

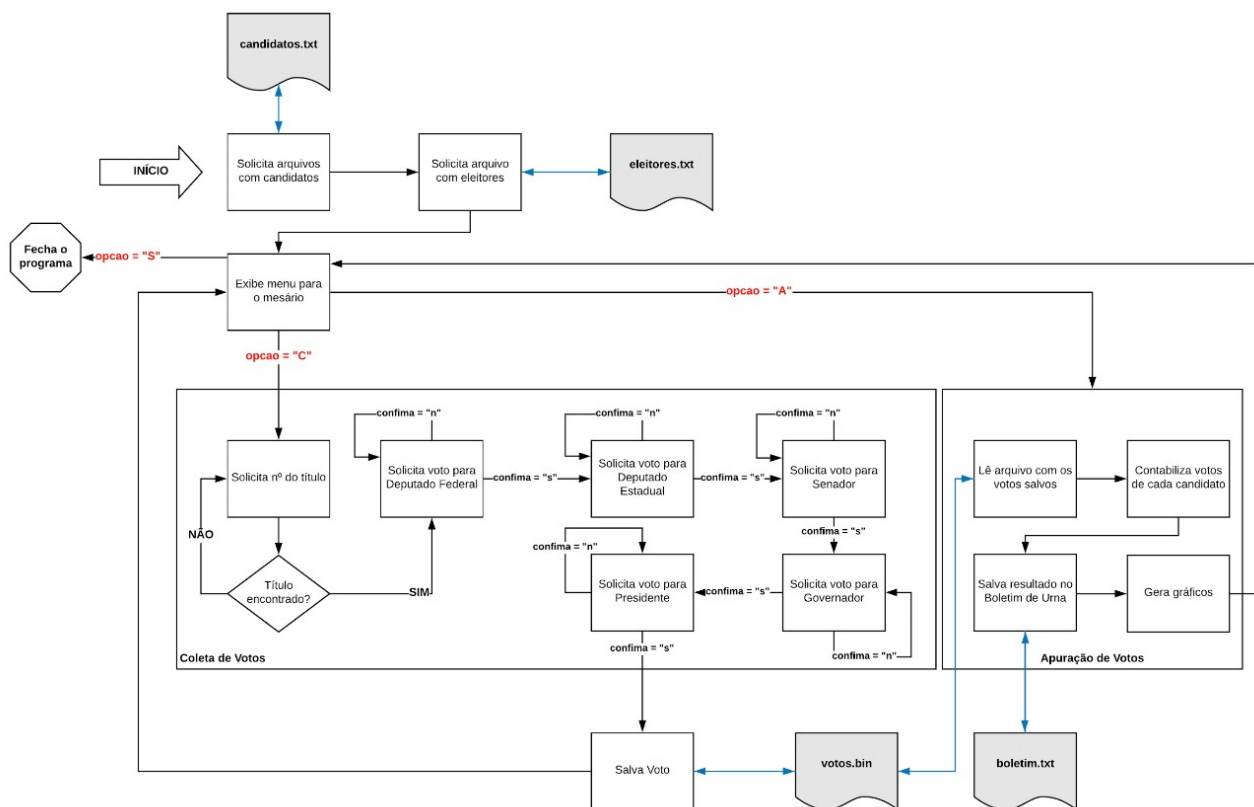
Estruturas de Dados – Trabalho Final

Este trabalho tem como objetivo o desenvolvimento de uma urna eletrônica em Python, e visa reforçar, por meio da prática, os conceitos de listas, dicionários, funções e arquivos vistos em sala de aula.

CrITÉRIOS de Correção:

1. Envio do trabalho – Arquivos fonte (.py) e vídeo de apresentação (link);
2. Atendimento a todos os requisitos propostos;
3. Comentários no código;
4. Organização de responsabilidades em funções separadas (utilização de funções);
5. Organização dos dados em estruturas como listas e/ou matrizes e/ou dicionários, etc.
6. Corretude do código e correto funcionamento.

O fluxograma abaixo resume graficamente o comportamento que é esperado do programa. Nas próximas páginas é explicado com detalhes cada uma dessas funções.



1. Menu: o programa deverá exibir o seguinte menu para o mesário:

- 1 – Ler arquivo de candidatos
- 2 – Ler arquivo de eleitores
- 3 – Iniciar votação
- 4 – Apurar votos
- 5 – Mostrar resultados
- 6 – Fechar programa

2. Leitura de dados dos candidatos via arquivo: o programa deverá ler os dados de cada candidato, salvos previamente em um arquivo de texto. Não deve ser possível iniciar executar as opções 3, 4 ou 5 sem que esses dados sejam lidos antes. Caso o usuário tente, deve ser exibida uma mensagem de erro. Ao executar a opção 1 no menu o programa deverá perguntar para o mesário qual arquivo deverá ser utilizado para buscar os dados. Feito isso, ele deverá ler cada um dos dados e salvá-los em variáveis internas do programa (listas, dicionários, etc).

Entrada:

Informe a localização dos dados dos candidatos: candidatos.txt

● **Cada candidato deve ter salvo os seguintes dados:** nome; número; partido; estado; cargo pretendido (definido de acordo com a tabela abaixo):

CÓDIGO	CARGO	DÍGITOS
F	Deputado Federal	4
E	Deputado Estadual	5
S	Senador	3
G	Governador	2
P	Presidente	2

Exemplo de arquivo de entrada:

```
1 José da Silva, 7321, PTB, MG, F
2 Maria do Carmo, 7321, PCB, SP, F
3 Pedro de Azevedo, 731, PTB, PR, S
4 Antônio de Melo, 151, PCB, SP, S
5 Carlos Nascimento, 18151, PXB, MG, E
6 Guilherme Guimarães, 73215, PTB, SP, E
7 Paulo da Cunha, 15, PXB, SP, P
8 Emília Santos, 15, PCT, MG, G
...
...
```

3. Leitura de dados dos eleitores via arquivo: ao selecionar a opção 2 no menu, o programa deverá ler os dados de cada eleitor, salvo previamente em outro arquivo de texto. Não deve ser possível iniciar executar as opções 3, 4 ou 5 sem que esses dados sejam lidos antes. Caso o usuário tente, deve ser exibida uma mensagem de erro. Da mesma forma, o programa deverá ler cada um dos dados e salvá-los em variáveis internas do programa (listas, dicionários, etc).

Entrada:

Informe a localização dos dados dos eleitores: eleitores.txt

- Cada eleitor deve ter salvo os seguintes dados: nome; RG; Título de Eleitor; Município e Estado.

Exemplo de arquivo de entrada:

1	Pedro de Assis, MG17845369, 015443020272, Formiga, MG
2	Carla Machado, SP156987456, 848242120221, Osasco, SP
3	Félix Nascimento, MG18123654, 670351800264, Pains, MG
4	Luís Silva, RS16457956, 732787030230, Santa Maria, RS
...	
...	

4. Coleta de votos: após o carregamento do arquivo indicado pelo mesário, deve-se iniciar o processo de coleta de votos. O mesário deverá informar em qual estado a urna está localizada:

Entrada:

UF onde está localizada a urna: MG

Antes da coleta de cada voto o mesário deverá informar o nº do título de eleitor do eleitor. Em seguida, o nome e o estado desse eleitor deverão ser mostrados na tela.

Os votos deverão ser coletados na seguinte ordem:

- 1º Deputado Federal
- 2º Deputado Estadual
- 3º Senador
- 4º Governador
- 5º Presidente.

OBS: exceto para o cargo de Presidente, os eleitores só poderão votar no candidato de mesmo estado que o seu. Caso não exista nenhum candidato naquele estado com o número informado, o programa deverá perguntar se deseja anular o voto. Observe que podem existir candidatos com mesmo número para o mesmo cargo, porém em estados diferentes, da mesma forma que o governador pode ter o mesmo número do presidente.

Exemplo de voto:

Informe o Título de Eleitor: 134567899875
Título não encontrado!
Informe o Título de Eleitor: 015443020272
Eleitor: Pedro de Assis
Estado: MG
Informe o voto para Deputado Federal: 7321
Candidato José da Silva | PTB
Confirma (S ou N)? S
Informe o voto para Deputado Estadual: 73215
Candidato não encontrado! Voto Nulo.
Confirma (S ou N)? N
Informe o voto para Deputado Estadual: 18151
Candidato Carlos Nascimento | PXB
Confirma (S ou N)? S
Informe o voto para Senador: B
Voto em Branco
Confirma (S ou N)? S
Informe o voto para Governador: 19
Candidato não encontrado! Voto Nulo.
Confirma (S ou N)? S
Informe o voto para Presidente: 18
Candidato Paulo da Cunha | PXB
Confirma (S ou N)? S
Voto registrado com sucesso!

Registrar novo voto (S ou N)? S

...

...

OBS: Caso a resposta para o input “Registrar novo voto (S ou N)” seja “N”, o programa deverá retornar ao menu principal.

5. Persistência de dados: ao final da coleta de um voto esse deverá ser escrito em um arquivo, para posterior contabilização. Como exemplo, após o voto acima e outros votos, esse arquivo estaria como mostrado abaixo (como ele é binário não seria possível visualizar dessa forma no bloco de notas):

Votos salvos:

```
{“UF”: “MG”, “F”: 7321, “E”: 18151, “S”: “B”, “G”: “N”, “P”: 18}  
{“UF”: “MG”, “F”: 1354, “E”: 13511, “S”: “151”, “G”: “18”, “P”: 19}  
{“UF”: “MG”, “F”: 7321, “E”: “B”, “S”: “132”, “G”: “13”, “P”: 18}  
{“UF”: “MG”, “F”: “B”, “E”: 16645, “S”: “453”, “G”: “15”, “P”: 10}  
{“UF”: “MG”, “F”: 1894, “E”: 12123, “S”: “151”, “G”: “18”, “P”: “B”}
```

...

...

OBS: Para salvar o dicionário onde estarão as informações de cada voto use a biblioteca Pickle do Python, mostrada em aula.

6. Apuração dos votos: após o mesário encerrar a coleta de votos (ele pode encerrar o programa ao final da votação de cada eleitor), a apuração dos votos deve ser iniciada. O programa deverá então ler o arquivo onde estão salvos os votos de cada pessoa (mostrado acima) e contabilizar, para cada candidato de cada estado, a quantidade de votos recebida e a porcentagem de eleitores que isso representa.

7. Divulgação dos resultados: após o processo de apuração dos votos, o programa deverá exibir esses valores em um arquivo de texto, o chamado Boletim de Urna (BU), seguindo o exemplo abaixo:

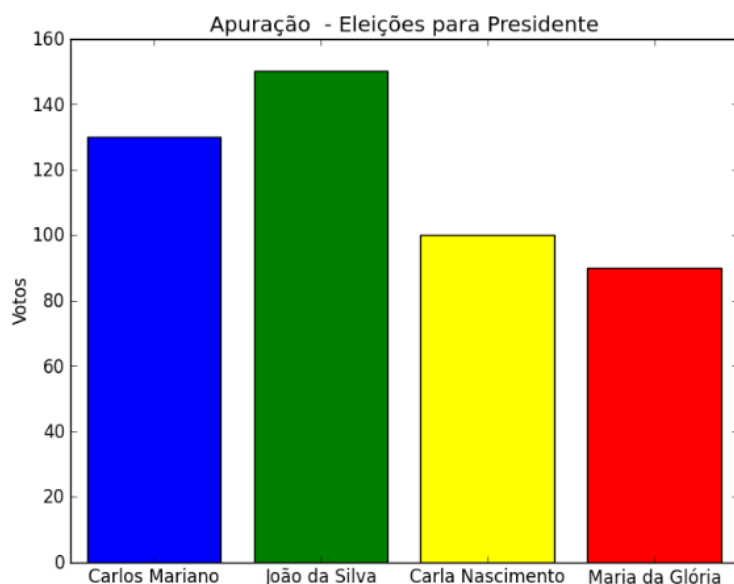
Apuração dos votos:

Eleitores Aptos: 130			
Total de Votos Nominais: 100			
Branco: 10			
Nulos: 15			
Candidato: Carlos Nascimento	Cargo: Deputado Estadual	Estado: MG	Votos: 25 (25%)
Candidato: José da Silva	Cargo: Deputado Federal	Estado: MG	Votos: 30 (30%)
Candidato: Antônio de Melo	Cargo: Senador	Estado: MG	Votos: 20 (20%)
Candidato: Cristiano Costa	Cargo: Deputado Federal	Estado: MG	Votos: 15 (15%)
Candidato: Paulo da Cunha	Cargo: Presidente	Estado: SP	Votos: 35 (35%)
Candidato: Cristina de Melo	Cargo: Presidente	Estado: MS	Votos: 28 (28%)
...			
...			

Após gerar o boletim de urna, o programa deverá, utilizando a biblioteca matplotlib, criar e exibir, para cada cargo da eleição, um gráfico de barras com o resultado.

Crie em seu código a função “`gera_grafico(título, votos)`”, onde o argumento título é uma string que deve ser colocada como título do gráfico a ser gerado, e o argumento votos é um dicionário com a chave sendo o nome do candidato e o seu valor a quantidade de votos que obteve.

Exemplo de gráfico gerado para o cargo de Presidente:



Exemplo do dicionário “votos” utilizado nesse gráfico:

```
votos = {"João da Silva": 150,  
        "Carlos Mariano": 130,  
        "Maria da Glória": 90,  
        "Carla Nascimento": 100}
```

OBS: esse dicionário deverá ser montado com os dados obtidos no processo de apuração dos votos, ou seja, ele vai conter os votos salvos para cada candidato.

---- FIM DA DESCRIÇÃO E REQUISITOS DO TRABALHO ----

MINHAS CONSIDERAÇÕES:

Decidi focar só na votação da região sudeste porque eu gostaria que tivesse pelo menos 2 candidatos de cada cargo em cada estado (para que a exibição dos resultados e dos gráficos ficasse mais interessante e realista).

O código ficaria muito repetitivo caso eu optasse por fazer a votação incluindo todos os 27 estados do Brasil e a lista de candidatos ficaria enorme (pelo menos dois candidatos de cada cargo para cada estado = 270 candidatos).

Então, pra simplificar, inclui somente candidatos e eleitores da região sudeste nesse programa.

Outro detalhe: Tentei desenvolver uma urna que funcionasse (obviamente!) mas que fosse também à prova de usuário. Para isso, tentei imaginar todos os caminhos possíveis que o usuário poderia percorrer e acabar quebrando o programa. Então, dependendo do comportamento do usuário, aparecem mensagens de erro explicando o problema e, assim, evitando que o programa quebre.

Alguns exemplos:

No menu inicial com as 7 opções, e o usuário deve digitar o número da opção.

O programa não prossegue enquanto o usuário não digitar um número válido.

Esse tipo de verificação acontece sempre que o usuário é solicitado a digitar algo.

Seja opções do menu, ao digitar a UF da urna, o título de eleitor, o número dos candidatos, o arquivo de candidatos ou de eleitores, ao perguntar se o usuário quer anular, votar em branco ou confirmar o voto... Todos os inputs possuem validação!

Então sempre que o usuário é solicitado a digitar algo, o programa não vai prosseguir enquanto o usuário não digitar uma entrada válida, e a mensagem de erro vai apontar exatamente qual é o problema.

Outro exemplo interessante é caso o usuário tente pular etapas, por exemplo:

- Tentando exibir os resultados antes de iniciar a votação...
- Tentando gerar os gráficos antes de apurar os votos...
- Tentando iniciar a votação sem carregar o arquivo de candidatos ou de eleitores...

O programa possui mecanismos para evitar que isso aconteça e para que funcione corretamente.

Recursos interessantes:

O mesmo eleitor não pode votar duas vezes. Tente!

Pausa de pelo menos 1 segundo ao exibir mensagens de erro (pra que o usuário consiga ler a mensagem).

Votar em branco pressionando B ou simplesmente pressionando “Enter” sem digitar nada.

Anular o voto digitando um número que não existe ou de um candidato que seja de um estado diferente do estado do eleitor.

O programa é capaz de identificar quando todos os eleitores já votaram e solicita que a votação seja encerrada.

A urna pode receber votos de eleitores de vários estados diferentes e é capaz de fazer a apuração corretamente, separando os resultados por estado.

Ao final da votação, a urna emite um efeito sonoro idêntico ao das urnas reais!