UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CIÊNCIA DA COMPUTAÇÃO

KAIO NASCIMENTO LIMA

Processamento de dados da Justiça Eleitoral referentes à votação de deputados federais e estaduais

Vitória
2023

KAIO NASCIMENTO LIMA

Processamento de dados da Justiça Eleitoral referentes à votação de deputados federais e estaduais

Trabalho apresentado para a Disciplina Programação Orientada ao Objeto, pelo Curso de Ciência da Computação da Universidade Federal do Espírito Santo, ministrada pelo Prof. João Paulo Almeida.

Vitória

2023

1. Descrição da Implementação

No contexto do paradigma de programação orientada a objetos, foi apresentada a seguinte questão: Há dois arquivos no formato "CSV", um deles contém os dados de todos os candidatos da eleição atual, enquanto o outro armazena a contagem de votos e os destinatários correspondentes. Com base nesses dois arquivos, nosso software deve gerar 10 tipos distintos de relatórios, exibindo diversas informações relevantes para uma eleição específica. Para atingir esse objetivo, é crucial realizar a coleta e manipulação apropriada dos dados presentes nos arquivos CSV. Dadas as circunstâncias, optei por organizar o código em dois pacotes, denominados "eleicao" e "relatorios".

Dentro do pacote "eleicao", encontram-se as classes Candidato, Partido, Votacao e Eleicao.

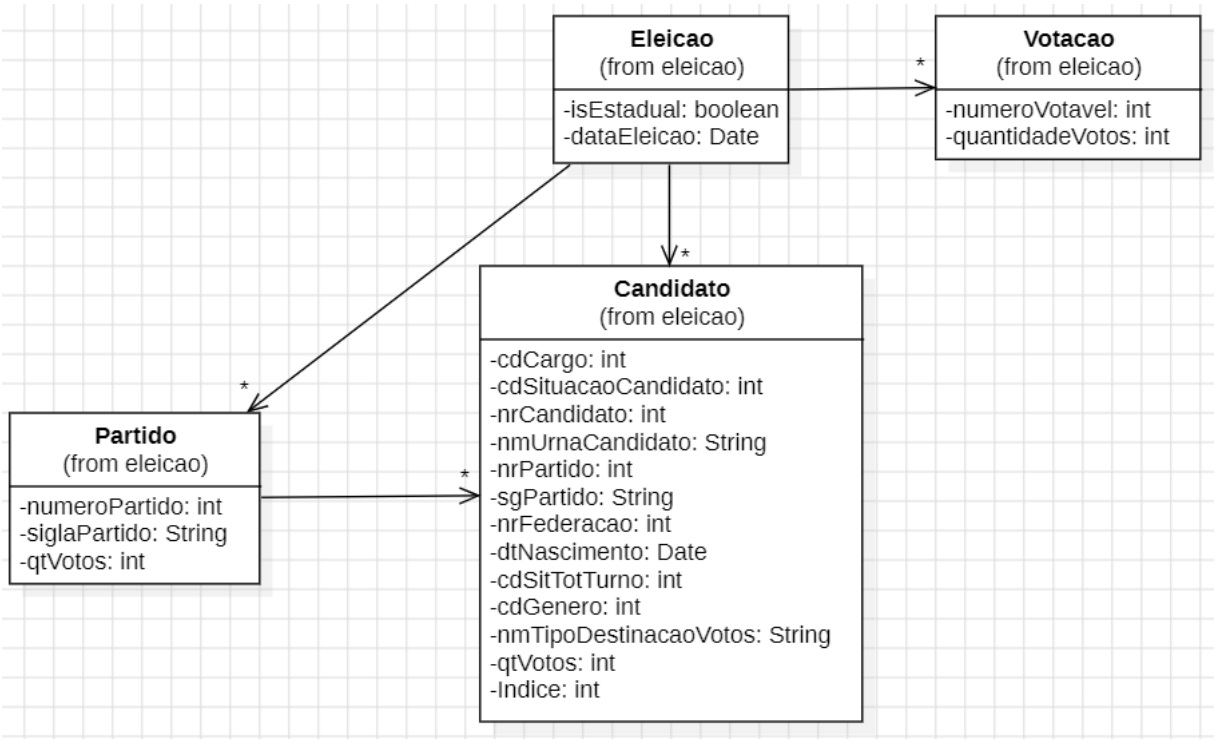

Diagrama UML do pacote Eleicao

A classe "Eleicao" é o núcleo central do programa, onde ocorrem as operações essenciais e fundamentais para o funcionamento de todas as outras classes nos dois pacotes. Nela, são realizadas tarefas como a leitura dos arquivos contendo informações dos candidatos e votações, a conversão dos dados para os tipos de dados apropriados, a atualização da contagem de votos para candidatos e

partidos, o estabelecimento de conexões entre candidatos e partidos e a implementação de métodos para classificar a lista de candidatos com base na quantidade de votos e para retornar o número de candidatos eleitos.

Na classe "Eleicao", encontramos o atributo booleano chamado "isEstadual", o qual é utilizado para processar exclusivamente as votações e candidatos relacionados aos processos eleitorais de Deputados Estaduais e Federais. O valor desse atributo é definido com base no primeiro argumento fornecido na chamada de execução do programa (se foi especificado "--estadual" ou "--federal").

Além disso, a classe "Eleicao" possui o atributo "dataEleicao" do tipo Date, que armazena a data da eleição. Esse atributo é utilizado principalmente para realizar cálculos e operações nos relatórios.

Conforme ilustrado no diagrama UML apresentado acima, a classe "Eleicao" também mantém coleções de objetos do tipo Votacao, Candidato e Partido. Essas coleções são fundamentais para armazenar e gerenciar as informações relacionadas às votações, candidatos e partidos envolvidos no processo eleitoral.

A classe "Partido" encapsula informações sobre partidos políticos, incluindo seu número, sigla/nome, quantidade de votos (votos de legenda) e uma coleção de candidatos associados a esse partido.

A classe "Votacao" desempenha o papel de representar cada linha do arquivo CSV de votações, abrangendo tanto informações sobre candidatos quanto partidos.

A classe “Candidato” possui atributos essenciais que desempenham um papel fundamental tanto no processamento como na manipulação dos dados relacionados a esse candidato, garantindo a adequada exibição nos relatórios. Esses atributos são cuidadosamente selecionados para contemplar as informações relevantes ao contexto do processamento eleitoral.

Dentro do pacote “relatorio”, encontram-se as classes Relatorio, RelatorioCandidato, RelatorioPartido e RelatorioEstatistico.

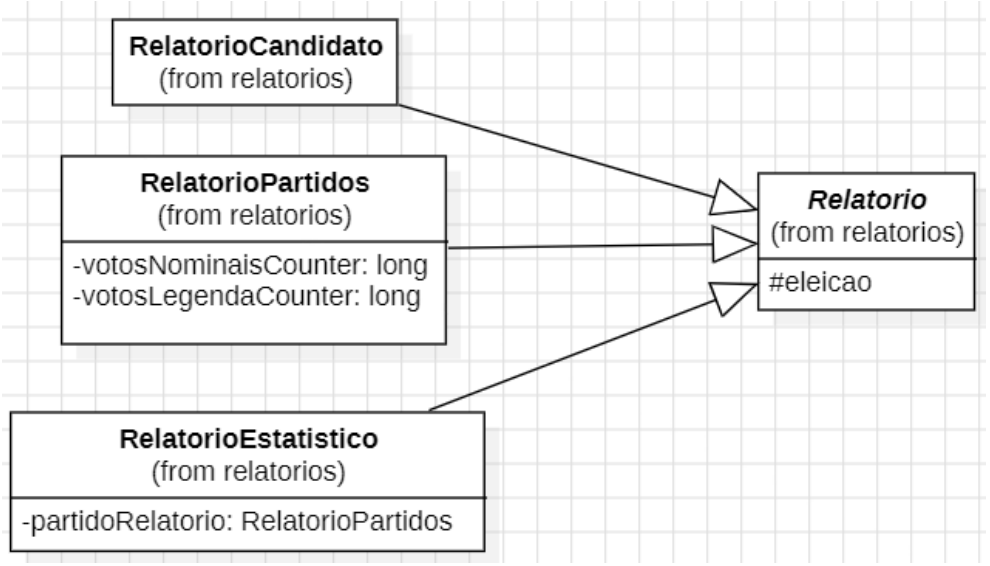

Diagrama UML do pacote "relatorios"

A classe "Relatorio" é uma classe abstrata que possui um atributo do tipo Eleicao. Todas as outras classes dentro desse pacote herdam esse atributo, estabelecendo uma relação de herança que permite o acesso e manipulação dos dados da eleição durante a geração dos relatórios.

É relevante destacar que cada classe herdeira realiza a manipulação adequada de strings, abordando questões de concordância em plural e singular, entre outros aspectos, assim como a formatação apropriada para números inteiros e de ponto flutuante na geração das saídas dos relatórios.

A classe RelatorioCandidato é responsável por utilizar os dados da Eleicao e fazer a impressão dos 5 primeiros relatórios solicitados, sendo eles:

1. Número de vagas (= número de eleitos);

2. Candidatos eleitos (nome completo e na urna), indicado partido e número de votos nominais;

3. Candidatos mais votados dentro do número de vagas;

4. Candidatos não eleitos e que seriam eleitos se a votação fosse majoritária;

5. Candidatos eleitos no sistema proporcional vigente, e que não seriam eleitos se a votação fosse majoritária, isto é, pelo número de votos apenas que um candidato recebe diretamente;

A classe RelatorioPartido também usa os dados da Eleicao para fazer a impressão dos relatórios 6 e 7, sendo eles:

6. Votos totalizados por partido e número de candidatos eleitos;

7. Primeiro e último colocados de cada partido (com nome da urna, número do candidato e total de votos nominais). Partidos que não possuírem candidatos com um número positivo de votos válidos devem ser ignorados;

A classe RelatorioEstatistico usa os dados da Eleicao e da RelatorioPartidos para fazer a impressão dos 3 relatórios restantes, sendo eles:

8. Distribuição de eleitos por faixa etária, considerando a idade do candidato no dia da eleição;

9. Distribuição de eleitos por sexo;

10. Total de votos, total de votos nominais e total de votos de legenda.

Na pasta "src", encontra-se o arquivo "App.class" que contém o método principal (main). Optei por manter este arquivo de forma concisa, concentrando apenas o necessário para possibilitar a visualização clara da estrutura do programa e até mesmo permitir a escolha do relatório desejado para impressão. Essa abordagem visa facilitar a compreensão e navegação no código-fonte.

```
//Gera Eleição
Eleicao eleicao = new Eleicao(isEstadual, arquivoCandidatos, arquivoVotacao, dataEleicao);

//Gera Relatórios
RelatorioCandidato relatorio = new RelatorioCandidato(eleicao);
relatorio.imprimirRelatorio();

RelatorioPartidos relatorioPartidos = new RelatorioPartidos(eleicao);
relatorioPartidos.gerarRelatorioPartidos();

RelatorioEstatistico relatorioEstatistico = new RelatorioEstatistico(eleicao, relatorioPartidos);
relatorioEstatistico.gerarRelatorioEstatistico();
```

Trecho do método main

2. Testes

Neste tópico, será possível observar a comparação entre os resultados esperados e obtidos em alguns dos testes realizados para analisar a exatidão do programa. Foram realizados testes tanto para deputados estaduais quanto federais.

- Alagoas (Estadual)
  - Arquivo

Resultado esperado:

Observe os índices 13 e 20 do resultado esperado e suas similaridades

```
13 - PSDB - 45, 3.996 votos (0 nominal e 3.996 de legenda), 0 candidato eleito
14 - PTB - 14, 2.656 votos (734 nominais e 1.922 de legenda), 0 candidato eleito
15 - PSOL - 50, 1.683 votos (905 nominais e 778 de legenda), 0 candidato eleito
16 - UP - 80, 1.453 votos (1.327 nominais e 126 de legenda), 0 candidato eleito
17 - SOLIDARIEDADE - 77, 1.348 votos (351 nominais e 997 de legenda), 0 candidato eleito
18 - PMB - 35, 484 votos (283 nominais e 201 de legenda), 0 candidato eleito
19 - PSTU - 16, 379 votos (86 nominais e 293 de legenda), 0 candidato eleito
20 - CIDADANIA - 23, 348 votos (0 nominal e 348 de legenda), 0 candidato eleito
21 - DC - 27, 140 votos (98 nominais e 42 de legenda), 0 candidato eleito
```

Considere agora o total de votos de legenda e o total de votos

```
Total de votos válidos:    1.662.218
Total de votos nominais:   1.549.561 (93,22%)
Total de votos de legenda: 112.657 (6,78%)
```

Resultado obtido:

```
13 - PTB - 14, 2.656 votos (734 nominais e 1.922 de legenda), 0 candidato eleito
14 - PSOL - 50, 1.683 votos (905 nominais e 778 de legenda), 0 candidato eleito
15 - UP - 80, 1.453 votos (1.327 nominais e 126 de legenda), 0 candidato eleito
16 - SOLIDARIEDADE - 77, 1.348 votos (351 nominais e 997 de legenda), 0 candidato eleito
17 - PMB - 35, 484 votos (283 nominais e 201 de legenda), 0 candidato eleito
18 - PSTU - 16, 379 votos (86 nominais e 293 de legenda), 0 candidato eleito
19 - DC - 27, 140 votos (98 nominais e 42 de legenda), 0 candidato eleito
```

```
Total de votos válidos: 1.657.874
Total de votos nominais: 1.549.561 (93.47%)
Total de votos de legenda: 108.313 (6.53%)
```

Com exceção dessas pequenas divergências, o programa alcançou os resultados esperados nos demais relatórios.

- Acre (Estadual)
  - Arquivo

Resultado esperado:

```
21 - REDE - 18, 152 votos (0 nominal e 152 de legenda), 0 candidato eleito
22 - PMN - 33, 138 votos (71 nominais e 67 de legenda), 0 candidato eleito
```

```
Total de votos válidos:    435.770
Total de votos nominais:   411.114 (94,34%)
Total de votos de legenda: 24.656 (5,66%)
```

Resultado obtido:

```
21 - PMN - 33, 138 votos (71 nominais e 67 de legenda), 0 candidato eleito
```

```
Total de votos válidos: 435.618
Total de votos nominais: 411.114 (94.37%)
Total de votos de legenda: 24.504 (5.63%)
```

Com exceção dessas pequenas divergências, o programa alcançou os resultados esperados nos demais relatórios.

- Pernambuco (Federal)
  - Arquivo

Resultado esperado:

```
30 - PROS - 90, 827 votos (0 nominal e 827 de legenda), 0 candidato eleito
31 - PMB - 35, 408 votos (27 nominais e 381 de legenda), 0 candidato eleito
32 - PCO - 29, 126 votos (0 nominal e 126 de legenda), 0 candidato eleito
```

```
Total de votos válidos:    4.970.816
Total de votos nominais:   4.777.467 (96,11%)
Total de votos de legenda: 193.349 (3,89%)
```

Resultado obtido:

```
30 - PMB - 35, 408 votos (27 nominais e 381 de legenda), 0 candidato eleito
```

```
Total de votos válidos: 4.969.863
Total de votos nominais: 4.777.467 (96.13%)
Total de votos de legenda: 192.396 (3.87%)
```

Com exceção dessas pequenas divergências, o programa alcançou os resultados esperados nos demais relatórios.

- Rio Grande do Sul e Minas Gerais (Estadual)
  - Arquivo RS
  - Arquivo MG

Obtivemos um resultado totalmente satisfatório, alcançando 100% de precisão nos dados

Em geral, a conclusão com base nos testes revela uma uniformidade nas divergências das comparações, impactando principalmente nos resultados dos relatórios 6 (votos totalizados por partido e número de candidatos eleitos) e 10 (total de votos, total de votos nominais e total de votos de legenda) dos testes.

3. Bugs conhecidos (*“known bugs”*)

Considerando os testes conduzidos mediante a utilização dos arquivos providos pelo TSE e a execução do script de testes disponibilizado no Classroom, foi identificado um “bug” que influencia, em proporções mínimas, os resultados referentes aos votos de legenda. A origem desse problema reside na etapa de leitura do arquivo de votação, onde não são preservados nas coleções os partidos que não apresentam votos nominais, mesmo que possuam votos de legenda, isto é, partidos que não estão vinculados a nenhum candidato. Embora essa constatação seja evidente, nota-se que, ao considerar o resultado em porcentagem, a margem de divergência entre os dados obtidos e os dados esperados permanece, em sua maioria, relativamente pequena.

4. Conclusão

O trabalho em questão apresenta um sistema robusto para processamento e análise de dados eleitorais, empregando o paradigma de programação orientada a objetos em Java. A implementação abrange a leitura de arquivos CSV que contém informações sobre candidatos e votações, seguida pela manipulação dos dados para gerar relatórios eleitorais. A estrutura do programa, organizada nos pacotes "eleicao" e "relatorios", demonstra clareza e modularidade, fomentando a reusabilidade e manutenção do código. Os testes realizados indicam resultados predominantemente satisfatórios, evidenciando a eficácia do sistema na produção de relatórios precisos, mesmo diante de pequenas divergências em casos específicos. Apesar dessas mínimas discrepâncias, o programa atingiu seus objetivos, destacando-se pela capacidade de proporcionar análises detalhadas do processo eleitoral e contribuir significativamente para a compreensão e avaliação dos resultados eleitorais.

5. Links úteis

O código também está disponível no meu repositório do GitHub, possibilitando futuras correções, otimizações e adições. Isso oferece uma plataforma de controle de versão para o desenvolvimento contínuo do projeto.

Repositório (GitHub) | analise-eleitoral-deputados

Linkedin