

## Resolução:

```
# Listas dos candidatos
candidatos_prefeito = []
candidatos_governador = []
candidatos_presidente = []

# Listas dos eleitores
eleitores = []

# Dicionários para armazenar os votos
votos_prefeito = {}
votos_governador = {}
votos_presidente = {}

# Função para exibir o menu do projeto
def exibir_menu():
    print("+++++++ MENU - SIMULADOR DO SISTEMA DE VOTAÇÃO ++++++")
    print("1. Cadastrar Candidatos")
    print("2. Cadastrar Eleitores")
    print("3. Votar")
    print("4. Apurar Resultados")
    print("5. Relatório e Estatísticas")
    print("6. Encerrar")

# Função para cadastrar os candidatos
def cadastrar_candidatos():
    cont = 0
    while cont == 0:
        cargo = input("Digite o cargo do candidato (Prefeito / Governador / Presidente): ")
        if cargo.lower() == 'prefeito':
            candidato = {}
            candidato['nome'] = input("Digite o nome do candidato a Prefeito: ")
            candidato['idade'] = input("Digite a idade do candidato: ")
            candidato['numero'] = input("Digite o número do candidato: ")
            candidato['partido'] = input("Digite o partido do candidato: ")
            candidatos_prefeito.append(candidato)
        elif cargo.lower() == 'governador':
            candidato = {}
            candidato['nome'] = input("Digite o nome do candidato a Governador: ")
            candidato['idade'] = input("Digite a idade do candidato: ")
            candidato['numero'] = input("Digite o número do candidato: ")
            candidato['partido'] = input("Digite o partido do candidato: ")
            candidatos_governador.append(candidato)
        elif cargo.lower() == 'presidente':
```

```

        candidato = {}
        candidato['nome'] = input("Digite o nome do candidato a
Presidente: ")
        candidato['idade'] = input("Digite a idade do
candidato: ")
        candidato['numero'] = input("Digite o número do
candidato: ")
        candidato['partido'] = input("Digite o partido do
candidato: ")
        candidatos_presidente.append(candidato)
    else:
        print("Cargo inválido!")

    opcao = input("Deseja cadastrar outro candidato? (S/N): ")
    if opcao.lower() == 'n':
        cont = cont + 1

# Função para cadastrar os eleitores
def cadastrar_eleitores():
    cont2 = 0
    while cont2 == 0:
        eleitor = {}
        eleitor['nome'] = input("Digite o nome do eleitor: ")
        eleitor['cpf'] = input("Digite o CPF do eleitor: ")
        eleitores.append(eleitor)
        opcao = input("Deseja cadastrar outro eleitor? (S/N): ")
        if opcao.lower() == 'n':
            cont2 = cont2 + 1

# Função para votar
def votar():
    cpf_eleitor = input("Digite o CPF do eleitor: ")
    eleitor_encontrado = False

    for eleitor in eleitores:
        if cpf_eleitor == eleitor['cpf']:
            eleitor_encontrado = True
            print("Eleitor:", eleitor['nome'])

            print("----- Votação para Prefeito -----")
            for candidato in candidatos_prefeito:
                print("Número:", candidato['numero'])
                print("Nome:", candidato['nome'])
                print("Partido:", candidato['partido'])
                print('-----')

            voto_prefeito = int(input("Digite o número do candidato
ou -1 para voto em branco ou -2 para voto nulo: "))
            if voto_prefeito == -1:
                print("Voto em branco registrado.")
                voto_prefeito = 'branco'
            elif voto_prefeito == -2:
                print("Voto nulo registrado.")
                voto_prefeito = 'nulo'
            else:
                voto_prefeito = str(voto_prefeito)

            confirmacao = input("Deseja confirmar o voto? (S/N): ")

```

```

if confirmacao.lower() == 's':
    if voto_prefeito in votos_prefeito:
        votos_prefeito[voto_prefeito] += 1
    else:
        votos_prefeito[voto_prefeito] = 1
    print("Voto registrado.\n")

print("----- Votação para Governador -----")
for candidato in candidatos_governador:
    print("Número:", candidato['numero'])
    print("Nome:", candidato['nome'])
    print("Partido:", candidato['partido'])
    print('-=-=-=-=-=-')

voto_governador = int(input("Digite o número do
candidato ou -1 para voto em branco ou -2 para voto nulo: "))
if voto_governador == -1:
    print("Voto em branco registrado.")
    voto_governador = 'branco'
elif voto_governador == -2:
    print("Voto nulo registrado.")
    voto_governador = 'nulo'
else:
    voto_governador = str(voto_governador)

confirmacao = input("Deseja confirmar o voto? (S/N): ")
if confirmacao.lower() == 's':
    if voto_governador in votos_governador:
        votos_governador[voto_governador] += 1
    else:
        votos_governador[voto_governador] = 1
    print("Voto registrado.\n")

print("----- Votação para Presidente -----")
for candidato in candidatos_presidente:
    print("Número:", candidato['numero'])
    print("Nome:", candidato['nome'])
    print("Partido:", candidato['partido'])
    print('-=-=-=-=-=-')

voto_presidente = int(input("Digite o número do
candidato ou -1 para voto em branco ou -2 para voto nulo: "))
if voto_presidente == -1:
    print("Voto em branco registrado.")
    voto_presidente = 'branco'
elif voto_presidente == -2:
    print("Voto nulo registrado.")
    voto_presidente = 'nulo'
else:
    voto_presidente = str(voto_presidente)

confirmacao = input("Deseja confirmar o voto? (S/N): ")
if confirmacao.lower() == 's':
    if voto_presidente in votos_presidente:
        votos_presidente[voto_presidente] += 1
    else:
        votos_presidente[voto_presidente] = 1
    print("Voto registrado.\n")

```

```

        break

    if not eleitor_encontrado:
        print("CPF não encontrado!")

# Função para apurar os resultados
def apurar_resultados():
    print("----- RANKING DO RESULTADO PARA PRESIDENTE")
    print("-----")
    print("Nome\t\tPartido\t\tTotal de Votos\t% votos Válidos")
    ranking_presidente = sorted(votos_presidente.items(),
reverse=True)
    for i, (candidato, votos) in enumerate(ranking_presidente):
        if candidato == 'branco':
            nome = 'Votos em branco'
            partido = '-'
        elif candidato == 'nulo':
            nome = 'Votos nulos'
            partido = '-'
        else:
            for c in candidatos_presidente:
                if c['numero'] == candidato:
                    nome = c['nome']
                    partido = c['partido']
                    break
            percentual_votos_validos = (votos /
sum(votos_presidente.values())) * 100
            print(f"{i+1}.\t{nome}\t\t{partido}\t\t{votos}\t\t\t{percentual_votos_validos:.2f}%")

            total_votos_presidente = sum(votos_presidente.values())
            total_votos_validos_presidente = total_votos_presidente -
votos_presidente.get('branco', 0) - votos_presidente.get('nulo', 0)
            percentual_votos_validos_presidente =
(total_votos_validos_presidente / total_votos_presidente) * 100
            percentual_branco_presidente = (votos_presidente.get('branco',
0) / total_votos_presidente) * 100
            percentual_nulo_presidente = (votos_presidente.get('nulo',
0) / total_votos_presidente) * 100

            print("\nTotal de votos =", total_votos_presidente)
            print("Total de votos válidos e % =",
total_votos_validos_presidente,
f"({percentual_votos_validos_presidente:.2f}%)")
            print("Total de brancos e % =", votos_presidente.get('branco',
0), f"({percentual_branco_presidente:.2f}%)")
            print("Total de nulos e % =", votos_presidente.get('nulo', 0),
f"({percentual_nulo_presidente:.2f}%)")

        print("\n----- RANKING DO RESULTADO PARA GOVERNADOR")
        print("-----")
        print("Nome\t\tPartido\t\tTotal de Votos\t% votos Válidos")
        ranking_governador = sorted(votos_governador.items(),
reverse=True)
        for i, (candidato, votos) in enumerate(ranking_governador):
            if candidato == 'branco':
                nome = 'Votos em branco'
                partido = '-'

```

```

        elif candidato == 'nulo':
            nome = 'Votos nulos'
            partido = '-'
        else:
            for c in candidatos_governador:
                if c['numero'] == candidato:
                    nome = c['nome']
                    partido = c['partido']
                    break
            percentual_votos_validos = (votos /
sum(votos_governador.values())) * 100
            print(f"{i+1}. \t{nome} \t \t{partido} \t \t{votos} \t \t{percentual_votos_validos:.2f}%")

            total_votos_governador = sum(votos_governador.values())
            total_votos_validos_governador = total_votos_governador -
votos_governador.get('branco', 0) - votos_governador.get('nulo', 0)
            percentual_votos_validos_governador =
(total_votos_validos_governador / total_votos_governador) * 100
            percentual_branco_governador = (votos_governador.get('branco',
0) / total_votos_governador) * 100
            percentual_nulos_governador = (votos_governador.get('nulo',
0) / total_votos_governador) * 100

            print("\nTotal de votos =", total_votos_governador)
            print("Total de votos válidos e % =",
total_votos_validos_governador,
f"({percentual_votos_validos_governador:.2f}%)")
            print("Total de brancos e % =", votos_governador.get('branco',
0), f"({percentual_branco_governador:.2f}%)")
            print("Total de nulos e % =", votos_governador.get('nulo', 0),
f"({percentual_nulos_governador:.2f}%)")

            print("\n----- RANKING DO RESULTADO PARA PREFEITO
-----")
            print("Nome\t\tPartido\t\tTotal de Votos\t% votos Válidos")
            ranking_prefeito = sorted(votos_prefeito.items(), reverse=True)
            for i, (candidato, votos) in enumerate(ranking_prefeito):
                if candidato == 'branco':
                    nome = 'Votos em branco'
                    partido = '-'
                elif candidato == 'nulo':
                    nome = 'Votos nulos'
                    partido = '-'
                else:
                    for c in candidatos_prefeito:
                        if c['numero'] == candidato:
                            nome = c['nome']
                            partido = c['partido']
                            break
                    percentual_votos_validos = (votos /
sum(votos_prefeito.values())) * 100
                    print(f"{i+1}. \t{nome} \t \t{partido} \t \t{votos} \t \t{percentual_votos_validos:.2f}%")

                    total_votos_prefeito = sum(votos_prefeito.values())
                    total_votos_validos_prefeito = total_votos_prefeito -
votos_prefeito.get('branco', 0) - votos_prefeito.get('nulo', 0)

```

```

    percentual_votos_validos_prefeito =
(total_votos_validos_prefeito / total_votos_prefeito) * 100
    percentual_branco_prefeito = (votos_prefeito.get('branco',
0) / total_votos_prefeito) * 100
    percentual_nulos_prefeito = (votos_prefeito.get('nulo', 0) /
total_votos_prefeito) * 100

    print("\nTotal de votos =", total_votos_prefeito)
    print("Total de votos válidos e % =",
total_votos_validos_prefeito,
f"({percentual_votos_validos_prefeito:.2f}%)")
    print("Total de brancos e % =", votos_prefeito.get('branco',
0), f"({percentual_branco_prefeito:.2f}%)")
    print("Total de nulos e % =", votos_prefeito.get('nulo', 0),
f"({percentual_nulos_prefeito:.2f}%)")

# Função para fazer o relatório e estatísticas
def gerar_relatorio():
    total_votos_prefeito = sum(votos_prefeito.values())
    total_votos_governador = sum(votos_governador.values())+
    total_votos_presidente = sum(votos_presidente.values())

    print("----- Relatório e Estatísticas -----")
    print("Total de votos para Prefeito:", total_votos_prefeito)
    print("Total de votos para Governador:",
total_votos_governador)
    print("Total de votos para Presidente:",
total_votos_presidente)

    percentual_branco_prefeito = votos_prefeito.get('branco', 0) /
total_votos_prefeito * 100 if total_votos_prefeito > 0 else 0
    percentual_branco_governador = votos_governador.get('branco',
0) / total_votos_governador * 100 if total_votos_governador > 0
else 0
    percentual_branco_presidente = votos_presidente.get('branco',
0) / total_votos_presidente * 100 if total_votos_presidente > 0
else 0

    percentual_nulo_prefeito = votos_prefeito.get('nulo', 0) /
total_votos_prefeito * 100 if total_votos_prefeito > 0 else 0
    percentual_nulo_governador = votos_governador.get('nulo', 0) /
total_votos_governador * 100 if total_votos_governador > 0 else 0
    percentual_nulo_presidente = votos_presidente.get('nulo', 0) /
total_votos_presidente * 100 if total_votos_presidente > 0 else 0

    print(f"Percentual de votos em branco para Prefeito:
{percentual_branco_prefeito:.2f}")
    print(f"Percentual de votos em branco para Governador:
{percentual_branco_governador:.2f}")
    print(f"Percentual de votos em branco para Presidente:
{percentual_branco_presidente:.2f}")

    print(f"Percentual de votos nulos para Prefeito:
{percentual_nulo_prefeito:.2f}")
    print(f"Percentual de votos nulos para Governador:
{percentual_nulo_governador:.2f}")
    print(f"Percentual de votos nulos para Presidente:
{percentual_nulo_presidente:.2f}")

```

```

# Função principal do código
def main():
    cont_saida = 0
    while cont_saida == 0:
        exibir_menu()
        opcao = int(input("Digite a opção que deseja: "))

        if opcao == 1:
            cadastrar_candidatos()
        elif opcao == 2:
            cadastrar_eleitores()
        elif opcao == 3:
            votar()
        elif opcao == 4:
            apurar_resultados()
        elif opcao == 5:
            gerar_relatorio()
        elif opcao == 6:
            print()
            print("Ok, votação encerrada!")
            cont_saida = cont_saida + 1
        else:
            print("Opção não encontrada!")

# A PARTIR DO MOMENTO QUE O CÓDIGO COMEÇAR A RODAR, O PROGRAMA VAI
# CHAMAR A FUNÇÃO MAIN
if __name__ == '__main__':
    main()

```



**UNIVERSIDADE PRESBITERIANA MACKENZIE**

Faculdade de Computação e Informática – FCI

**Ciência da Computação**  
Algoritmos e Programação I  
Laboratório



## **PROJETO PRÁTICO 2º BIMESTRE**

### *SIMULADOR DE UM SISTEMA DE VOTAÇÃO*

**OBJETIVO:** *Este projeto tem como objetivo aplicar o conteúdo trabalhado durante o semestre na disciplina de Algoritmos e Programação I e consolidar o aprendizado dos conceitos nela abordados. Para tal, simularemos um sistema de votação simplificado, que compreende o cadastro dos candidatos e eleitores; processo da apuração, além da emissão de relatórios e dados estatísticos.*

*Ao ser iniciado, o programa da Urna Eletrônica apresenta um menu principal interativo, através do qual o usuário pode selecionar as diferentes operações do sistema, A Figura 1 ilustra a tela principal com o menu de opções disponíveis.*

*O término da execução do programa ocorre somente quando o usuário digita a opção 7 e confirma que deseja encerrar, no final do expediente de votação por exemplo. Para todas as demais escolhas do menu, após realizar tarefa apontada, o programa retorna para a tela principal reexibindo o menu e ficando pronto para uma nova interação do usuário.*

*Cada opção escolhida dispara uma sequência de operações responsável por realizar a tarefa especificada. Assim, de acordo com a escolha, uma função específica deve ser invocada para desempenhar a ação correspondente. O código deverá ser estruturado e organizado em funções separadas, de acordo com o menu.*

**+++++++ MENU - SIMULADOR DO SISTEMA DE VOTAÇÃO +++++++**

- 1. Cadastrar Candidatos**
- 2. Cadastrar Eleitores**
- 3. Votar**
- 4. Apurar Resultados**
- 5. Relatório e Estatísticas**
- 6. Gravar Apuração**
- 7. Encerrar**

**Opção escolhida:**



*Figura 1 Tela principal com menu de opções*

*A seguir são apresentadas descrição básicas da(s) tarefa(s) que cada item do menu deve realizar:*

### **1. Cadastrar Candidatos**

- *Ler o Nome, Idade, Número, Partido e Cargo que disputa e adicionar em uma lista de candidatos. Devemos ter uma lista dos candidatos a presidente, outra dos candidatos a governador e outra para prefeito.*
- *Após cada inserção, o programa deve perguntar se deseja inserir outro candidato ou não. O cadastro deve ser interrompido quando o usuário digitar NÃO nesta opção.*

### **2. Cadastrar Eleitores**

- *Ler o Nome e CPF e adicionar em uma lista de eleitores.*
- *Após cada inserção, o programa deve perguntar se deseja inserir outro eleitor ou não. O cadastro deve ser interrompido quando o usuário responder digitando NÃO.*

### **3. Votar**

- *Os votos são coletados em 3 etapas: Prefeito -> Governador -> Presidente; ou seja, primeiramente o voto para Prefeito, em seguida para Governador e por fim para Presidente.*
- *A informação de cada voto é dada a partir da entrada, considerando o seguinte esquema:*
  - *Voto para um candidato em particular: número do candidato*
  - *Voto branco: digitar -1*
  - *Voto nulo: digitar -2*
- *Em cada uma das etapas (Prefeito -> Governador -> Presidente), após a entrada do número do candidato, a urna eletrônica deve mostrar o nome do candidato, seu partido e pedir uma confirmação do voto. No caso de voto nulo ou branco uma mensagem adequada de confirmação também é apresentada.*
- *Cada voto é então registrado nas listas de candidatos para que seja feita posteriormente a apuração do total de votos.*

### **4. Apurar Resultados**

- *Mostra quem são os candidatos vencedores para cada cargo, e*
- *Apresenta um ranking ordenado do resultado da eleição (do mais votado para o menos), separado por cargo, além de alguns dados estatísticos conforme o exemplo dado na Figura 2. Atenção: caso ocorra um empate entre 2 ou mais candidatos, você deve escolher a idade como um critério de desempate e mostrar, portanto, o de maior idade como 1º do ranking (vencedor).*

- Dica: Pense em uma formatação estilo tabela para a apresentação do ranking, como no exemplo a seguir:

RANKING DO RESULTADO PARA PRESIDENTE			
Nome	Partido	Total de Votos	% votos Válidos
1.			
2.			
3.			
...			
Total de votos =			
Total de votos válidos e % =			
Total de brancos e % =			
Total de nulos e % =			

RANKING DO RESULTADO PARA GOVERNADOR			
Nome	Partido	Total de Votos	% votos Válidos
1.			
2.			
3.			
...			
Total de votos =			
Total de votos válidos e % =			
Total de brancos e % =			
Total de nulos e % =			

RANKING DO RESULTADO PARA PREFEITO			
Nome	Partido	Total de Votos	% votos Válidos
1.			
2.			
3.			
...			
Total de votos =			
Total de votos válidos e % =			
Total de brancos e % =			
Total de nulos e % =			

Figura 2 - Tabelas com resultados da eleição por categoria

## 5. Relatório e Estatísticas

Nesta opção do menu, seu programa deve:

- *Exibir uma lista dos eleitores que votaram, ordenados por nome.*
- *Verificar se a quantidade de eleitores bate com o total de votos que foram registrados na eleição (auditoria), imprimindo uma mensagem do tipo: Votação Auditada! ou Problema na Auditoria!*
- *Mostrar qual partido elegeu mais políticos e a quantidade*
- *Mostra qual partido elegeu menos políticos e a quantidade*

## 6. Gravar Apuração

*Esta opção do menu não é obrigatória no projeto, mas será valorizada se implementada. Ao escolher esta opção do menu o programa irá gravar, em um arquivo texto chamado apuracao.txt, o resultado da apuração das eleições. A ideia é escrever no arquivo as mesmas tabelas impressas na opção 4 do menu, Apurar Resultados.*

Abaixo, segue um pequeno exemplo de como criar o arquivo e gravar as informações nele:

```
arquivo = open('apuracao.txt','w')
arquivo.write("\n*** NESTE ARQUIVO VOCÊS IRÃO GRAVAR AS TABELAS COM OS RESULTADOS DA APURAÇÃO ***\n")
arquivo.write("\n*** USE O MÉTODO WRITE COMO SE FOSSE O PRINT ***\n")
arquivo.write("\nReferência para estudo/consulta: https://www.w3schools.com/python/python_file_handling.asp")
arquivo.close()
```

Consultar: [https://www.w3schools.com/python/python\\_file\\_handling.asp](https://www.w3schools.com/python/python_file_handling.asp)

## INFORMAÇÕES ADICIONAIS E ORIENTAÇÕES:

---

*O projeto será **AVALIADO** de acordo com os seguintes critérios:*

### **Código:**

- *Funcionamento correto do menu e das funções implementadas;*
- *O quão fiel é o programa quanto à descrição da atividade;*
  - *Indentação, comentários e legibilidade do código;*
  - *Clareza na nomenclatura de variáveis e funções;*

### **Relatório:**

- *Organização, completude, clareza e ausência de erros de linguagem no relatório;*

### **Apresentação:**

- *Qualidade da apresentação realizada: clareza, conhecimento e cumprimento do tempo estabelecido.*

**Atenção:** só é permitido usar recursos e conteúdos abordados na disciplina. Caso tenha dúvida sobre a utilização de algo, converse com o professor antes!

### **Observações:**

- *O trabalho pode ser feito em grupo (**máximo 4 pessoas e no mínimo 3**).*
- *Um único aluno do grupo deverá publicar o trabalho no Moodle.*
- *O trabalho será apresentado em sala demonstrando o que foi implementado. Na apresentação o grupo deve comentar se conseguiu atender a todas as solicitações, se não, o que faltou. Escolher apenas algumas coisas que achou legal na implementação e que deseja mostrar!*
  - *Tempo de apresentação: **5 a 8 min** (máximo)*

### **O que deverá ser entregue no Moodle?**

- *O código fonte desenvolvido e comentado (inserir comentários nos blocos principais de cada função, documentando o que foi feito)!*
- *Um relatório no formato PDF contendo:*
  - *Dados dos integrantes do grupo (nome e TIA).*
  - *Decisões relativas à implementação.*
  - *Printscreen com os testes de execução de todas as opções do Menu mostrando cada funcionalidade implementada. Atenção: incluir os testes no relatório na ordem do menu.*
  - *Conclusões explicando se implementou tudo, se faltou algo e suas considerações gerais sobre o projeto.*
  - *Nas conclusões, inserir a tabela abaixo indicando o que foi implementado e o que faltou (coloque um **X** para indicar), além do nome do aluno que implementou a funcionalidade. Todos devem participar da implementação. Na apresentação o professor poderá perguntar detalhes a cada integrante.*

Funcionalidade	Implementaram?			Aluno Responsável
	Sim	Não	Parcial	
Cadastrar Candidatos				
Cadastrar Eleitores				
Votar				
Apurar Resultados				
Relatório e Estatísticas				
Gravar Apuração				