UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA
PROGRAMAÇÃO III

Dayane Silva Erlacher Castro
Gabrielly Barcelos Cariman

PRIMEIRO TRABALHO DE IMPLEMENTAÇÃO JAVA

Vitória-ES

Fevereiro/2022

LISTA DE FIGURAS

SUMÁRIO

1 DESCRIÇÃO DA IMPLEMENTAÇÃO

Este trabalho consiste em um sistema para processar dados obtidos da justiça eleitoral referentes à votação de vereadores em um município. O objetivo é desenvolver os conceitos de orientação a objeto, utilizando a linguagem Java.

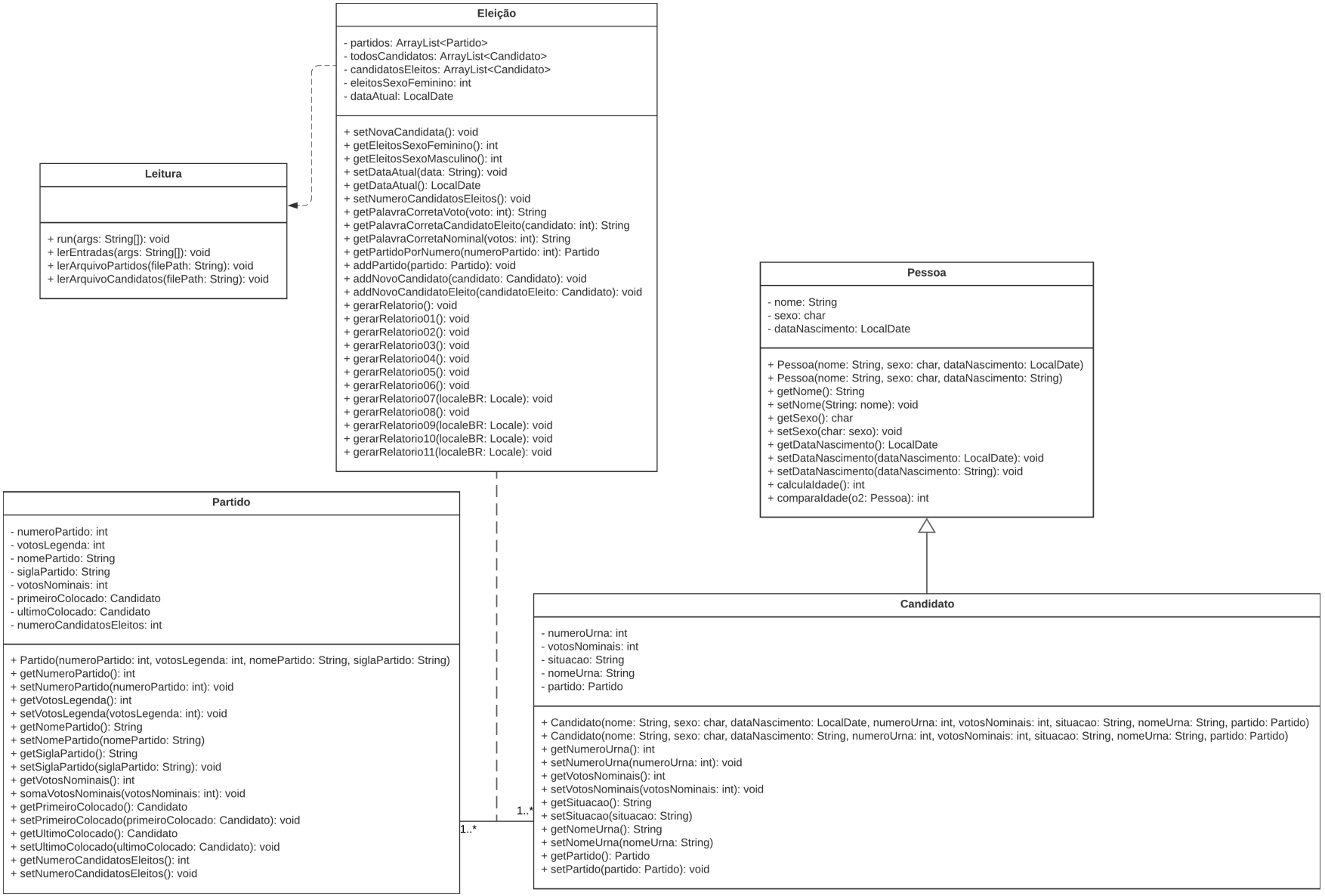
O projeto contém seis classes principais, sendo elas:

- Pessoa
- Candidato
- Partido
- Leitura
- Eleição
- Main

1.1 Relacionamento e modelagem das classes

A modelagem das classes, o que cada uma possui de atributos, métodos e os seus relacionamentos, pode ser observado no diagrama de classes UML da Figura 1, desenvolvido na plataforma Lucidchart ().

Figura 1 – Diagrama de Classes UML



Fonte: Produção do próprio autor.

A classe Pessoa possui como atributo nome, sexo e data de nascimento, ela possui métodos que lidam com esses atributos e os *getters* e *setters*. Além disso, ela apresenta um relacionamento de especialização com a classe Candidato, porque Candidato é um tipo mais especifico de Pessoa e representa a herança recebida de Candidato de sua classe mãe.

As classes Candidato e Partido apresentam os métodos *getters* e *setters* para manipularem os seus atributos privados. Além disso, elas também possuem classes internas que implementam a interface *Comparator* para ordenar as listas de Candidatos e Partidos, de acordo com o formato pedido nos relatórios. Essas classes apresentam o relacionamento bidirecional de associação. Isso porque, um Partido contém no mínimo um e no máximo dois Candidatos que representam o primeiro e o ultimo colocado, enquanto um Candidato possui como atributo apenas um Partido, porém vários Candidatos podem possuir o mesmo Partido como atributo.

A classe Eleição foi projetada para gerar e gerenciar todas as manipulações necessárias para que os relatórios fossem produzidos. Como atributos necessários para a geração dos relatórios, essa classe apresenta a data atual que é informada por linha de comando, a quantidade de eleitos que são do gênero feminino, um *ArrayList* com todos os Partidos e dois com Candidatos, em que um contém todos os Candidatos da Eleição e o outro apenas os Candidatos Eleitos. Além dos *getters*, *setters* e outros métodos usados para manipular os atributos privados da classe, existe o método gerar relatório que chama outros métodos específicos de cada relatório que formata as saídas desejadas manipulando os dados necessários. Todos os atributos dessa classe são estáticos, mesmo que isso torne o projeto um pouco mais procedural, o objetivo era que a informação ficasse guardada pela classe e que não fosse mais individual para cada objeto, para que o programa lide com apenas uma Eleição por vez. Com isso, todos os métodos também são estáticos para manipular atributos do mesmo tipo. Eleição é considerada uma classe associativa com as classes Candidatos e Partidos. Isso porque, muitos Candidatos e muitos Partidos participam de um Eleição e seus atributos não poderiam ser armazenados nas classes Partido e Candidato, porque esses atributos não fazem sentido em um contexto fora o da Eleição.

A classe Leitura foi produzida apenas para manipular a leitura dos CSVs de entradas e instanciação dos objetos criados a partir das leituras, por isso, todos os seus métodos são estáticos e não possui atributos. Seu método *run* é o primeiro e único método chamado na *Main*, a partir dele, todas as funcionalidades do programa são ativadas. Essa classe apresenta um relacionamento de dependência com a classe Eleição, porque a Classe Eleição depende das informações que a classe Leitura coleta para a definição de seus atributos.

1.2 Tratamento de exceções

As exceções tratadas nesse projeto partem do princípio de que o usuário pode errar suas entradas de dados ou os documentos podem estar com erros de digitação ou preenchimento. Toda a lógica do código depende de três entradas de dados, logo, o único método chamado na *Main* trata todas essas exceções de forma a impedir que o método de gerar relatórios seja chamado, caso ocorra uma exceção.

O principal método, chamado *run*, chama o método *lerEntradas* que lança apenas uma exceção. Essa verifica se existe a quantidade de argumentos esperados, caso não tenha a exceção *IllegalArgumentException* é lançada com a seguinte mensagem: "3 parâmetros são esperados". O método chama três outros em sequência, um para cada argumento esperado.

O primeiro método é para salvar a data da eleição, ela deve ser uma *String* e possuir tamanho 10 (dd/MM/yyyy), caso não possua esses requisitos, a *FormatoIncorretoDataArgumentoException* é lançada com a seguinte mensagem: "O terceiro parâmetro esperado é uma data no formato dd/MM/yyyy". Após, verificamos se os campos dia e mês são válidos, caso algum não seja, a exceção *DataFormatException* é lançada com a mensagem: "Data inválida. Campo dia maior que 31 ou campo mês maior que 12".

Os dois seguintes métodos manipulam arquivos, logo a *FileNotFoundException* e *IOException* são lançadas. Eles ainda possuem uma exceção quanto ao cabeçalho do arquivo CSV. Para partidos, a *FormatoArquivoPartidoIncorretoException* com a mensagem: "Cabeçalho do csv de Partidos não está no formato esperado". Para candidatos, a *FormatoArquivoCandidatoIncorretoException* com a mensagem: "Cabeçalho do csv de Candidato não está no formato esperado". Ademais, as classes possuem uma exceção para tratar o caso em que o arquivo esteja completamente vazio, a *ArquivoEntradaVazioException*. Além dessas exceções, os métodos possuem uma condicional para verificar se cada linha do arquivo possui exatamente a quantidade esperada de dados, caso não tenha, ela não é considerada. Ou seja, o partido ou candidato não é adicionado a base de dados.

Ademais, o método de Partidos possui outra verificação, caso a data de nascimento não esteja em um formato aceitável, mesmos critérios especificados para a data da eleição, a linha de dados é desconsiderada. Isto é, o candidato não é adicionado a base de dados.

O restante do código não possui exceções, visto que todos os dados manipulados na geração de relatórios dependem diretamente dos dados de entrada. Todas as exceções foram mapeadas para evitar erros, caso qualquer uma seja acionada os relatórios não são

gerados. Isso porque, caso uma exceção for lançada os dados necessários para gerá-los são comprometidos. Para casos mais simples, não geramos exceções e descartamos os dados corrompidos.

2 DESCRIÇÃO DOS TESTES

Inicialmente, foram realizados testes específicos sob as funções, a fim de confirmar se todos cumpriam os pré requisitos. Após, foram desenvolvidos os métodos que imprimem os relatórios exatamente como especificado. A partir desse momento, os testes foram realizados, primeiramente, utilizando as entradas referentes ao município de Vitória e com checagem manual. Posteriormente, com o *script* desenvolvido e disponibilizado pelo professor.

Caso a entrada de dados esteja incompleta, com falta de dados, a linha é descartada. Essa possível fonte de erro na construção do objeto foi identificada nos testes e para evitar foi implementado a solução.

3 BUGS

Nos teste realizados não foram encontrados *bugs*. Um detalhe registrado é o caso não especificado. Existir dois candidatos com o mesmo número de votos nominais e a mesma idade. Para esse caso, o comportamento do código é de nos relatórios de 2 a 5, a ordem do arquivo de entrada é respeitada. Ou seja, o formato aplicado é o de fila, o primeiro candidato nessa situação ocupa a colocação mais alta. Para o relatório 8 funciona da mesma maneira, o primeiro a ser definido "Primeiro lugar em número de votos do Partido" se manterá na posição. Porém, o mesmo se aplica ao "Último lugar em número de votos do Partido", o primeiro candidato definido nessa categoria se manterá nela.

Quando os arquivos possuem a linha de cabeçalho, mas não possuem os dados, não existe eleição, logo a saída será vazia, contendo apenas os títulos. Quando apenas o arquivo de partidos está vazio, também não existe eleição e a saída também é vazia. Porém, quando apenas o arquivo de candidatos está vazio, a saída conterá alguns dados dos partidos. Isso porque os partidos existem mesmo que não ocorra eleição no momento.

REFERÊNCIAS

LUCIDCHART. Disponível em: <https://www.lucidchart.com/pages/pt/landing?utm_source=google&utm_medium=cpc&utm_campaign=_chart_pt_allcountries_mixed_search_brand_exact_&km_CPC_CampaignId=1500131167&km_CPC_AdGroupID=59412157138&km_CPC_Keyword=lucidchart&km_CPC_MatchType=e&km_CPC_ExtensionID=&km_CPC_Network=g&km_CPC_AdPosition=&km_CPC_Creative=294337318298&km_CPC_TargetID=kwd-33511936169&km_CPC_Country=1031797&km_CPC_Device=c&km_CPC_placement=&km_CPC_target=&mkwid=seFG43w3S_pcrid_294337318298_pkw_lucidchart_pmt_e_pdv_c_slid___pgrid_59412157138_ptaid_kwd-33511936169_&gclid=CjwKCAjw9uKIBhA8EiwAYPUS3Ea5dWA8cWLzUHVOTkbkMxw5RMQwUEH2ItcKpVxdm18U7hTBwE>. Acesso em: 07 fev. 2022. Citado na página 3.