

# Desenvolvimento de Sistemas de Software

## MIEINF

### 2015/2016

Octávio Maia



a71369

João Silva



a72023

Rui Freitas



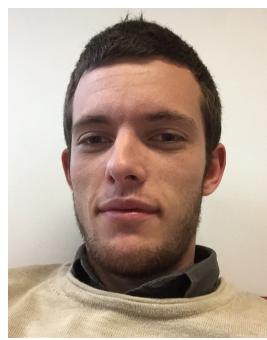
a72399

Pedro Pinto



a71929

Nuno Oliveira



a72028

30 de Dezembro de 2015



# Conteúdo

<b>1 Parte 1</b>	<b>3</b>
1.1 Introdução . . . . .	3
1.2 Modelo de Domínio . . . . .	4
1.2.1 Assembleia da República . . . . .	4
1.2.2 Presidência da República . . . . .	5
1.3 Modelos de Use Case e Diagramas de Sequência Software . . . . .	7
1.3.1 Cria Eleição . . . . .	7
1.3.2 Vota (Assembleia da República) . . . . .	8
1.4 Proposta de Interface (Mockup) e Máquinas de Estado . . . . .	9
1.4.1 Janela de login . . . . .	9
1.4.2 Gestão de eleição (Administrador) . . . . .	10
1.4.2.1 Criar Eleição . . . . .	11
1.4.2.2 Gerir Eleições da Assembleia da República . . . . .	11
1.4.2.3 Gerir Eleições da Presidência da República . . . . .	13
1.4.2.4 Inserção do Caderno de Recenseamento . . . . .	14
1.4.3 Janela Principal do eleitor . . . . .	15
1.4.3.1 Boletim de voto . . . . .	16
1.4.3.2 Resultados Assembleia República . . . . .	17
1.4.3.3 Resultados Presidência República . . . . .	18
1.5 Conclusão 1 <sup>a</sup> parte . . . . .	20
<b>2 Parte 2</b>	<b>21</b>
2.1 Diagramas Visual Paradigm . . . . .	21
2.1.1 Diagramas de Estrutura . . . . .	21
2.1.1.1 Diagrama de Classes . . . . .	21
2.1.1.2 Diagrama de Packages . . . . .	23
2.1.1.3 Diagrama de Instalação . . . . .	24
2.1.2 Diagramas de Interação . . . . .	25
2.1.2.1 Diagrama Sequencia Software . . . . .	25
2.2 Código . . . . .	26
2.2.1 Classes principais . . . . .	26

2.2.2	DAOs . . . . .	29
2.2.2.1	Estratégias utilizadas para otimizar os DAOs e Base de Dados . . . . .	29
2.2.3	Interface . . . . .	31
2.3	Decisões Estruturais . . . . .	37
2.4	Conclusão parte 2 . . . . .	38
.1	Anexo . . . . .	39
.1.1	Modelo Lógico da Base de Dados . . . . .	39

# Capítulo 1

## Parte 1

### 1.1 Introdução

Este trabalho prático tem como base a criação de um sistema de suporte à realização de eleições.

Como indicado no enunciado o sistema possui dois modos distintos de funcionamento.

- Permite a criação de um ato eleitoral, permitindo definir o tipo de eleição, candidatos, listas, partidos, data limite, entre outros.
- Por outro, permite também a gerência do ato eleitoral, ou seja, permite o registo de votos, atribuição de mandatos, cálculo do vencedor, criação de segunda volta, entre outros.

Como referido no enunciado, os dois modos de funcionamento foram disponibilizados numa única aplicação, havendo distinção entre os votantes e os administradores do sistema.

A aplicação em causa tem como base as normas disponibilizadas no site da Comissão Nacional de Eleições<sup>1</sup>, para os atos eleitorais realizados em Portugal e suporta dois tipos de eleição:

1. Assembleia da República
2. Presidência da República

Nesta fase do projeto iremos entregar a fase inicial do projeto, contendo os Modelos de Domínio/Use Case, bem como os Diagramas de Sequencia Software/Máquinas de Estado e uma proposta da futura interface *Mockup*.

---

<sup>1</sup><http://www.cne.pt/content/legislacao-eleitoral>

## 1.2 Modelo de Domínio

O modelos de domínio que propomos, são os apresentados de seguida que tentam abordar todos os aspetos importantes para o suporte das funcionalidades que é suposto a aplicação dar suporte, e ao mesmo tempo captar todos os aspetos legais importantes para seguir esses requisitos, abstraindo de aspetos não essenciais para a aplicação, tais como a gestão de tarefas administrativas e burocráticas.

### 1.2.1 Assembleia da República

O modelo de domínio da Assembleia da República é representado por uma eleição, que é efectuada numa determinada data, e tem um caderno de recenseamento, onde contém as informações de todos os cidadãos permitidos a votar. Cada eleição cria um mapa de eleição quando iniciada, e preenche uma assembleia. As assembleias são constituídas por deputados, cada um deles correspondentes a um mandato, eleito por um círculo eleitoral. Cada círculo eleitoral pode ou não ter listas candidatas a Assembleia. Cada lista pode ser singular, se for constituída por apenas um partido, ou pode tratar-se de uma coligação, caso o número de partidos seja maior que 1. Tanto os partidos como as coligações são distinguidos pela sua nomeação, símbolo e sigla. Os partidos são liderados por um líder, que por sua vez pode também ser um eleitor. Existe um candidato suplente, para prevenir situações irregulares que possam acontecer com o candidato principal. Para identificação dos candidatos, decidimos atribuir-lhes idade, profissão, residência, naturalidade, filiação, e o número de cartão de cidadão.

A ordem de apresentação no boletim de voto de cada círculo eleitoral é gerada através de um sorteio. O boletim de voto é o espaço onde os eleitores podem deixar o seu voto. Voto este que pode ser válido, quando apenas é selecionado um partido no boletim de voto; branco, quando não é selecionado nenhum partido; e nulo, quando o voto do eleitor não é válido. No nosso projeto, cada eleitor é identificado por um número identificador, um PIN e pelo seu nome.

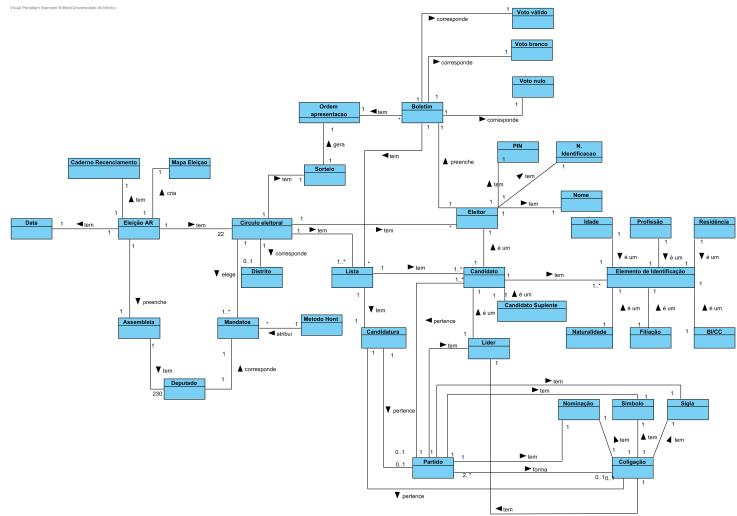


Figura 1.1: Modelo de Domínio da Assembleia da República.

### 1.2.2 Presidência da República

O modelo de domínio para às eleições da Presidência da República o grupo decidiu que toda a informação relativa aos eleitores deveria ser mantida igual as eleições da **Assembleia da República**, mesmo que não seja totalmente necessária para este tipo de eleição, como a informação relativa ao círculo pertencente a que pertence cada eleitor isso também facilitaria todo o processo de gestão dos cadernos de recenseamento que passa a ser igual. Por outro lado ao mantendo a informação dos círculos podemos oferecer na parte de consulta de resultados eleitorais informação com mais detalhe para cada círculo eleitoral.

Em questões mais relacionadas com a tipologia da eleição, esta é mais simples que a eleição da **Assembleia da República**, pois as listas que são apresentadas, são comuns a todo o país, eliminando a necessidade de diferenciar as listas para cada círculo e gestão dos boletins de voto. Também esta eleição na sua essência a lista é apenas constituída pelo candidato que corre à presidência não sendo necessário no nosso ponto de vista a aplicação guardar toda a restante informação, como quem apoiou a candidatura.

Outra diferença relativa as eleições da **Assembleia da República** é pelo facto de caso não seja obtida maioria absoluta na primeira volta por parte de um candidato ser necessário recorrer a uma nova eleição de segunda volta entre os dois candidatos mais votados na primeira volta, sendo certo que nesta segunda volta a decisão da eleição é tomada.

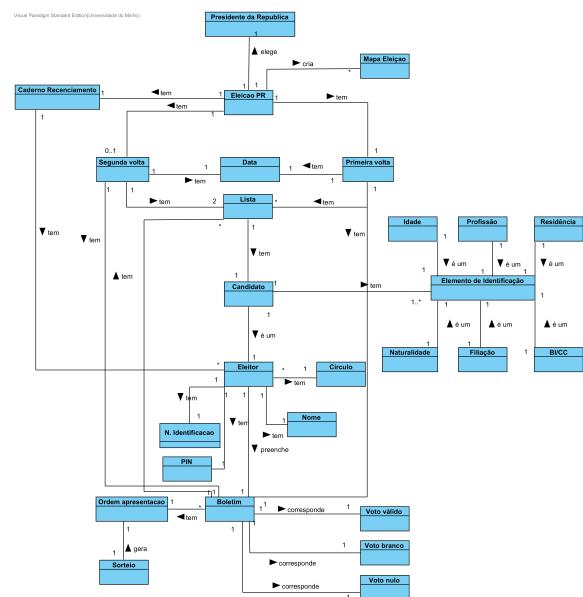


Figura 1.2: Modelo de Domínio da Presidência da República.

## 1.3 Modelos de Use Case e Diagramas de Sequência Software

Nesta secção definimos o modo de funcionamento da nossa aplicação. Aqui já começamos a pensar um pouco nas classes ou subsistemas que vamos utilizar. Entre os vários use cases criados, destacamos os seguintes:

### 1.3.1 Cria Eleição

Neste Use Case, o primeiro passo é pedir ao Administrador a data de início da eleição, a data de fim, os partidos participantes e as listas.

Após a obtenção de toda esta informação o subsistema de eleições cria a Eleição.

Mesmo após a eleição ter sido criada, é possível adicionar novas listas. Isto é feito pedindo ao sistema de eleições a adição de uma nova lista, que por sua vez pede ao subsistema de círculos a adição da lista. Após o subsistema de círculos adicionar a lista este reporta ao subsistema de eleições a adição com sucesso, que por sua vez reporta à interface.

Super Use Case	
Author	Octávio
Date	4Nov2015 16:58:41
Brief Description	Criação de uma eleição para a Assembleia da República
Preconditions	Não existe eleição deste tipo
Post-conditions	Eleição criada
Flow of Events	
Actor Input	
1	Pede data de inicio
2	Insere data de inicio
3	Pede data de fim
4	Insere data de fim
5	Pede partidos
6	Insere partidos
7	Pede listas
8	<<include>> Cria lista
9	Cria eleição
10	Adiciona listas à eleição
Alternativa 1 Insere mais listas (Passo 9)	
Actor Input	
1	Regressa a 8

Figura 1.3: Diagrama de Use Case do Cria Eleição.

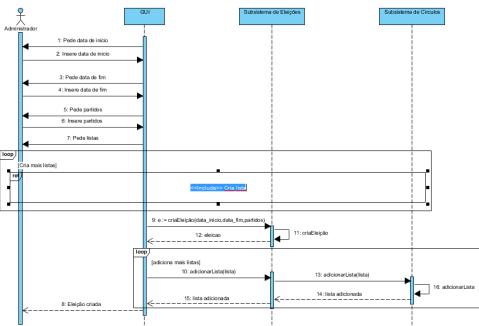


Figura 1.4: Diagrama de Sequência do Cria Eleição.

### 1.3.2 Vota (Assembleia da República)

Neste use case, o primeiro passo é pedir ao subsistema de eleição o boletim de voto do círculo ao qual o eleitor pertence. Para obter essa informação, é necessário ir ao subsistema do eleitor ver o seu círculo e, de seguida, para esse círculo, obter o seu boletim de voto.

Após a obtenção do boletim, este é mostrado ao eleitor, que tem três opções: ou escolhe apenas uma lista e atribui o seu voto, ou não escolhe nenhuma lista e atribui um voto em branco, ou escolhe múltiplas listas, e o voto é nulo. De seguida, o voto é registado na eleição e círculo em questão.

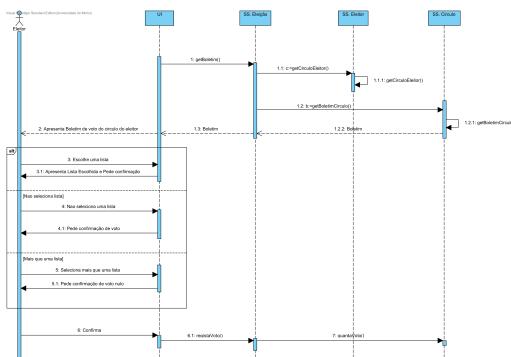


Figura 1.5: Diagrama de Sequência do Vota da Assembleia da República.

<b>Super Use Case</b>	Octávio													
<b>Author</b>	Octávio													
<b>Date</b>	4 Nov 2015 16:55:39													
<b>Brief Description</b>	Descrição do processo de voto.													
<b>Preconditions</b>	Existe eleição ativa, eleitor ainda não votou e boletim gerado.													
<b>Post-conditions</b>	Voto efetuado.													
<b>Flow of Events</b>	<table border="1"> <thead> <tr> <th>Actor Input</th> <th>System Response</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Apresenta boletim de voto do círculo do eleitor.</td> </tr> <tr> <td>2 Seleciona uma lista</td> <td></td> </tr> <tr> <td>3</td> <td>Apresenta lista escolhida e pede confirmação.</td> </tr> <tr> <td>4 Confirma</td> <td></td> </tr> <tr> <td>5</td> <td>Regista voto.</td> </tr> </tbody> </table>		Actor Input	System Response	1	Apresenta boletim de voto do círculo do eleitor.	2 Seleciona uma lista		3	Apresenta lista escolhida e pede confirmação.	4 Confirma		5	Regista voto.
Actor Input	System Response													
1	Apresenta boletim de voto do círculo do eleitor.													
2 Seleciona uma lista														
3	Apresenta lista escolhida e pede confirmação.													
4 Confirma														
5	Regista voto.													
<b>Alternativa 1</b> [Mais que uma lista] (Passo 2)	<table border="1"> <thead> <tr> <th>Actor Input</th> <th>System Response</th> </tr> </thead> <tbody> <tr> <td>1 Seleciona mais que uma lista</td> <td>Pede confirmação de voto nulo.</td> </tr> <tr> <td>2</td> <td></td> </tr> <tr> <td>3</td> <td>Regressa a 4</td> </tr> </tbody> </table>		Actor Input	System Response	1 Seleciona mais que uma lista	Pede confirmação de voto nulo.	2		3	Regressa a 4				
Actor Input	System Response													
1 Seleciona mais que uma lista	Pede confirmação de voto nulo.													
2														
3	Regressa a 4													
<b>Alternativa 2</b> [Não seleciona lista] (Passo 2)	<table border="1"> <thead> <tr> <th>Actor Input</th> <th>System Response</th> </tr> </thead> <tbody> <tr> <td>1 Não seleciona uma lista</td> <td>Pede confirmação de voto nulo.</td> </tr> <tr> <td>2</td> <td></td> </tr> <tr> <td>3</td> <td>Regressa a 4</td> </tr> </tbody> </table>		Actor Input	System Response	1 Não seleciona uma lista	Pede confirmação de voto nulo.	2		3	Regressa a 4				
Actor Input	System Response													
1 Não seleciona uma lista	Pede confirmação de voto nulo.													
2														
3	Regressa a 4													
<b>Alternativa 3</b> [Não confirma] (Passo 4)	<table border="1"> <thead> <tr> <th>Actor Input</th> <th>System Response</th> </tr> </thead> <tbody> <tr> <td>1 Não confirma</td> <td></td> </tr> <tr> <td>2</td> <td>Regressa a 1</td> </tr> </tbody> </table>		Actor Input	System Response	1 Não confirma		2	Regressa a 1						
Actor Input	System Response													
1 Não confirma														
2	Regressa a 1													

Figura 1.6: Diagrama de Use Case do Vota da Assembleia da República.

## 1.4 Proposta de Interface (Mockup) e Máquinas de Estado

Para este projeto, realizamos uma proposta de interface, utilizando a ferramenta *Evolus Pencil* que iremos apresentar de seguida.

### 1.4.1 Janela de login



Figura 1.7: Janela de login.

Esta é a janela proposta de login, janela esta que distingue automaticamente os Administradores do sistema dos restantes votantes, tendo cada um acesso a partes diferentes do programa.

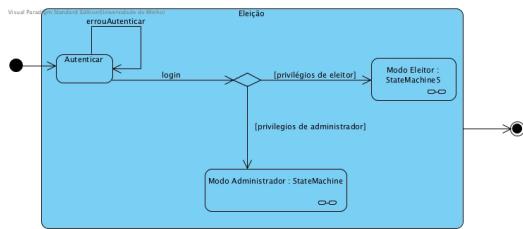


Figura 1.8: Maquina de Estado janela de login.

#### 1.4.2 Gestão de eleição (Administrador)

Após o login do Administrador, este é apresentado com uma janela em que é apresentada a eleição ativa (caso exista), que permite o término da mesma.

Numa segunda parte da mesma janela são apresentadas todas as eleições criadas e ativas bem como as eleições já terminadas.

Caso esteja no separador de eleições criadas é possível iniciar a eleição, criar uma nova eleição e gerir a mesma.

Uma outra parte da janela é relativa às eleições já realizadas, que permite a visualização dos resultados de uma aquando selecionada.

Em qualquer momento, nesta janela, o Administrador pode inserir uma nova versão do Caderno de Recenseamento.

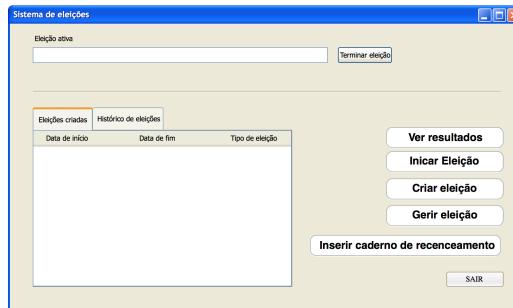


Figura 1.9: Janela principal do Administrador.

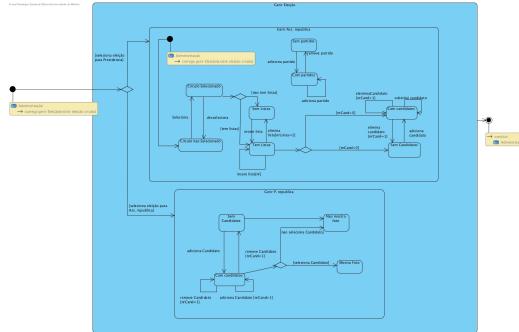


Figura 1.10: Maquina de Estado janela principal do Administrador.

#### 1.4.2.1 Criar Eleição

A janela de criação de eleição é a mesma para os dois tipos de eleição, o administrador uma vez nesta janela tem a possibilidade de escolher o tipo de eleição a criar (Assembleia da República ou Presidência da República), o Administrador terá também necessariamente inserir a data da eleição.

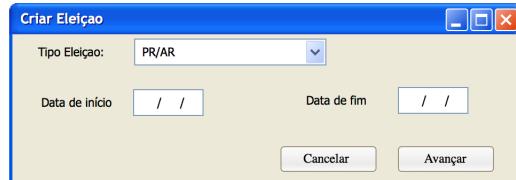


Figura 1.11: Janela de Criação de eleições.

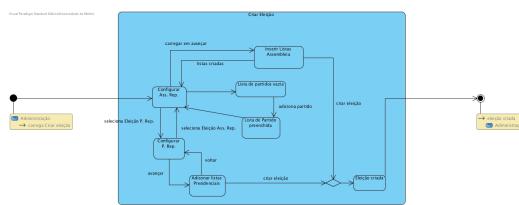


Figura 1.12: Maquina de estado da Janela de Criação de eleições.

#### 1.4.2.2 Gerir Eleições da Assembleia da República

A quando a seleção da gestão de uma eleição por parte do administrador este é direcionado para uma janela igual à apresentada onde o mesmo efetua

a todo o processo de gestão das eleições do tipo *Assembleia da República*.

Num primeiro passo o Administrador terá de selecionar o círculo que quer gerir, imediatamente é apresentado o número de mandatos que esse círculo tem direito e também uma lista com a sigla, nome e identificação dos partidos políticos de todas as listas existentes no círculo. Caso o Administrador tenha o desejo de adicionar uma nova lista este deverá inserir o respetivo nome, e sigla bem como selecionar todos os partidos pertencentes a essa lista, caso algum partido que se pretenda adicionar a uma lista não exista o utilizador numa parte superior da janela tem a possibilidade de o adicionar, deverá também inserir todos os candidatos que pertencem a essa lista, um a um ou por meio de ficheiro, só é possível inserir os candidatos restringido pelo número de mandatos do círculo em questão.

Num outro modo de operação esta janela permite eliminar um Partido quando não integrado em nenhuma lista, eliminar uma lista de um determinado círculo bem como fazer a substituição de um candidato numa lista de um círculo.

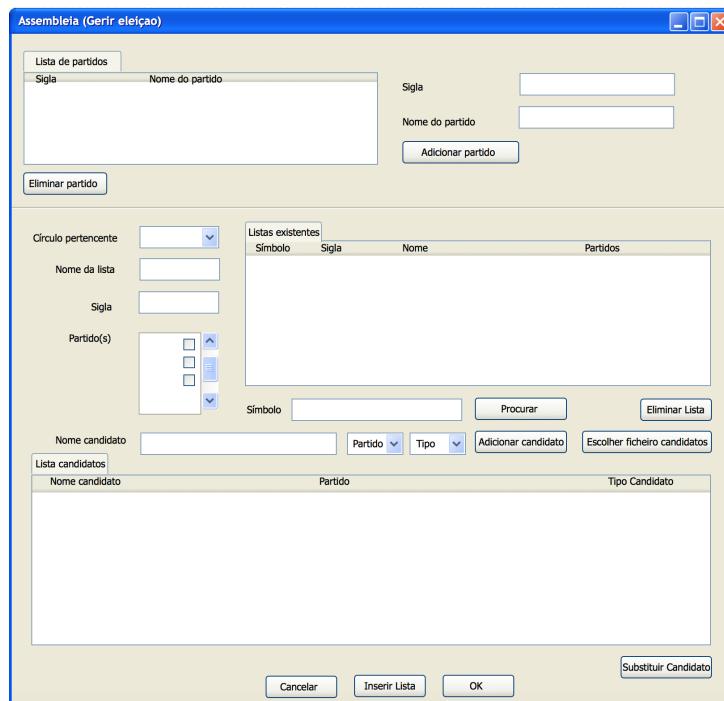


Figura 1.13: Gestão de eleições da Assembleia da República.

### 1.4.2.3 Gerir Eleições da Presidência da República

Para a gestão das eleições da presidência da república a janela apresentada como os candidatos são a nível nacional, não é necessário a gestão a nível de círculos. A janela permite adicionar um novo candidato inserindo o seu nome e uma foto para ser apresentada, caso o administrador pretenda remover um candidato também deverá ser feito nesta janela.

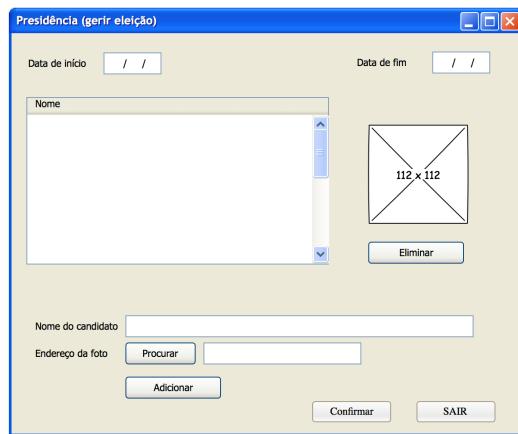


Figura 1.14: Gestão de eleições da Presidência da República.

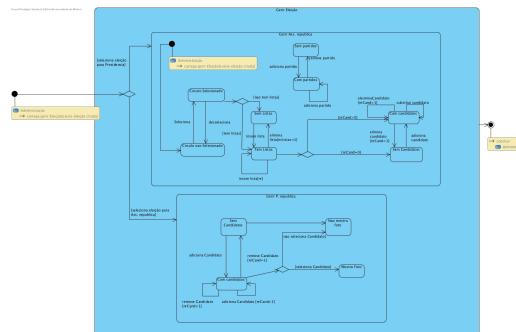


Figura 1.15: Maquina de estado das janela de Gestão de eleições AR e PR

#### **1.4.2.4 Inserção do Caderno de Recenseamento**

A Inserção do Caderno de Recenseamento é feita com base num ficheiro. Após a inserção é possível visualizar o numero de eleitores do círculo bem como o nome de cada um e o respetivo número de identificação.

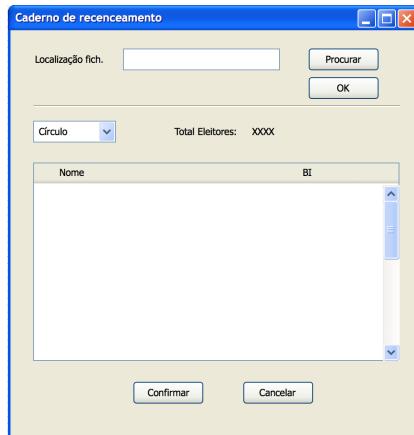


Figura 1.16: Inserção do Caderno de Recenseamento.

### 1.4.3 Janela Principal do eleitor

Se os dados de login estiverem corretos e forem referentes a um eleitor, aparecerá uma janela de eleitor, onde contém informações sobre o nome deste, o tipo de eleição que está a decorrer, a data de início e de fim, bem como o histórico de eleições já efetuadas. O utilizador tem a opção de votar para a eleição que está a decorrer, e também de analisar os resultados das eleições que já haviam decorrido.

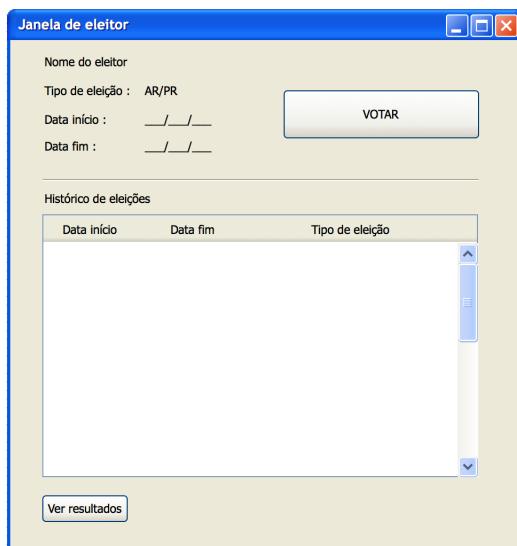


Figura 1.17: Janela Principal do eleitor

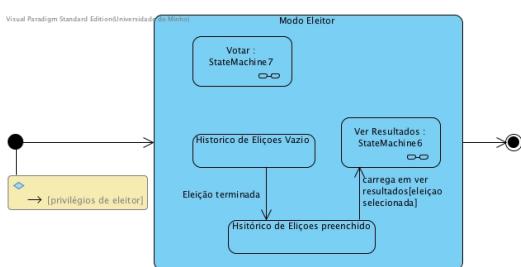


Figura 1.18: Maquina de estado do modo eleitor

#### 1.4.3.1 Boletim de voto

Caso o eleitor opte por votar, é-lhe apresentada uma janela que corresponde ao boletim de voto. Ao passar o cursor pelas opções, é apresentada à direita uma imagem representativa do partido. Esta janela contém uma lista de opções de votação, na qual cabe ao eleitor escolher uma delas. Caso o utilizador se engane, tem a opção de limpar a sua escolha.

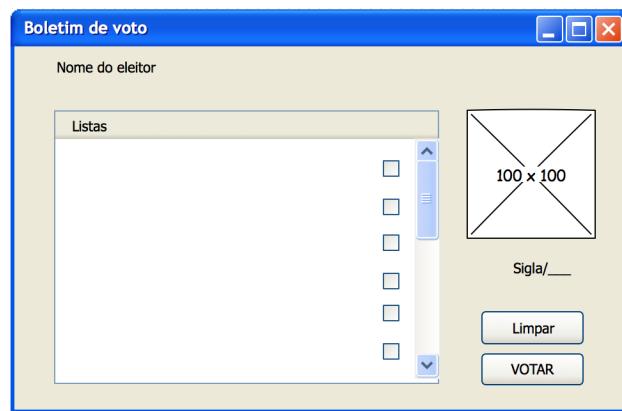


Figura 1.19: Boletim de Voto

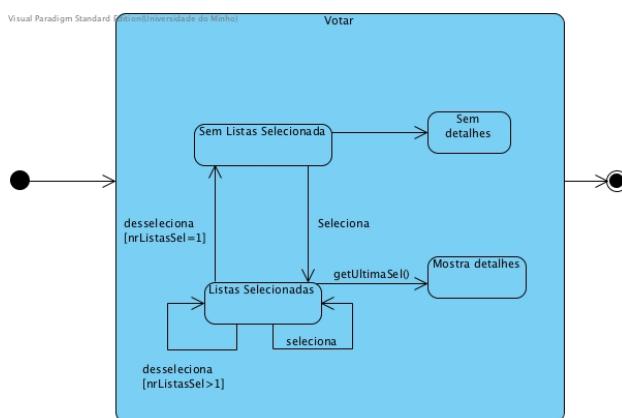


Figura 1.20: Maquina de estado da janela de Voto

#### 1.4.3.2 Resultados Assembleia República

Caso a eleição presente seja para a Assembleia da República, e o eleitor escolha a opção de ver resultados, é-lhe apresentado uma janela que contém os resultados globais das eleições anteriores (percentagens, lista, mandatos), bem como os resultados divididos por círculos, o que permite ao utilizador analisar detalhadamente e separadamente todas as informações que deseja.

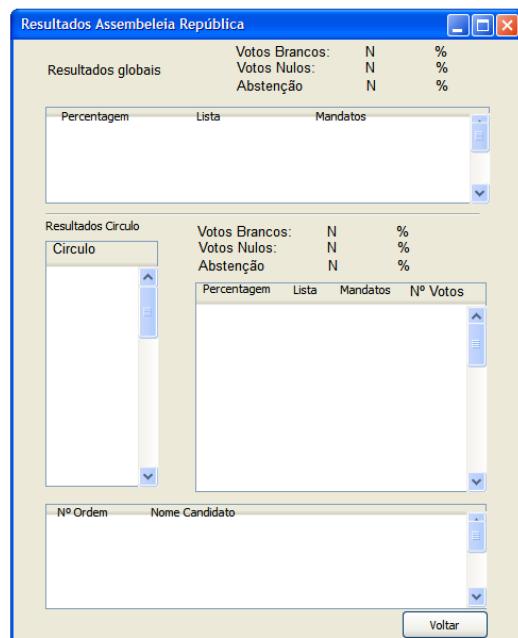


Figura 1.21: Resultados da Assembleia da República

### 1.4.3.3 Resultados Presidência República

Caso a eleição presente seja para a Presidência da República, e o eleitor escolha a opção de ver resultados, é-lhe apresentado uma janela idêntica à anterior, mas desta vez sem mandatos.

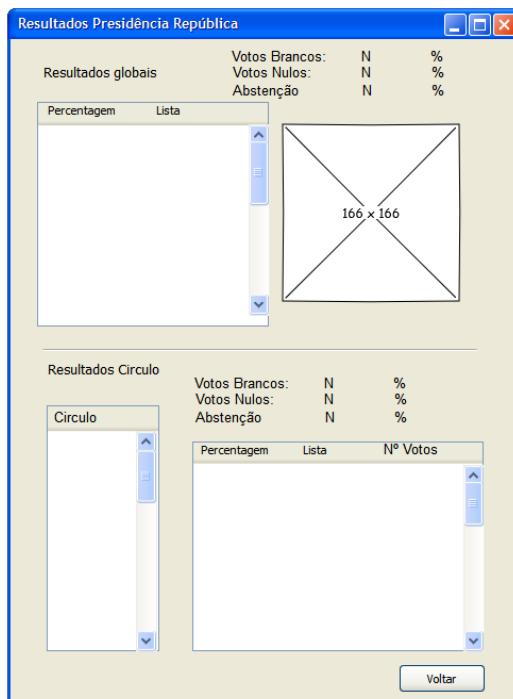


Figura 1.22: Resultados da Presidência da República

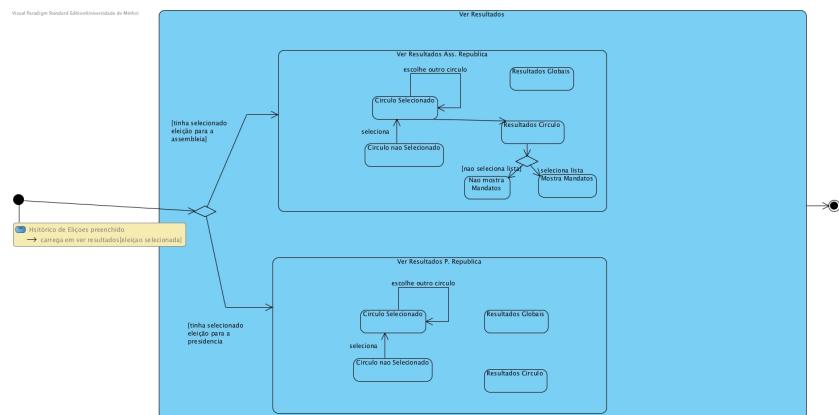


Figura 1.23: Maquina de estado da Janela Ver Resultados.

## 1.5 Conclusão 1<sup>a</sup> parte

Nesta primeira etapa do trabalho prático foram elaborados os Diagramas fundamentais à realização do mesmo. Entre os quais:

- Modelos de Domínio

Nestes modelos identificamos o que a nosso ver seria necessária para as funcionalidades que a nossa aplicação deverá implementar, pondo de parte questões mais burocráticas relacionadas com a administração legal das eleições.

Posto isto focamos maioritariamente na informação que o utilizador necessita aquando e após a realização do voto, ou seja, informação sobre as listas/candidatos, círculos e estatísticas eleitorais.

- Modelos de Use Case

Nestes modelos tentamos identificar as funcionalidades principais que a aplicação deveria ter para com o exterior. Sendo estas divididas em interações com o administrador e o eleitor comum.

- Diagramas de Sequência de Software

Aqui implementámos os use cases previamente definidos. Nesta fase já definimos um pouco a estrutura de classes da nossa aplicação, dividindo a lógica de negócio entre subsistemas. Entre os quais, estão os subsistemas de eleição, círculo (no caso da eleição para a Assembleia da República), lista, candidato (A.R.), e eleitor.

- Interface e Máquinas de Estado

Procurámos desenvolver uma proposta de interface intuitiva e de fácil utilização e navegação, que permitisse oferecer todas as funcionalidades implementadas na aplicação. Para definir o comportamento das janelas e a navegação entre eles, desenvolvemos modelos de máquinas de estado.

No decorrer do processo conceptual dos modelos utilizados sentimos dificuldade na junção do modelos elaborados pelos diferentes elementos do grupo pois a quando da sua junção grande parte da vezes o *Visual Paradigm* não deixava, o que levou a ser necessário fazer várias vezes os mesmos diagramas e consequente perda de tempo, precioso noutras partes do desenvolvimento.

# Capítulo 2

## Parte 2

### 2.1 Diagramas Visual Paradigm

#### 2.1.1 Diagramas de Estrutura

##### 2.1.1.1 Diagrama de Classes

De forma a seguir a metodologia abordada na disciplina, nesta fase, o grupo refinou o Modelo de domínio apresentados na primeira etapa. De modo a fazer com que este passe a conter linguagem mais próxima dos programadores e para que reflitam de forma total a arquitetura estrutural da aplicação contendo a representação de todas as classes consideradas para a solução.

A *Class* principal do da aplicação do Sistema de Gestão de Eleições é a *Class* SGE, esta *Class* é a *Facade* para todas as interfaces que desejem utilizar a aplicação a Gestão de Eleições devem “falar” com esta *Class*.

A *Class* SGE é a *Class* que sendo a principal trata de toda a gestão da lógica de negocio da aplicação controlando todas as restantes classes. Gere também a persistência de dados, para isso utiliza as classes do *Package* Data que estabelecem o *ORM* para um base de dados *MySQL*.

Esta *Class* de todos os métodos que implementa de modo a proporcionar a comunicação, destacam-se:

- *confirmarCadernoRecenciamento(Map<Integer, List<Eleitor> listaEleitores)*: este método é utilizado para quando quem utiliza o SGE confirma que deseja inserir um caderno de recenseamento que foi lido utilizando o método *lerCadernoRecenciamento(String path)*, este método faz com que toda a informação lida do caderno de recenseamento selecionado se torne persistente, ou seja é passada toda a informação ao *DAO* de eleitores que depois insere na base de dados a informação.

- *addVoto(Eleição e, Listável lista, Eleitor eleitor)*: Este método encarrega-se de adicionar um voto de um determinado eleitor, numa determinada lista, numa determinada eleição, sendo esta uma eleição da Assembleia da República ou da Presidência da República.
  - *initCírculos()*: Este método é essencial porque é onde se faz a povoação dos círculos. Para cada SGE o valor dos círculos vai ser igual, porque existe sempre a invocação de *initCírculos()*, que não recebe argumentos, logo os valores vão ser sempre os que foram implementados neste método. As implementações de algumas das restantes classes serão apresentadas posteriormente.

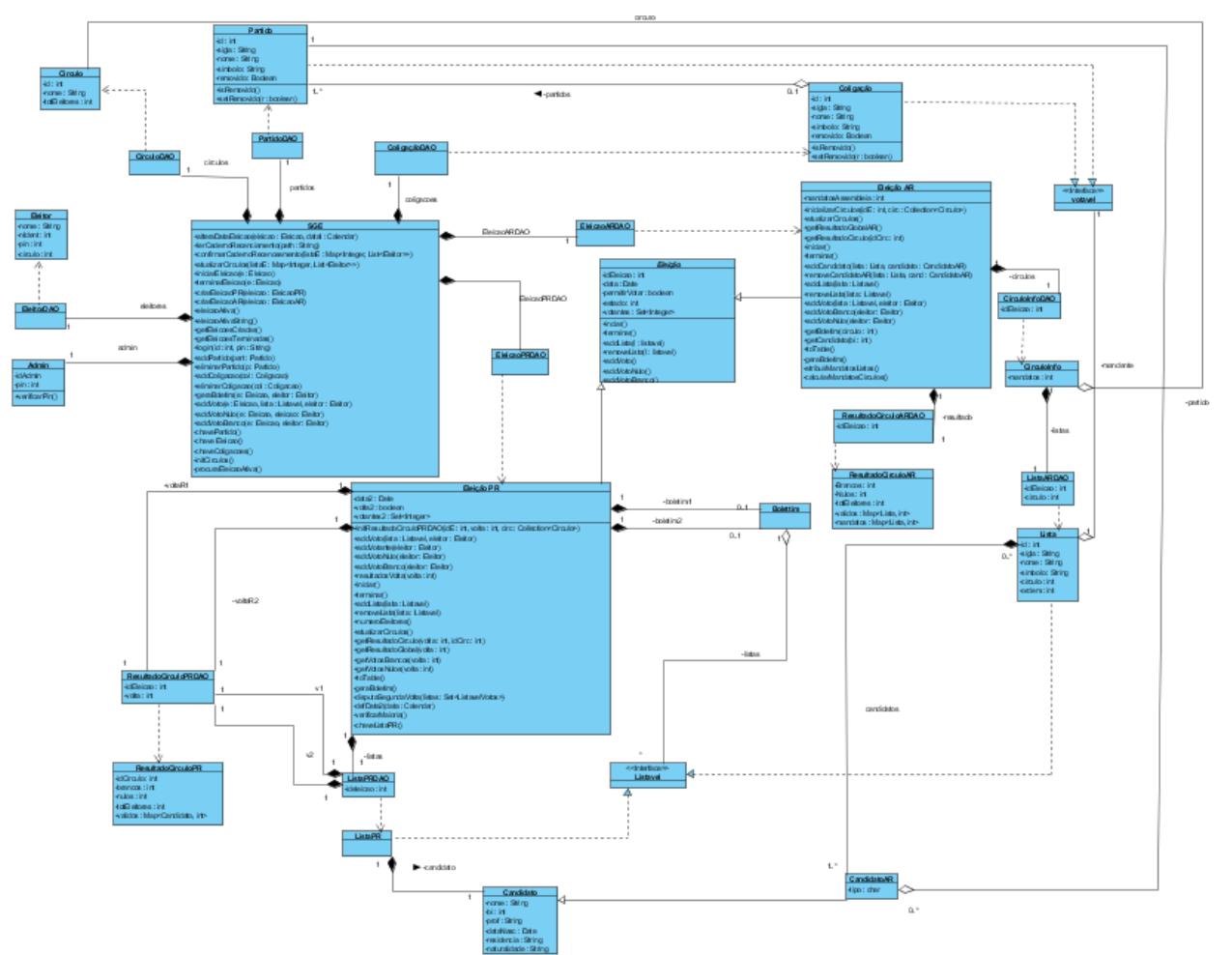


Figura 2.1: Diagrama de Classe

### 2.1.1.2 Diagrama de Packages

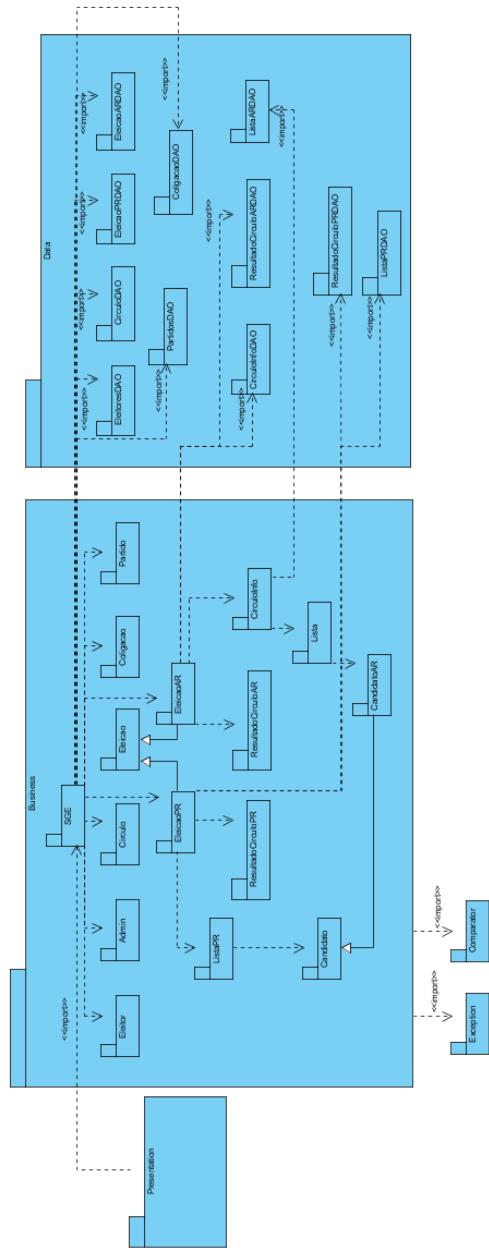


Figura 2.2: Diagrama de Packages

Os três principais pacotes definidos são *Presentation*, que trata da apresentação gráfica, em *Java Swing*; *Business*, que contém toda a lógica de negócio; e *Data*, responsável pela ligação à base de dados. A camada de apresentação comunica com a camada de lógica de negócio a partir da classe *SGE* (Sistema de Gestão de Eleições), que desempenha a função de *facade*. Não existe qualquer tipo de comunicação feita diretamente entre a camada de apresentação e a camada de dados. A comunicação entre o *package Business* e a persistência de Dados é feita recorrendo às várias classes *DAO* implementadas no *Package Data*, que comunicam com a base de dados *MySQL*.

### 2.1.1.3 Diagrama de Instalação

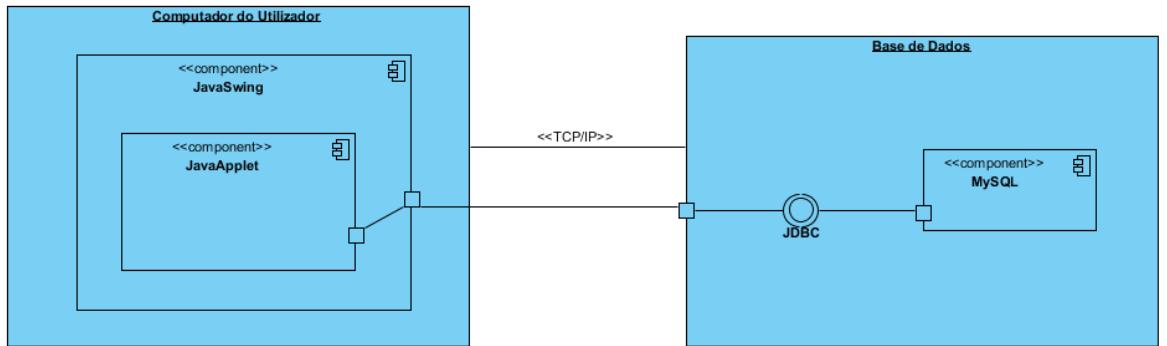


Figura 2.3: Diagrama de Instalação

Os dois grandes subsistemas da aplicação desenvolvida são o computador do utilizador e o servidor de base de dados. A comunicação é estabelecida por *TCP/IP*. O programa é uma aplicação Java, e, para a apresentação gráfica ao utilizador, é usado *JavaSwing*. Para ser possível a comunicação entre a aplicação em *Java* e a base de dados *MySQL*, recorre-se à interface *JDBC*<sup>1</sup>.

---

<sup>1</sup>Java Database Connectivity

## 2.1.2 Diagramas de Interação

### 2.1.2.1 Diagrama Sequencia Software

Devido à elevada extensão destes diagramas, não os vamos apresentar neste relatório, sendo possível consulta-los no projeto do *Visual Paradigm* que foi enviado juntamente com este relatório.

De forma resumida nesta etapa seguimos o roteiro apresentado nas aulas para a elaboração dos *DSS* utilizando para isso os diagramas apresentados anteriormente que continham a interação entre Subsistemas. Estes subsistemas deram lugar a *Classes* consideradas a quando do detalhe do modelo de domínio para diagrama de classes.

A interação entre as *classes* consideradas são também nesta etapa deram lugar a parte da *API* que cada classe deve implementar, ao contrário de simples frases representativas das funcionalidades pretendidas.

## 2.2 Código

### 2.2.1 Classes principais

O código que desenvolvemos para a criação da nossa aplicação foi desenvolvido em Java, orientada a objetos. Começamos por criar as classes principais, que achamos que seriam importantes definir.

- **Admin**

Contém informação relativa a um administrador do Sistema de gestão de eleições(o seu id e a password), úteis para fazer o login no sistema.

- **Boletim**

Contém informação de todas as listas que estão incluídas na eleição relativa ao boletim.

- **Candidato**

É a classe relativa a um candidato. A cada candidato diz respeito um número de identificação(bi), uma data de nascimento, um nome, uma profissão, uma residência, uma naturalidade e uma foto.

- **CandidatoAR**

CandidatoAR é uma subclasse de candidato, a diferença está no facto de este ter um Partido associado, um caractere que define o tipo de candidato (P:Primário,S:Secundário), e o número de ordem em que ele aparece na lista.

- **Círculo**

Cada círculo é composto por um id de Círculo, o nome da região e o número total de eleitores desse círculo em questão.

- **CírculoInfo**

A classe CírculoInfo contém um objeto do tipo círculo, e também informação das listas que pertencem àquele círculo, bem como o número de mandatos correspondentes.

- **Coligação**

Esta classe implementa uma interface votável, visto que faz parte dos objetos em que é permitido votar. Uma coligação é caracterizada por um elemento de identificação(id), uma sigla, um nome de coligação, um símbolo, a lista de partidos que fazem parte da coligação, e um elemento booleano, que nos diz se essa coligação foi removida do nosso Sistema ou não.

- **Eleicao**

Cada eleição é caracterizada por um id, uma data em que é iniciada, um booleano que indica se é permitido votar, um estado de eleição(criada, ativa ou terminada), e um set de todos os votantes que lhe são permitidos votar.

- **EleicaoAR**

EleicaoAR é uma subclasse de Eleição. São acrescentados o número de mandatos da Assembleia.

- **EleicaoPR**

EleicaoPR é também uma subclasse de Eleição, que contém um booleano que indica se a eleição que está a acontecer é referente à 1<sup>a</sup> ou 2<sup>a</sup> volta, a data de início, os resultados de eleição relativos a cada círculo, da volta 1 ou da volta 2, uma listagem das listas concorrentes à eleição, e uma listagem de todos os votantes da segunda volta.

- **Eleitor**

A cada eleitor diz respeito um nome, o id do Círculo a que pertence, o seu número de identificação e a password que será utilizada para fazer login no sistema.

- **Hondt**

É uma classe apenas com métodos de classe que servem para distribuir tanto o número de mandatos da Assembleia pelos respetivos círculos, como o número de mandatos de um determinado círculo pelas respetivas listas.

- **Lista**

No nosso sistema, uma lista é caracterizada pelo seu identificador de lista, o círculo a que pertence, o número de ordem que aparece no boletim, a sigla, o nome de lista, o símbolo da lista, o mandante em que é possível votar, e um ArrayList com os candidatos da lista da Assembleia da República.

- **ListaPR**

A classe ListaPR implementa, tal como a classe Lista, uma interface Listável, que faz com que estas listas possam ser listáveis. Uma ListaPR (Presidência da República) é caracterizada pelo seu identificador de listaPR, o id da eleição a decorrer, um booleano que traduz o número da volta da eleição, ordem 1 e 2 (ordem em que aparecem no Boletim(tanto na volta 1 como na volta 2)), e o candidato à Presidência da República.

- **ListavelVotos**

A classe ListavelVotos é utilizada para uma determinada lista armazenar o número de votos numa eleição.

- **Partido**

A classe Partido pertence também ao grupo de classes que implementa a interface Votável, e é caracterizada pelo seu id de Partido, a sigla, o nome de Partido, o símbolo, e um booleano a indicar se o Partido já foi removido ou não do sistema.

- **ResultadoCirculoAR**

ResultadoCirculoAR foi usada para representar os resultados referentes a um determinado círculo de uma eleição da Assembleia da República. Informação essa que se traduz no número total de eleitores, e nos votos atribuidos a esse círculo.

- **ResultadoCirculoPR**

ResultadoCirculoPR foi usada para representar os resultados referentes a um determinado círculo de uma eleição da Presidência da República. Informação essa que se traduz no número total de eleitores, e nos votos atribuidos nesse círculo.

- **ResultadoGlobalAR e ResultadoGlobalPR**

As classes ResultadoGlobalAR e ResultadoGlobalPR foram criadas de maneira a armazenar os resultados e dados relativos às eleições, mas desta vez de todos os círculos.

- **VotavelVotos**

Criamos a Class VotavelVotos de maneira a agrupar o número dos votos de um determinado Votável(Partido ou Coligação).

- **SGE**

SGE é a nossa classe principal, onde estão incluídos os métodos relevantes, que serão utilizados na nossa interface. Faz-se a declaração das classes principais, e dos DAOs(falaremos destes posteriormente).

## 2.2.2 DAOs

Para a realização deste projeto, construímos uma base de dados relacional em *MySQL* para armazenar toda a informação necessária para meter o Sistema de Eleições a correr. Para o nosso programa aceder a essa informação, nós criamos as classes *DAO*, que nos forneciam métodos de acesso à base de dados.

Poderíamos abordar todos os *DAOs* independentemente, mas a verdade é que são quase todos muito semelhantes. Cada método visível para o exterior destas classe solicita sempre um Concepção à base de Dados, este pedido é feito à classe *Connector*, podendo este conector estar configurado com *Auto-Commit* ou não dependendo do tipo de operações pretendidas.

São também definidos métodos auxiliares alguns *Private* e outros *Protected* que executam as *queries* à base de dados, contêm *PreparedStatement*s por questões de segurança contra *SQL injection*. A criação destes métodos deve-se à tentativa de minimizar as conceções simultâneas à Base de Dados.

### 2.2.2.1 Estratégias utilizadas para otimizar os DAOs e Base de Dados

Decidimos, que na base de dados, na tabela referente às listas candidatas a uma Eleição de assembleia da República continha o id de um partido e o id de uma coligação, devendo um deles ser *null*. Caso o *idPartido* seja diferente de *null*, o *idColigação* deverá ser *null*, e vice-versa, porque não existe uma lista que tenha um partido e uma coligação a concorrer à mesma eleição.

Sentimos também a necessidade de criar uma tabela para cada tipo de eleição resultando assim em dois *DAOs*, o *EleicaoDAOAR* e *EleicaoDAOPR*, esta divisão deve-se a facilitar a representação dos objetos na base de dados. Apesar desta divisão existe também uma tabela Eleição que contem os campos comuns aos dois tipos de eleição, cujo os dados são tratados pelos dois *DAO's* referidos.

Não sentimos necessário implementar certos *DAO's*, como por o que traria da comunicação dos dados dos Candidatos pois tais informações são transferidas de/para a base de dados noutras *DAO's*. Neste caso, a informação de um candidato é obtida quando se vai buscar informação à tabela das listas a que esse candidato pertence.

Todos os *DAO's* implementam a interface *Map<K,V>*<sup>2</sup>, como foi sugerido nas aulas, devido a inicialmente ser sugerido que a representação de dados fosse feita em *Map*, mas também pelo facto de o modelo relacional ser muito semelhante ao *Map*, sendo que para cada chave obtemos um dado

---

<sup>2</sup>*java.util.Map*

Valor.

Algumas entidades presentes na nossa abordagem, na base de dados teriam a necessidade de terem chaves compostas, como por exemplo o *ResultadoCirculoAR* que a sua chave seria o Id da eleição a que esses resultados pertence e o id do circulo em questão. Posto isto o *DAO* representativo desta classe tem como variável de instância o id da Eleição, sendo a sua chave quando referido apenas o Id do Circulo.

Como referido a cima de forma a reduzir o número de conexões existentes à base de dados, nos DAOs decidimos criar métodos auxiliares sendo alguns *protected*, de maneira a poderem ser utilizados por outras classes do mesmo *package*, por exemplo o *ListaPRDAO* utiliza métodos de *ResultadoCirculoPRDAO*, um outro exemplo é o método *values()* que utiliza o método *keySet()* e seguidamente múltiplas vezes o método *get()* com todos os valores retornados pelo no *Set*. Estes recebem como argumento uma conexão, e a partilham. Isto resulta numa diminuição significativa do número de conexões à base de Dados, impedindo falhas de sistema inconvenientes, também resulta numa melhor qualidade nas transações pois apenas é utilizada uma conexão e caso alguma operação que seja realizada utilizando esta conexão falhe todas as operações anteriores são desfeitas. Este método faz com que seja muito difícil levar a base de dados a um estado de inconsistência, permitindo assim maior fidelidade na aplicação.<sup>3</sup>

---

<sup>3</sup>\*Em anexo encontra-se o modelo lógico da base de dados utilizada.

### 2.2.3 Interface

Apesar da proposta de interface apresentada anteriormente e de tentarmos manter tais interfaces, durante o desenvolvimento do projeto apercebemos-nos de algumas impossibilidades tecnológicas relativas a alguns aspectos da janelas idealizadas.

Por outro lado foram efetuadas alterações a algumas janelas devido à necessidade ou não de introdução de certos dados. Também existiram alterações devido à elevada complexidade das interfaces apresentadas, um dos grandes problemas que nos deparamos na execução do projeto, houve necessidade de simplificar. O programa inicia com uma janela de login.

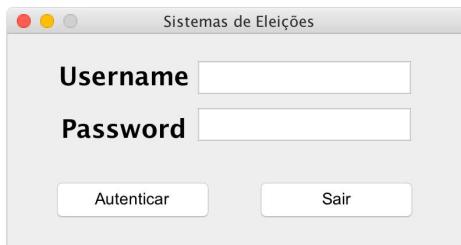


Figura 2.4: Login

Caso seja um eleitor a fazer o login, é-lhe apresentado a seguinte janela:

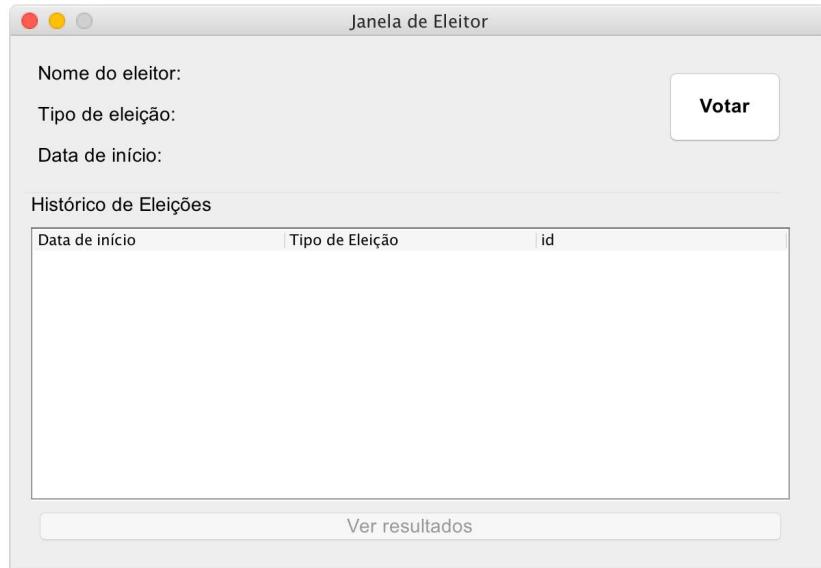


Figura 2.5: Janela de eleitor

Nesta janela é permito ao eleitor votar numa lista, ou ver os resultados tanto da Assembleia da República como da Presidência da República, de eleições que já decorreram.



Figura 2.6: Resultados da Assembleia da República



Figura 2.7: Resultados na Presidência da República

Caso o eleitor escolha votar numa lista, é-lhe apresentado o boletim de voto.

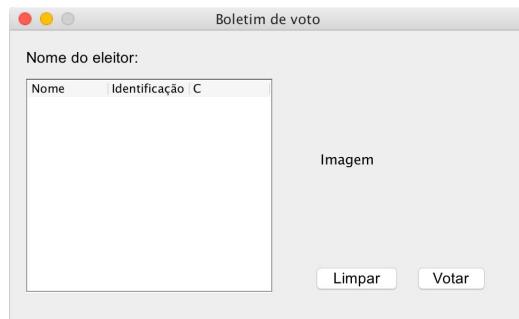


Figura 2.8: Boletim de voto

Se for um administrador a fazer o login, é-lhe apresentada a seguinte janela:

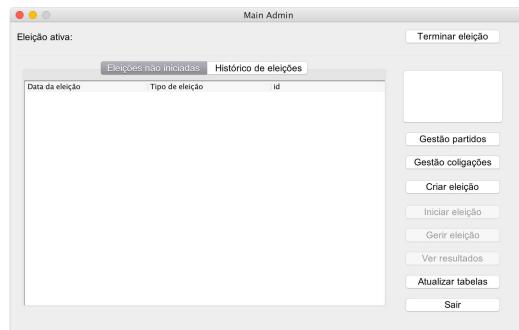


Figura 2.9: Main admin

O administrador pode optar por fazer a gestão de partidos ou coligações:

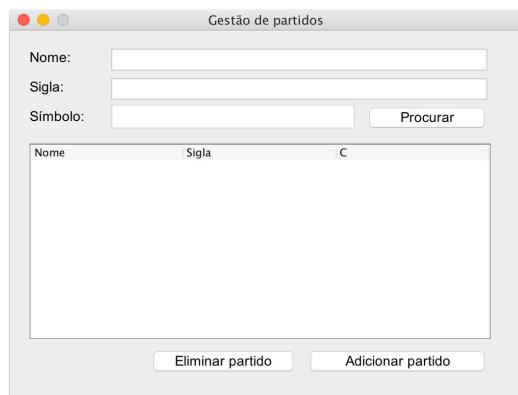


Figura 2.10: Gestão de partidos

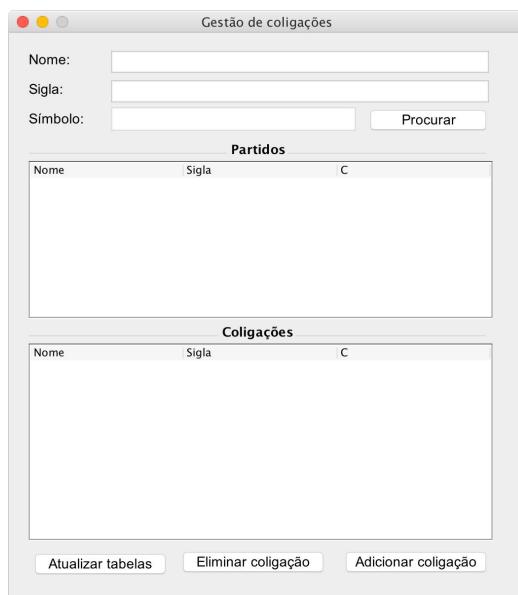


Figura 2.11: Gestão de coligações

Pode também fazer a inserção dos cadernos de recenseamento:

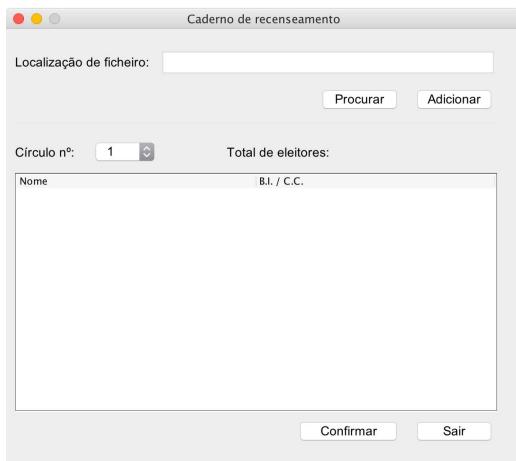


Figura 2.12: Inserção dos cadernos de recenseamento

O administrador pode criar uma nova eleição:

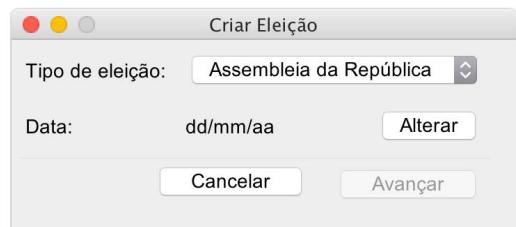


Figura 2.13: Criar eleição

Caso já tenha sido criada uma eleição, o administrador pode ter a opção de gerir tanto uma eleição da Assembleia como da Presidência da República:

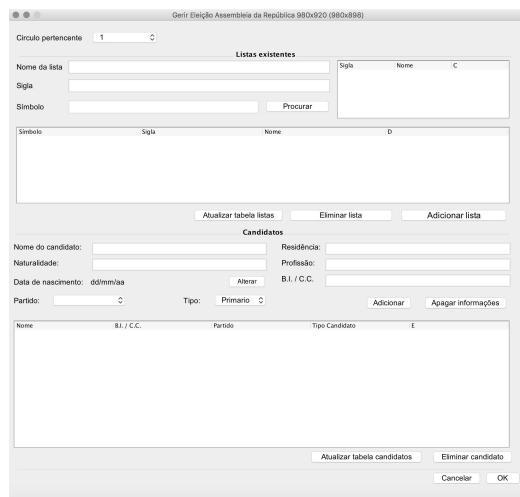


Figura 2.14: Gestão da Assembleia da República



Figura 2.15: Gestão da Presidência da República

## 2.3 Decisões Estruturais

Nesta secção abordaremos algumas decisões que tomamos na execução do projeto que ainda não haviam sido referidas nas secções anteriores.

Uma das preocupações que tivemos foi sempre que alguma alteração importante na aplicação ocorra, por exemplo um eleitor exerce o seu direito de voto, esta alteração no estado da aplicação seja refletida na persistência de dados. Isto ocorre para que não seja possível um eleitor votar mais que uma vez na mesma eleição. Esta decisão estrutural, numa futura iteração do modelo de desenvolvimento em espiral poderia ser otimizado para utilizar um sistema de *cache*, que efetuaria a gestão inteligente das escritas em persistência.

Inexistência da classe Boletim na base de dados. De modo a utilizar menos recursos de bases de dados optamos por não introduzir na Base de Dados um mapeamento da entidade boletim que consideramos. Para solucionar esta decisão introduzimos em todos as classes *Listáveis* pelo menos uma variável que representa o numero de ordem num boletim para que seja possível reconstruir um Boletim a qualquer momento de execução, incluindo na leitura dos dados das classes de persistência.

Os DAO não implementam todos os métodos referidos na interface *Map<K, V>*, com por exemplo o método

```
public void putAll(Map<? extends K, ? extends V> m);  
public Set<java.util.Map.Entry<Integer, Coligacao>> entrySet();
```

uma vez que a sua implementação seria mais custosa para o grupo e tais métodos nunca seriam utilizados, o grupo achou que não havia necessidade de os definir. Contudo perto do fim e com alguma dificuldades na gestão das conceções à base de dados o grupo tomou em consideração que numa próxima iteração de desenvolvimento devirá ser implementado o método *putALL()* para a class *EleitoresDAO* uma vez que os eleitores são as únicas entidade que são inseridas em massa na base de dados.

## 2.4 Conclusão parte 2

O nosso projeto está longe da perfeição devido à ambição da perfeição e ao elevado numero de funcionalidades que desejadas para a aplicação.

Caso existisse a possibilidade de começar o trabalho de novo, provavelmente simplificariamos muita coisa como por exemplo a exibição dos resultados de uma Eleição de presidencia da republica por círculo, quando não era estritamente necessário.

Relativamente à interface, uma vez que não temos elevados conhecimentos sobre o assunto, não se pode dizer que seja muito “user-friendly”, que seria um aspeto passível de sofrer alterações.

Inicialmente o grupo tinha como ideia de incluir na interface imagens dos candidatos e partidos ou coligações. No entanto, foram encontradas dificuldades devido ao facto de o o que ser guardado na base de dados ser a localização de uma imagem e não a imagem, o que leva a situações de erro sempre que a aplicação fosse executada em diferentes maquinas ou a imagem não estivesse disponível na mesma localização.

Contudo, pensamos que este projeto da maneira que o abordamos, ficou um projeto mais extenso que o suposto, o que levou a que não ficasse desenvolvido a 100% , as falhas que existem com mais algumas iterações do processo de desenvolvimento de software seguido poderiam ser colmatadas.

De forma a terminar, todos os elementos do grupo estão de acordo em que os vários modelos utilizados são ferramentas poderosas no processo de modelação e desenvolvimento de sistemas. Tais processos permitem modelar o comportamento e estrutura do sistema, podendo ser interpretados por praticamente qualquer pessoa com os mínimos conhecimentos técnicos de engenharia de software. Este facto é muito importante para quem opta pelo modelo de implementação em espiral, pois facilita o processo de validação do desenvolvimento.

## .1 Anexo

### .1.1 Modelo Lógico da Base de Dados

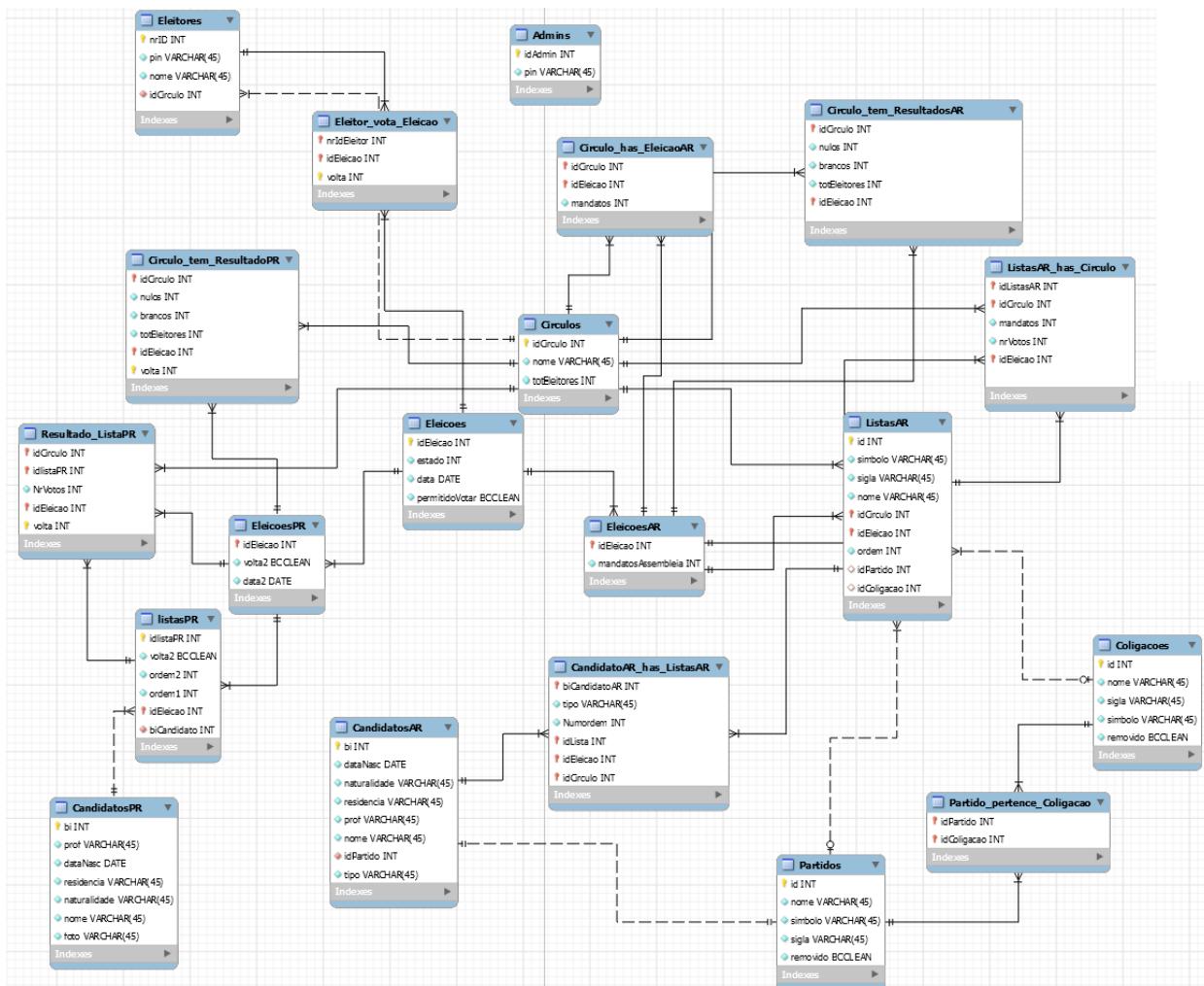


Figura 16: Modelo Lógico

# **Lista de Figuras**

1.1	Modelo de Domínio da Assembleia da República. . . . .	5
1.2	Modelo de Domínio da Presidência da Repúblca. . . . .	6
1.3	Diagrama de Use Case do Cria Eleição. . . . .	7
1.4	Diagrama de Sequência do Cria Eleição. . . . .	8
1.5	Diagrama de Sequência do Vota da Assembleia da Repúblca. . .	8
1.6	Diagrama de Use Case do Vota da Assembleia da Repúblca. .	9
1.7	Janela de login. . . . .	9
1.8	Maquina de Estado janela de login. . . . .	10
1.9	Janela principal do Administrador. . . . .	10
1.10	Maquina de Estado janela principal do Administrador. . . .	11
1.11	Janela de Criação de eleições. . . . .	11
1.12	Maquina de estado da Janela de Criação de eleições. . . . .	11
1.13	Gestão de eleições da Assembleia da Repúblca. . . . .	12
1.14	Gestão de eleições da Presidência da Repúblca. . . . .	13
1.15	Maquina de estado das janela de Gestão de eleições AR e PR .	13
1.16	Inserção do Caderno de Recenseamento. . . . .	14
1.17	Janela Principal do eleitor . . . . .	15
1.18	Maquina de estado do modo eleitor . . . . .	15
1.19	Boletim de Voto . . . . .	16
1.20	Maquina de estado da janela de Voto . . . . .	16
1.21	Resultados da Assembleia da Repúblca . . . . .	17
1.22	Resultados da Presidência da Repúblca . . . . .	18
1.23	Maquina de estado da Janela Ver Resultados. . . . .	19
2.1	Diagrama de Classe . . . . .	22
2.2	Diagrama de Packages . . . . .	23
2.3	Diagrama de Instalação . . . . .	24
2.4	Login . . . . .	31
2.5	Janela de eleitor . . . . .	31
2.6	Resultados da Assembleia da Repúblca . . . . .	32
2.7	Resultados na Presidência da Repúblca . . . . .	32
2.8	Boletim de voto . . . . .	33

2.9	Main admin . . . . .	33
2.10	Gestão de partidos . . . . .	34
2.11	Gestão de coligações . . . . .	34
2.12	Inserção dos cadernos de recenseamento . . . . .	35
2.13	Criar eleição . . . . .	35
2.14	Gestão da Assembleia da República . . . . .	36
2.15	Gestão da Presidência da República . . . . .	36
16	Modelo Lógico . . . . .	39

# Bibliografia

- [1] LEI ELEITORAL do PRESIDENTE DA REPÚBLICA Decreto-Lei nº 319-A/76, de 3 de maio, acedida em 18/09/2015.
- [2] LEI ELEITORAL da ASSEMBLEIA DA REPÚBLICA Lei nº 14/79, de 16 de maio, acedida em 18/09/2015.
- [3] José Creissac Campos, António Nestor Ribeiro, com contributos de F.M. Martins Desenvolvimento de Sistemas Software, MiEI – 3º ano / 1º semestre, 2015/2016,