

## Programação - Trabalho Final - Parte 01

O trabalho apresentado a seguir deverá ser desenvolvido em grupos de no máximo 3 (três) pessoas, apresentado e entregue na aula do dia **27/11/2018**.

### Critérios de Correção: (obrigatórios)

- 1) apresentação do trabalho;
- 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) correteude do código e correto funcionamento.

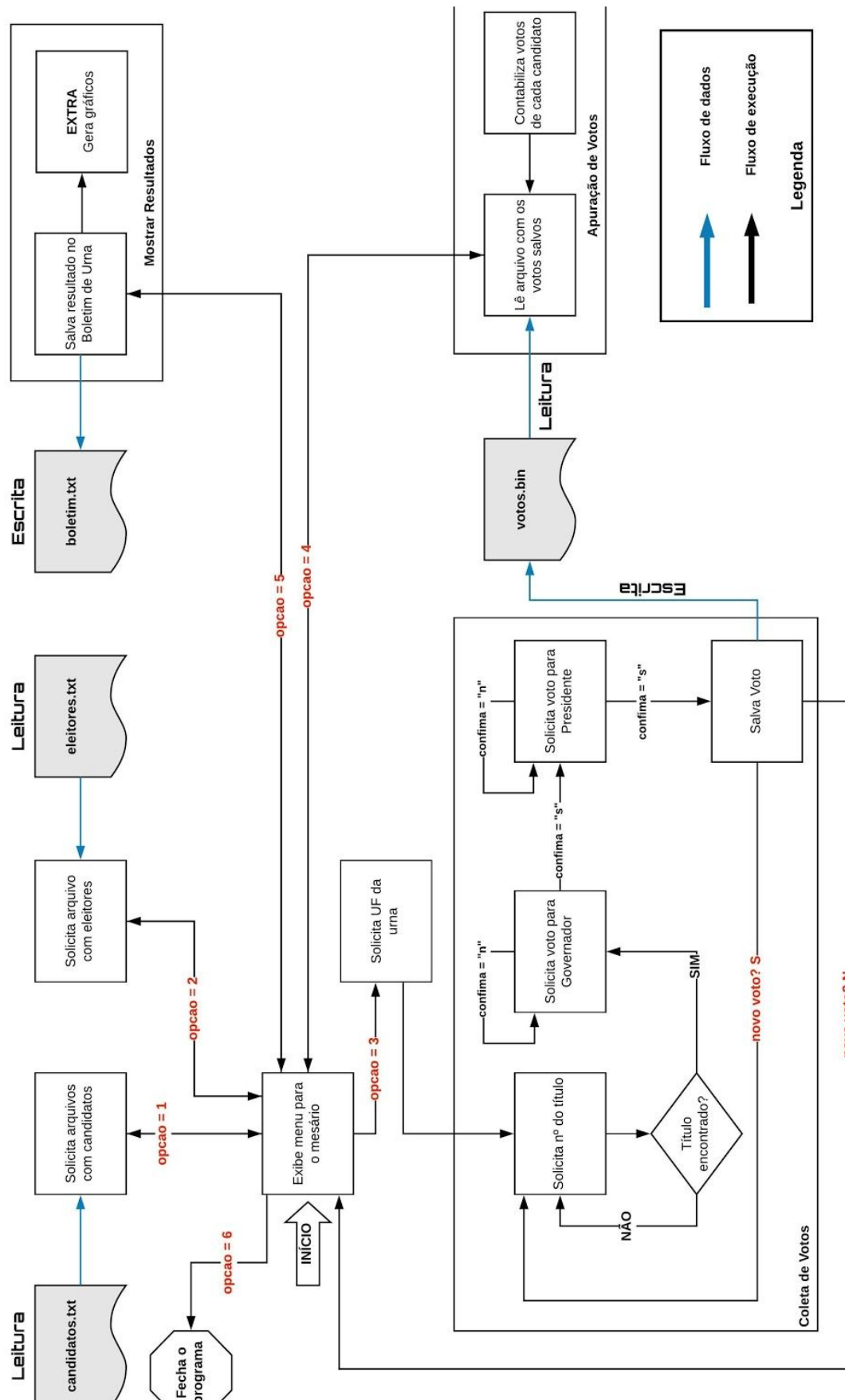
**Trabalhos copiados de outros grupos serão, ambos, desconsiderados na correção, recebendo assim nota 0 (zero).**

### Funcionalidades a serem atendidas:

O presente trabalho tem como objetivo o desenvolvimento de uma urna eletrônica simplificada, este trabalho visa reforçar, por meio da prática, os conceitos de listas, dicionários, funções e arquivos vistos na sala de aula.



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 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 e cargo pretendido (definido de acordo com a tabela abaixo):

| CÓDIGO | CARGO      | DÍGITOS |
|--------|------------|---------|
| "G"    | Governador | 2       |
| "P"    | Presidente | 2       |

**Exemplo de arquivo de entrada:**

```
1 José da Silva, 73, PTB, MG, G
2 Maria do Carmo, 32, PCB, SP, P
3 Pedro de Azevedo, 73, PTB, PR, P
4 Antônio de Melo, 32, PCB, SP, G
5 Carlos Nascimento, 31, P XB, MG, G
5 Guilherme Guimarães, 15, P XB, SP, G
6 Paulo da Cunha, 15, PCT, RJ, G
7 Emília Santos, 81, PCE, RJ, G
8 Cristiano Costa, 81, PCE, SP, P
...
...
```

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 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 dos arquivos indicados pelo mesário, caso ele selecione no menu a opção 3, deve-se iniciar o processo de coleta de votos. O mesário deverá primeiramente informar em qual estado (UF) 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, dessa forma, caso o título seja encontrado nos dados lidos do arquivo de eleitores, seus dados deverão ser mostrados na tela.

**Atenção:** o título tem que estar cadastrado no mesmo estado onde está a urna, caso o número do título exista, porém esteja cadastrado em estado diferente deve-se considerar que não foi encontrado

Os votos deverão ser coletados na seguinte ordem:

- 1º Governador
- 2º Presidente

**OBS 01:** para o cargo de Governador 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: 732787030230
Título não encontrado em MG!
```

```
Informe o Título de Eleitor: 015443020272
Eleitor: Pedro de Assis
RG: MG17845369
Cidade: Formiga
Estado: MG
```

```
Informe o voto para Governador: 32
Candidato não encontrado! Voto Nulo.
Confirma (S ou N)? N
```

```
Informe o voto para Governador: 73
Candidato: José da Silva | PTB
Confirma (S ou N)? S
```

```
Informe o voto para Presidente: 81
Candidato: Cristiano Costa | PCE
Confirma (S ou N)? S
```

```
Voto registrado com sucesso!
```

```
-----
```

```
Registrar novo voto (S ou N)? S
```

```
-----
```

```
...
```

```
...
```

**OBS 02:** 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. **Dica:** para salvar o dicionário onde vão estar as informações de cada voto use a biblioteca Pickle do Python mostrado no slide.

**Votos salvos:**

```
{“UF”: “MG”, “G”: “73”, “P”: 81}  
{“UF”: “MG”, “G”: “31”, “P”: 32}  
{“UF”: “MG”, “G”: “31”, “P”: 81}  
{“UF”: “MG”, “G”: “N”, “P”: 73}  
{“UF”: “MG”, “G”: “31”, “P”: “B”}  
...
```

6. **Apuração dos votos:** após o mesário encerrar a coleta de votos (ao final da votação de cada eleitor ele pode encerrar o programa), a apuração dos votos pode ser iniciada (opção 04 do menu). O programa deverá então ler o arquivo onde estão salvos os votos de cada pessoa (mostrado acima) e contabilizar, para cada candidato a quantidade de votos recebida. Essa contabilização deve ser salva internamente em alguma estrutura, por exemplo um dicionário, como mostra o exemplo abaixo.

**Votos apurados:**

```
{“73”: 30, “31”: 25, “b”: 4, “n”: 5} # governador  
{“81”: 15, “32”: 28, “73”: 35, “b”: 6, “n”: 10} # presidente
```

7. **Divulgação dos resultados:** após o processo de apuração dos votos, o mesário poderá solicitar (opção 05 do menu) a divulgação dos resultados. O programa então deverá exibir esses valores em um arquivo texto, chamado Boletim de Urna (BU), seguindo o exemplo abaixo. O arquivo mostra cada candidato, sua quantidade de votos e sua porcentagem (quantidade de votos / total de votos nominais (retirando brancos, nulos e que não compareceram):

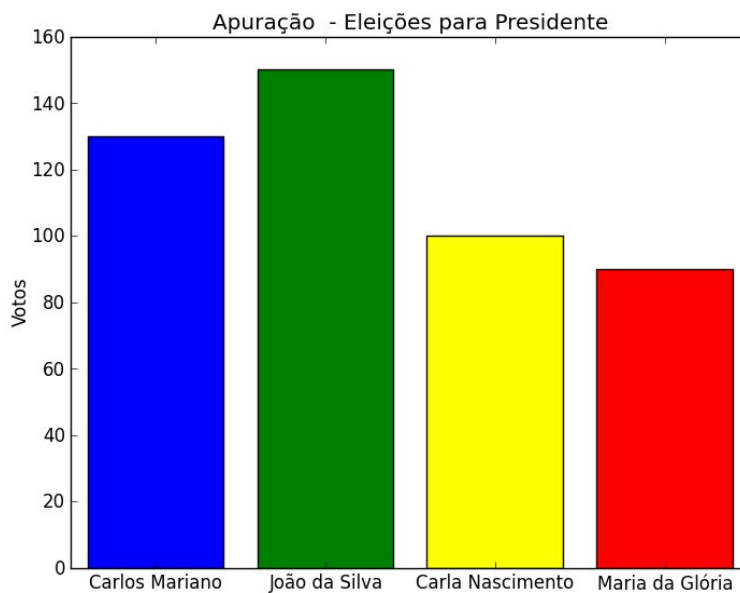
**Divulgação dos votos:**

```
Estado: MG  
Eleitores Aptos: 130  
Total de Votos Nominiais: 100  
Brancos: 10  
Nulos: 15  
  
Candidato: José da Silva | Cargo: Governador | Estado: MG | Votos: 30 (30%)  
Candidato: Carlos Nascimento | Cargo: Governador | Estado: MG | Votos: 25 (25%)  
Candidato: Pedro de Azevedo | Cargo: Presidente | Estado: PR | Votos: 35 (35%)  
Candidato: Maria do Carmo | Cargo: Presidente | Estado: SP | Votos: 28 (28%)  
Candidato: Cristiano Costa | Cargo: Presidente | Estado: SP | Votos: 15 (15%)
```

### Atividade Extra (Desafio) - 5 pontos

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 mostrando de forma gráfica o resultado. Crie em seu código a função “`gera_grafico(título, votos)`”, onde `votos` é um dicionário com a chave sendo o nome do candidato e o seu valor a quantidade de votos que obteve, `título` é uma string que deve ser colocada como título do gráfico a ser gerado. Estude e teste tal biblioteca para implementar tal função.

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.

Alguns materiais de estudo:

- Visualização de Dados com Python (matplotlib): <http://bit.ly/2Ek1YJa>
- Guia básico para plotar gráficos (parte 1): <http://bit.ly/2pTsG0Z>
- Guia básico para plotar gráficos (parte 2): <http://bit.ly/2ChRK9l>
- Gráficos com Python e Matplotlib #1: <http://bit.ly/2NCKmHL>
- Gráficos com Python e Matplotlib #2: <http://bit.ly/2yGogyv>

**Dicas:**

- Antes de iniciar a programação coloque as ideias no papel, por exemplo: como armazenar os dados lidos dos arquivos? Devo usar listas ou dicionários? Como montar um dicionário, o que deve ser chave e o que deve ser valor?
- Reveja as aulas (e códigos) de estruturas sequenciais, condicionais e repetição e leitura e escrita em arquivos texto/binário. Elas podem ser bem úteis para o trabalho.
- Organize o trabalho em funções para cada tipo de funcionalidade. Isso pode ajudar muito na organização do seu código.
- Procure testar os diversos casos que podem ocorrer no programa. Por exemplo, casos de candidato inválido, dentre outros.
- Comente o código a medida que vai escrevendo, assim, ao final do trabalho o código já estará comentado. Com isso também você entende melhor o que está fazendo. Para comentar utilize #.
- Em caso de erros, execute passo a passo seu programa para encontrar onde está o erro. Utilize também a técnica de depuração de código.
- Em caso de dúvidas entre em contato com o professor ou com o monitor da disciplina nos horários de monitoria.