UNIVERSIDADE FEDERAL DO ESPIRITO SANTO

Segundo Trabalho de Programação III

Relatório

Trabalho referente à disciplina  
de Programação III do curso de  
Ciência de Computação do Departamento de Informática da  
Universidade Federal do Espírito  
Santo.

Lucas Moraes Soares de SouzaLucas Mendonça Emery Cade

VitóriaNovembro de 2018

**Sumário**

**1 Introdução 1  
2 Desenvolvimento 2**2.1 Candidato . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2  
2.1.1 Atributos . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2  
2.1.2 Função setColigacao . . . . . . . . . . . . . . . . . . . . . . 2  
2.1.3 Função setColocacao . . . . . . . . . . . . . . . . . . . . . . 2  
2.1.4 Função toString . . . . . . . . . . . . . . . . . . . . . . . . . 2  
2.2 Diagrama . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3  
2.3 Partido . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4  
2.3.1 Atributos . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4  
2.3.2 Função toString . . . . . . . . . . . . . . . . . . . . . . . . . 4  
2.3.3 Função imprimeNomePartido . . . . . . . . . . . . . . . . . 4  
2.4 Coligação . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5  
2.4.1 Atributos . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5  
2.4.2 Função coligacao . . . . . . . . . . . . . . . . . . . . . . . . 5  
2.4.3 Função toString . . . . . . . . . . . . . . . . . . . . . . . . . 5  
2.4.4 Função imprimeColigacao . . . . . . . . . . . . . . . . . . . 5  
2.4.5 Função comparaColigacao . . . . . . . . . . . . . . . . . . . 5  
2.5 Lista candidatos . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6  
2.5.1 Atributos . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6  
2.5.2 Função preencheListas . . . . . . . . . . . . . . . . . . . . . 6  
2.6 Lista partidos . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7  
2.6.1 Atributos . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7  
2.7 Lista coligacao . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8  
2.7.1 Atributos . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8  
2.8 Lista coligacao . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9  
2.8.1 Atributos . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9  
2.9 Dados . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10  
2.9.1 Atributos . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10  
2.10 IO . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11  
2.10.1 Atributos . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11  
2.10.2 Função le arquivo . . . . . . . . . . . . . . . . . . . . . . . . 11  
2.10.3 Função imprimeSaida . . . . . . . . . . . . . . . . . . . . . 11  
2.11 Testador . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12  
2.11.1 Atributos . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12  
**3 Conclusão 1**

**1 Introdução**

O segundo trabalho de implementação tinha como objetivo exercitar os conceitos básicos de programação orientada a objeto aprendidos no primeiro trabalho, aprender os conceitos básicos da linguagem C++ e aprender como entender críticas e melhorias possíveis para o sistema implementado da primeira vez, para a segunda implementação.

Como no trabalho passado, a partir de um arquivo “.csv” referente a uma eleiçao municipal, o programa deveria:

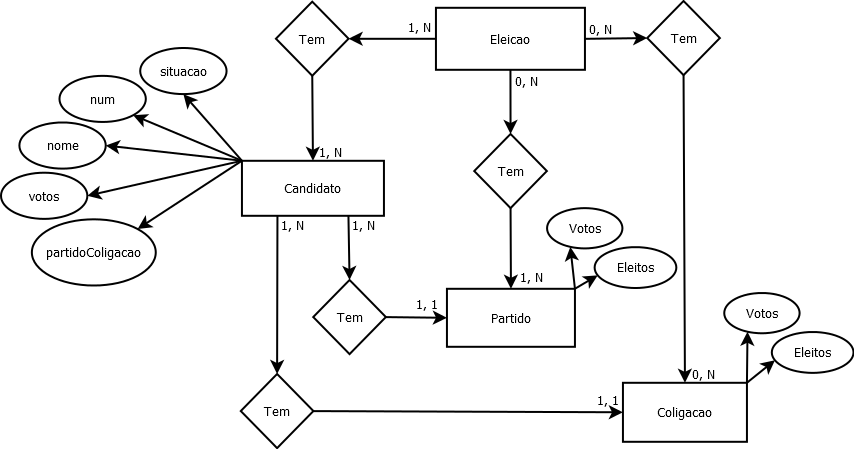
1. Armazenar as informações lidas no arquivo.
2. Gerar as seguintes informações, ordenadas, na saída padrão:
   1. Número de vagas.
   2. Candidatos eleitos (sempre indicando partido, número de votos e coligação).
   3. Candidatos mais votados, dentro do número de vagas.
   4. Candidatos não eleitos, mas que seriam eleitos caso fosse usado o sistema majoritário.
   5. Candidatos eleitos no sistema vigente, e que não seriam eleitos caso fosse usado o sistema majoritário.
   6. As coligações e seus respectivos totais de votos.
   7. Os partidos e seus respectivos totais de votos.
   8. O total de votos nominais da eleição.
3. Liberar a memória alocada para a execução do código.

As listas estão ordenadas pelas especificações do item em questão ou em ordem decrescente de votos.

Foram usadas as classes Eleição[3.1], Candidato[3.2], Partido[3.3], Coligação[3.4], Entrada[3.5] e Saída[3.6] (os últimos dois sendo classes organizacionais que só servem para entrada e saída de dados). As classes são divididas em um arquivo cabeçalho “[Classe].h” e um arquivo de implementação “[Classe].cpp”, sem acentos e símbolos especiais nos nomes. As classes serão explicadas em suas respectivas seções do relatório.

**2 Diagrama**

As classes não-organizacionais e seus atributos são organizadas da seguinte forma:



**3 Classes**

**3.1 Eleição**

A classe eleição é a classe geral que engloba todas as estruturas e listas do trabalho. Seus atributos são:

* candidatos;

Uma lista de ponteiros para instâncias da classe Candidato.

* partidos;

Uma lista de ponteiros para instâncias da classe Partido.

* coligacoes;

Uma lista de ponteiros para instâncias da classe Coligação.

* vagas;

Um inteiro que armazena o número de vagas da eleição

Todos os atributos apresentam um método “get” próprio.

O único método “set” é o “setvagas()”, que percorre a lista de candidatos e calcula o número de candidatos eleitos, que correspone ao número de vagas.

O método “partidoColigacaoVagas()” executa o “setvagas()” e percorre a lista de candidatos executando, para cada candidato, o método seguinte.

O método “adicionaPartidoColigacao(Candidato\*)” divide a string referente ao partido e a coligação do candidato (ou identifica que o candidato não tem uma coligação) e chama as duas funções seguintes.

Os métodos “adicionaPartido(Candidato\*, string)” e “adicionaColigacao(Candidato\*, string)” fazem o mesmo processo para adicionar um partido e uma coligação, respectivamente, no