

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA

Programação III

Primeiro Trabalho 2016/2 – Vereadores

Novembro 2016

Lucas Sartori Moraes
Matheus Hemerly Risso

Sumário

1	Introdução	3
2	Objetivos	4
3	Metodologia de implementação	5
3.1	Entrada e saída de dados	5
3.2	Classes	5
3.2.1	Leitor	7
3.2.2	Candidato	7
3.2.3	Partido	8
3.2.4	Coligacao	9
3.2.5	Eleicao	9
3.2.6	Main	9
4	Testes e Known Bugs	10
4.1	Vitória	10
4.2	Vila Velha	11
4.3	Rio de Janeiro	12
5	Resultados e Avaliação	13
6	Referências Bibliográficas	13

1 Introdução

O primeiro trabalho da disciplina de Programação III consiste em implementar um sistema na linguagem Java que leia arquivos .csv distribuídos pelo sistema eleitoral brasileiro e gere uma saída legível com certos aspectos que mostrem a diferença entre o sistema atual (de quociente eleitoral) e um sistema majoritário de votos.

O sistema implementado possui cinco classes que se comunicam entre si para produzir uma série de análises sobre a corrida eleitoral. Estas classes são divididas em um leitor, responsável por interpretar o arquivo csv; classes referentes a candidato, partido e coligação; e uma classe global responsável por organizar e imprimir as análises da eleição.

Os resultados dos cálculos realizados são interessantes tanto para analisar as intenções de voto de cada cidade, quanto para entender como o sistema proporcional de votos funciona na prática. Este fato se deve à contabilização dos votos de partidos e coligações e a comparação entre a lista de candidatos eleitos e mais votados.

2 Objetivos

O objetivo deste trabalho é desenvolver uma aplicação na linguagem Java que faz uma análise dos dados das eleições municipais para vereador, comparando a eleição proporcional à majoritária.

Um outro objetivo que não deve ser desconsiderado é o de desenvolver os conhecimentos em Java e em linguagens orientadas a objetos. Para isso, optou-se por escolher uma aplicação real que pudesse ser útil para a comunidade brasileira.

Para isso, o sistema se baseia nos arquivos disponibilizados pelo governo brasileiro em formato csv e poderá ser usado futuramente para outras análises eleitorais.

3 Metodologia de implementação

3.1 Entrada e saída de dados

A entrada de dados, como foi citado anteriormente, é o arquivo csv referente à eleição a ser analisada que deve ser passado como único argumento pela linha de comando. Para isso utilizou-se a funcionalidade `args` presente no cabeçalho da função `main`.

Após a análise, o sistema disponibiliza os resultados como saída na linha de comando. Os dados de saída são (nesta ordem): número de vagas, a lista dos candidatos eleitos, candidatos mais votados, candidatos que se beneficiaram do sistema proporcional (aqueles que foram eleitos mesmo não estando entre os mais votados), candidatos que deveriam ter sido eleitos se a votação fosse majoritária, ranking de coligações, ranking de partidos e total de votos nominais.

As listas que comparam candidatos também mostram o total de votos de cada candidato e seu respectivo partido e coligação (se existir). Da mesma forma, os rankings de coligações e partidos também mostram o total de votos e o número de candidatos eleitos de cada um.

Como explicitado na própria saída do programa, as listas de candidatos beneficiados e prejudicados pelo sistema proporcional mostram a posição de cada um deles baseado na posição do ranking de candidatos mais votados.

3.2 Classes

As classes utilizadas para a manipulação dos dados serão explicitadas nesta seção. As relações entre classes são bem simples e são representadas pelo diagrama UML na Imagem 1.

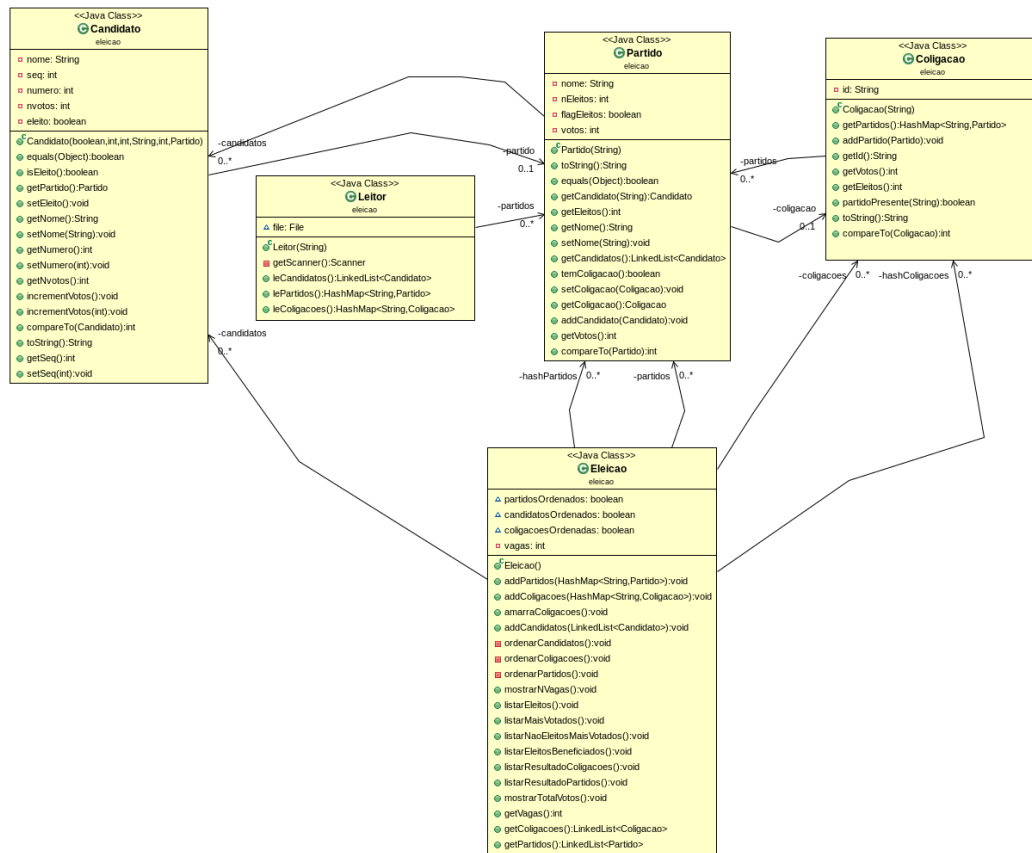


Imagem 1 - Diagrama representativo das classes.

As classes **Candidato**, **Partido** e **Coligação** implementam a interface **Comparable**, disponibilizando o método **compareTo** para realizar as ordenações das listas dessas classes. A ordenação é feita por número de votos e, no caso de partidos e coligações, o número de candidatos eleitos é usado como critério de desempate.

Estas mesmas classes também possuem um *Override* do método **toString** que passa a retornar uma *String* que corresponde à saída na formatação escolhida.

É importante dizer que todos os atributos das classes deste projeto são privados e possuem *getters* e *setters* que permitem manipular seus dados.

3.2.1 Leitor

A classe **Leitor** é uma classe usada somente para a manipulação de dados de entrada. Julgamos conveniente ter uma classe para tal para que pudéssemos obter melhor legibilidade e modularização no código. Uma outra vantagem por exemplo é a possibilidade da criação de vários objetos leitores na classe principal, podendo, caso o usuário queira, manipular o código fonte para mais funcionalidades. Tem como argumento construtor a localização do arquivo de entrada. A classe possui os seguintes métodos:

lePartidos, leColigacoes, leCandidatos: Métodos que se encarregam de ler os partidos, coligações e candidatos presentes no arquivo **.csv** usado como entrada. O funcionamento de todos os três métodos é bastante similar, apenas se utilizando de métodos para manipulação de strings e de um Scanner que se encarrega de delimitar as tokens do arquivo **.csv**. Os métodos para leitura de coligação e partido tem como assinatura **HashMap** enquanto o método para leitura de candidato tem como assinatura **LinkedList** por motivos de conveniência na amarração de cada objeto.

3.2.2 Candidato

A classe **Candidato** representa cada um dos candidatos que participaram das eleições. Esta classe possui atributos que caracterizam o candidato (fornecidos pelo arquivo de entrada). Estes atributos são:

nome: String que representa o nome do candidato.

seq: Inteiro que representa a posição do candidato na lista do arquivo csv.

numero: Inteiro que representa o número de identificação do candidato.

nvotos: Inteiro que representa o número de votos do candidato nestas eleições.

eleito: Booleano utilizado pra identificar se o candidato foi ou não eleito.

partido: Instância da classe **Partido** (tratada na próxima seção) que é utilizada para "amarrar" um candidato ao seu partido.

3.2.3 Partido

A classe **Partido** representa cada um dos partidos que participaram das eleições. Os atributos desta classe são ligeiramente diferentes da classe **Candidato**, pois um partido possui uma lista com seus vários candidatos e pode fazer parte de uma coligação. Seus atributos são:

nome: String que representa o nome do partido.

candidatos: Lista encadeada que contém todos os candidatos pertencentes ao partido.

coligacao: Instância da classe **Coligacao** que é usada para "amarrar" o partido a sua coligação.

nEleitos: Inteiro que representa o número de candidatos eleitos pelo partido. É calculado pela função **getEleitos()** e armazena este valor para o caso da função ser chamada múltiplas vezes, evitando que o valor tenha que ser calculado novamente.

flagEleitos: Booleano utilizado para indicar se o valor de **nEleitos** já foi calculado anteriormente.

votos: Inteiro que representa o total de votos do partido.

3.2.4 Coligacao

A classe `Coligacao` representa cada uma das coligações que participaram das eleições. Os atributos desta classe são ligeiramente diferentes da classe `Partido`, pois uma coligação possui uma lista com seus vários Partidos e pode fazer parte de uma eleição. Seus atributos são:

`id`: O nome da coligação, contendo seus vários partidos em uma `String`.

`partidos`: `HashMap` que contém referência aos objetos referenciando partidos presentes naquela coligação.

3.2.5 Eleicao

A classe `Eleicao` possui vários atributos que podem ser resumidos em duas categorias. Além do inteiro que guarda o número de votos, ela possui coleções e *flags*.

As coleções são compostas por `HashMaps` e `LinkedLists` que contém os partidos, coligações e candidatos referentes à eleição. Os mapas são usados para um melhor desempenho e pela facilidade de utilização das chaves como o atributo `nome` dos objetos. As listas são utilizadas quando precisamos de uma ordenação por votos.

As flags são utilizadas pelos métodos de ordenação para saber se as listas já foram ordenadas, poupando esforço da CPU.

Os métodos desta classe são os métodos responsáveis por organizar e imprimir todos os resultados do programa. Eles realizam ordenações e comparações para imprimir as listas corretamente. Seus próprios nomes são autoexplicativos e podem ser vistos na Imagem 1.

3.2.6 Main

A classe `Main` é a que possui o método *main* e é a responsável por instanciar todas as classes citadas anteriormente e, depois disso, chamar os métodos da classe `Eleicao` para imprimir os resultados.

4 Testes e Known Bugs

Foram feitos três testes para verificar o funcionamento do programa, usando três arquivos .csv diferentes. Foram os arquivos referentes aos municípios de Vitória, Rio de Janeiro e Vila Velha.

4.1 Vitória

O resultado obtido para o município de Vitória se encontra no arquivo `vitória.txt`. Percebemos que foi obtido um partido a mais do que o exemplo de saída do professor, sendo esse o PRTB. Após analisar, vimos que no nosso código, mesmo que um partido não tenha nenhum candidato direto, será contabilizado se estiver em uma coligação. O PRTB se encontra em uma coligação, logo também é contabilizado como partido, sendo essa considerada uma *feature* do nosso código.

```
Número de vagas: 15

Vereadores eleitos:
1 - FABRÍCIO GANDINI (PPS, 7611 votos) - Coligação: PPS / PROS
2 - DENNINHO (PPS, 6167 votos) - Coligação: PPS / PROS
3 - DAVI ESMAEL (PSB, 5165 votos) - Coligação: PSB / PHS
4 - VINÍCIUS SIMÕES (PPS, 4403 votos) - Coligação: PPS / PROS
5 - NEUZINHA (PSDB, 3652 votos) - Coligação: PSC / PSDB
6 - LEONIL (PPS, 3577 votos) - Coligação: PPS / PROS
7 - NATHAN MEDEIROS (PSB, 3316 votos) - Coligação: PSB / PHS
8 - MAZINHO (PSD, 3090 votos) - Coligação: SD / PMB / PSD / PRTB / PTC
9 - DALTO NEVES (PTB, 3083 votos)
10 - MAX DA MATA (PDT, 2339 votos)
11 - WANDERSON MARINHO (PSC, 2208 votos) - Coligação: PSC / PSDB
12 - SANDRO PARRINI (PDT, 2024 votos)
13 - LUIZ PAULO AMORIM (PV, 1696 votos) - Coligação: PP / PV / REDE
14 - ROBERTO MARTINS (PTB, 1547 votos)
15 - CLEBER FELIX (CLEBINHO) (PP, 1524 votos) - Coligação: PP / PV / REDE

Candidatos mais votados (em ordem decrescente de votação e respeitando o número de vagas):
1 - FABRÍCIO GANDINI (PPS, 7611 votos) - Coligação: PPS / PROS
2 - DENNINHO (PPS, 6167 votos) - Coligação: PPS / PROS
3 - DAVI ESMAEL (PSB, 5165 votos) - Coligação: PSB / PHS
4 - VINÍCIUS SIMÕES (PPS, 4403 votos) - Coligação: PPS / PROS
5 - CAMILA VALADÃO (PSOL, 3727 votos)
6 - NEUZINHA (PSDB, 3652 votos) - Coligação: PSC / PSDB
7 - ZEZITO MAIO (PMDB, 3590 votos)
8 - LEONIL (PPS, 3577 votos) - Coligação: PPS / PROS
9 - NATHAN MEDEIROS (PSB, 3316 votos) - Coligação: PSB / PHS
10 - LUIZ EMANUEL (PPS, 3138 votos) - Coligação: PPS / PROS
11 - MAZINHO (PSD, 3090 votos) - Coligação: SD / PMB / PSD / PRTB / PTC
12 - DALTO NEVES (PTB, 3083 votos)
13 - ROGERINHO PINHEIRO (PHS, 2961 votos) - Coligação: PSB / PHS
14 - DEVANIR FERREIRA (PRB, 2948 votos) - Coligação: PRB / PEN / PPL / PSDC / PC do B
15 - MAURÍCIO LEITE (PMDB, 2864 votos)

Teriam sido eleitos se a votação fosse majoritária, e não foram eleitos:
(com sua posição no ranking de mais votados)
5 - CAMILA VALADÃO (PSOL, 3727 votos)
7 - ZEZITO MAIO (PMDB, 3590 votos)
10 - LUIZ EMANUEL (PPS, 3138 votos) - Coligação: PPS / PROS
```

Print do resultado diretamente do terminal.

4.2 Vila Velha

O resultado obtido para o município de Vila Velha se encontra no arquivo `vilavelha.txt`. Discurso análogo ao de Vitória, foi obtido um resultado correto de acordo com as especificações do trabalho, sem maiores discussões se mostrando necessárias.

```
Número de vagas: 17

Vereadores eleitos:
1 - ARNALDINHO BORG (PMDB, 5392 votos) - Coligação: DEM / PMDB / SD
2 - IVAN CARLINI (DEM, 4726 votos) - Coligação: DEM / PMDB / SD
3 - RICARDO CHIABAI (PPS, 4268 votos) - Coligação: PDT / PSB / PPS
4 - ROGERIO CARDOSO (DEM, 3863 votos) - Coligação: DEM / PMDB / SD
5 - ANADELSON PEREIRA (PSDC, 3135 votos) - Coligação: PRB / PSDC / REDE
6 - OSVALDO MATURANO (PRB, 2969 votos) - Coligação: PRB / PSDC / REDE
7 - BRUNO LORENZUTTI (PTN, 2831 votos)
8 - ADEILSON HORTI SUPER (PSD, 2827 votos) - Coligação: PSD / PSC / PV
9 - REGINALDO ALMEIDA (PSC, 2769 votos) - Coligação: PSD / PSC / PV
10 - TIA NILMA (PRP, 2419 votos)
11 - VALDIR DO RESTAURANTE (PT do B, 2346 votos) - Coligação: PP / PT do B / PC do B / PROS
12 - DONA ARLETE (PSL, 2183 votos) - Coligação: PSL / PMB
13 - MIRIM MONTEBELLER (PTN, 2174 votos)
14 - HELIOSANDRO (PR, 2053 votos) - Coligação: PSDB / PR
15 - PATRICIA CRIZANTO (PMB, 2007 votos) - Coligação: PSL / PMB
16 - ZÉ DO RENASCER (PTC, 1941 votos) - Coligação: PTB / PTC / PEN
17 - PM CHICO SIQUEIRA (PHS, 1585 votos) - Coligação: PHS / PMN

Candidatos mais votados (em ordem decrescente de votação e respeitando o número de vagas):
1 - ARNALDINHO BORG (PMDB, 5392 votos) - Coligação: DEM / PMDB / SD
2 - IVAN CARLINI (DEM, 4726 votos) - Coligação: DEM / PMDB / SD
3 - RICARDO CHIABAI (PPS, 4268 votos) - Coligação: PDT / PSB / PPS
4 - ROGERIO CARDOSO (DEM, 3863 votos) - Coligação: DEM / PMDB / SD
5 - JOEL RANGEL (PMDB, 3609 votos) - Coligação: DEM / PMDB / SD
6 - MARCOS RODRIGUES (DEM, 3310 votos) - Coligação: DEM / PMDB / SD
7 - ANADELSON PEREIRA (PSDC, 3135 votos) - Coligação: PRB / PSDC / REDE
8 - OSVALDO MATURANO (PRB, 2969 votos) - Coligação: PRB / PSDC / REDE
9 - BRUNO LORENZUTTI (PTN, 2831 votos)
10 - ADEILSON HORTI SUPER (PSD, 2827 votos) - Coligação: PSD / PSC / PV
11 - REGINALDO ALMEIDA (PSC, 2769 votos) - Coligação: PSD / PSC / PV
12 - TIA NILMA (PRP, 2419 votos)
13 - VALDIR DO RESTAURANTE (PT do B, 2346 votos) - Coligação: PP / PT do B / PC do B / PROS
14 - ZÉ FERREIRA DA MATERNIDADE (PSC, 2198 votos) - Coligação: PSD / PSC / PV
15 - DONA ARLETE (PSL, 2183 votos) - Coligação: PSL / PMB
16 - MIRIM MONTEBELLER (PTN, 2174 votos)
17 - JOÃO ARTEM (PSB, 2087 votos) - Coligação: PDT / PSB / PPS

Teriam sido eleitos se a votação fosse majoritária, e não foram eleitos:
```

Print do resultado diretamente do terminal.

4.3 Rio de Janeiro

O resultado obtido para o município de Rio de Janeiro se encontra no arquivo `rio.txt`. Discurso análogo ao de Vitória e Vila Velha, foi obtido um resultado correto de acordo com as especificações do trabalho, sem maiores discussões se mostrando necessárias

```
Número de vagas: 51

Vereadores eleitos:
1 - CARLOS BOLSONARO (PSC, 106657 votos)
2 - TARCÍSIO MOTTA (PSOL, 98473 votos) - Coligação: PSOL / PCB
3 - CESAR MAIA (DEM, 71468 votos)
4 - ROSA FERNANDES (PMDB, 57868 votos)
5 - MARIELLE FRANCO (PSOL, 46502 votos) - Coligação: PSOL / PCB
6 - JUNIOR DA LUCINHA (PMDB, 45124 votos)
7 - VERA LINS (PP, 36117 votos)
8 - JOÃO MENDES DE JESUS (PRB, 31516 votos)
9 - TERESA BERGHER (PSDB, 30566 votos) - Coligação: PSDB / PPS
10 - LEANDRO LYRA (NOVO, 29217 votos)
11 - CARLO CATIADO (DEM, 28122 votos)
12 - JORGE FELIPPE (PMDB, 28104 votos)
13 - JAIRINHO (PMDB, 26047 votos)
14 - THIAGO K. RIBEIRO (PMDB, 24900 votos)
15 - ALEXANDRE ISQUIERDO (DEM, 24701 votos)
16 - MARCELINO D'ALMEIDA (PP, 24196 votos)
17 - CHIQUINHO BRAZÃO (PMDB, 23923 votos)
18 - TÂNIA BASTOS' (PRB, 22930 votos)
19 - RAFAEL ALOÍSIO FREITAS (PMDB, 22897 votos)
20 - BISPO INALDO SILVA (PRB, 22735 votos)
21 - ZICO (PTB, 21565 votos)
22 - WILLIAN COELHO (PMDB, 20678 votos)
23 - VERÔNICA COSTA (PMDB, 19946 votos)
24 - DR. CARLOS EDUARDO (SD, 19822 votos) - Coligação: SD / PSL
25 - REIMONT (PT, 19626 votos) - Coligação: PT / PC do B
26 - RENATO CINCO (PSOL, 17162 votos) - Coligação: PSOL / PCB
27 - LUCIANA NOVAES (PT, 16679 votos) - Coligação: PT / PC do B
28 - MARCELO ARAR (PTB, 16230 votos)
29 - VAL (PEN, 15388 votos) - Coligação: PRTB / PROS / PEN
30 - PAULO HESSINA (PROS, 15346 votos) - Coligação: PRTB / PROS / PEN
31 - ROGERIO ROCAL (PTB, 15055 votos)
32 - DR. JOÃO RICARDO (PMDB, 14994 votos)
33 - RENATO MOURA (PDT, 13572 votos)
34 - MARCELLO SICILIANO (PHS, 13553 votos)
35 - DRº JORGE MANAIA (SD, 13249 votos) - Coligação: SD / PSL
36 - PAULO PINHEIRO (PSOL, 13141 votos) - Coligação: PSOL / PCB
37 - JONES MOURA (PSD, 12722 votos)
38 - DR GILBERTO (PMN, 12165 votos)
```

Print do resultado diretamente do terminal.

5 Resultados e Avaliação

A implementação desse trabalho em Java foi um passo essencial na compreensão e aprimoramento das nossas competências relacionadas à linguagens de alto nível orientadas a objetos. Os desafios encontrados ao longo do trabalho e a resolução dos mesmos nos tornaram mais íntimos com o desenvolvimento deste novo paradigma.

Além disso, o desenvolvimento da ferramenta não trouxe muitas complicações e permitiu a criação de uma ferramenta útil para estudar uma parcela do processo eleitoral brasileiro, sublinhando sérios problemas na elaboração do mesmo, como partidos fantasmas e problemas com o sistema de quociente. Podemos concluir que a informática é uma ferramenta fundamental na democracia atual.

6 Referências Bibliográficas

<http://docs.oracle.com/javase/8/docs/api/index.html>