finalidades. Dentre elas, a principal é possuir a capacidade de reduzir o tamanho do código de modo que, determinados grupos de instruções possam ser reaproveitadas sem que haja a necessidade de repeti-los em outros trechos do programa.

## **2.1. Manual de Uso**

O software desenvolvido como produto para solucionar o problema descrito na introdução desse relatório, foi implementado na linguagem de programação Python na versão 3.7. Ademais, dentre as diversas ferramentas de desenvolvimento disponíveis como IDLE, PyCharm e outras, foi usado o Visual Code Studio, disponibilizado pela Microsoft gratuitamente para os sistemas operacionais Windows, Linux e MAC.

**1ª Condição de Uso.** Visando atender a uma gama mais abrangente de possíveis usuários, o software foi testado e projetado para funcionar em sua plenitude nos sistemas operacionais Windows e Linux. Ademais, devido as mudanças sofridas pelo Python ao longo de suas versões, a solução implementada pode apresentar falhas ao ser lida pelo

interpretador cuja versão seja mais antiga que a 3.7.

**2ª Condição de Uso.** Comandos externos a linguagem foram usados com a finalidade de limpar a tela do programa sempre que necessário, porém, o mesmo apresenta uma falha nesse aspecto quando utilizado no IDLE e Pycharm. Desse modo, é recomendado ao usuário efetuar a interpretação dos códigos desenvolvidos apenas no Prompt de Comandos do Windows e no Terminal do Linux, bem como no Visual Code Studio.

**3ª Condição de Uso.** Levando em conta que ao longo das sessões de resolução do problema proposto mais de uma versão dos arquivos a serem trabalhados foi produzida, então nesse projeto piloto a versão dos arquivos adotada foi a primeira a ser disponibilizada pelos técnicos do IBGE. Assim, embora as mudanças tenham sido sutis no que se refere a ordem em que os dados se apresentam, o modo como o software foi projetado não é capaz de lidar com tais mudanças, sendo expressamente recomendado fazer o uso do programa apenas na primeira versão dos arquivos: “tecnicosIBGE.txt”, “regioes.txt” e “exemploPesquisa.txt”.

**4ª Condição de Uso.** Pensando em ofertar ao usuário uma determinada organização, todos os arquivos a serem importados deve estar contidos na pasta “*File-Base*” localizada no mesmo diretório do programa. Além disso, vale destacar que todos os arquivos devem se encontrar na codificação *UTF-8* para que os mesmos possam ser lidos corretamente pelo programa, logo para evitar eventuais problemas é necessário salvar os arquivos na codificação informada a partir de uma outra janela do editor de textos.

**5ª Condição de Uso.** O processo de execução no Windows se dá através do comando *py* seguido do nome completo do arquivo em seu diretório, em quanto que no Linux o processo ocorre da mesma forma, com o diferencial de que o comando usado é o *python3* ao invés de *py*, como no Windows.

**Considerações Finais Sobre o Manual.** Satisfazendo todas as condições de uso, ao ser aberto o programa deverá surgir na tela do usuário com um menu contendo nove opções, sendo as sete primeiras para finalidades distintas, a oitava para efetuar todas as sete anteriores e por fim a nona opção possibilita que o usuário saia do programa quando quiser. No que se refere a questão de número dois, foi estabelecido em sessão que fica a cargo do desenvolvedor escolher de que modo seu *software* exibe os dados, podendo considerar o ponto 1.02 do questionário ou não. Nesse caso, o a solução descrita no relatório não faz uso dessa possibilidade.

## 2.2. Organização do Código

O código da solução está organizado sistematicamente da seguinte forma:

1. Primeiro passo, assim que o programa é iniciado, o procedimento que corresponde ao menu principal é chamado, nele a tela do usuário é limpa e em seguida é exibida a mensagem “Carregando, aguarde...”.

2. Segundo passo, dentro desse procedimento, duas matrizes são criadas recebendo os dados dos arquivos “regioes.txt” e “exemploPesquisa.txt” já validados.

3. Terceiro passo, se tudo ocorrer como o planejado, o programa entra em seu laço principal e exibe ao usuário um menu de nove opções. De modo que, assim que o mesmo emite algum comando de texto, o programa entra em uma série de condicionais que vão

chamar as instruções responsáveis executar as ações esperadas.

### 3. Conclusões

Apresentada ao longo desse relatório e construída na linguagem de programação python, a solução para o problema proposto possui uma interface caracterizada pelo uso de linhas de comando. Ademais, tendo origem em arquivos de texto na codificação *UTF-8*, todos os dados trabalhados são armazenados, processados e exibidos pelo programa através da memória temporária do computador. Assim, o software construído soluciona o problema proposto utilizando-se de recursos presentes não só na linguagem python, mas também no sistema operacional em que o mesmo estiver sendo interpretado. Desse modo, nos algoritmos que compreendem a solução do problema foram utilizadas variáveis, funções, procedimentos, listas, condicionais, estruturas de repetição, matrizes e dicionários.

A forma como importa, processa e exibe informações ao usuário de modo preciso, leve, detalhado e fluido constitui o que há de melhor na solução implementada. Essa que, por meio de suas funcionalidades pode constituir-se ferramenta essencial para o IBGE no que diz respeito a exibição organizada dos dados coletados ao longo de suas pesquisas em loco. No que tange a questão de número quatro, é possível determinar que não há a forma mais comum de abastecimento, levando em conta que a questão não pede as formas mais comuns e sim a mais comum. Além disso, sobre a questão de número seis, é possível determinar que mais de uma região são as que possuem o maior número de domicílios pesquisados, tendo em vista que a referida questão pede apenas a região com maior número de domicílios pesquisados e não as regiões.

Todavia, caso os arquivos sejam relativamente extensos, o modo como ocorre o processamento de dados pode provocar certa lentidão no software. Nesse sentido, a realização da leitura e processamento de informações pode se dar por meio de um banco de dados, tornando-se uma alternativa não só mais segura, mas como também mais leve. Outro ponto que tange a questão do processamento é que na solução pode ser implementada, a funcionalidade de exportação de dados em arquivo, assim dispensando a necessidade processar os dados novamente caso o usuário queira revê-los.

### Referências

- Forbellone, A. V. L. (2000). *Lógica de Programação: A Construção de Algoritmos e Estrutura de Dados*. Makron Books.
- IBGE (2017). Linha do tempo – síntese da história do ibge. Disponível em: <<https://memoria.ibge.gov.br/images/memoria/linha-do-tempo/LinhaDoTempoSemImagem.pdf>>. Acesso em: 14 Ago 2019.
- python.org (2016). Input and output - reading and writing files. Disponível em: <<https://docs.python.org/3/tutorial/inputoutput.html>>. Acesso em: 14 Ago 2019.
- Wazlawick, R. S. (2017). *Introdução a Algoritmos e Programação com Python*. Elsevier.