Faculdade de Engenharia da Universidade do Porto
Licenciatura em Engenharia Electrotécnica e de Computadores

Universidade do Porto
FEUP Faculdade de Engenharia

Serviços Web de Votação Electrónica

Relatório do Projecto de Fim de Curso da LEEC 2005/06

José Manuel dos Santos Lopes **Bruno Miguel Fernandes Pereira**

Orientador na FEUP: João Isidro Vila Verde
Orientador no INESC Porto: Mário Jorge Leitão

Programa Operacional Ciência e Inovação 2010
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR

Porto, 3 de Julho de 2006

Faculdade de Engenharia da Universidade do Porto
Licenciatura em Engenharia Electrotécnica e de Computadores

Universidade do Porto
FEUP Faculdade de Engenharia

Serviços Web de Votação Electrónica

Relatório do Projecto de Fim de Curso da LEEC 2005/06

José Manuel dos Santos Lopes **Bruno Miguel Fernandes Pereira**

Orientador na FEUP: João Isidro Vila Verde
Orientador no INESC Porto: Mário Jorge Leitão

Ciência.Inovação 2010 Programa Operacional Ciência e Inovação 2010

Porto, 3 de Julho de 2006

Resumo

O projecto de "Serviços Web de Votação Electrónica" apresentado neste relatório, foi desenvolvido no âmbito da disciplina de PSTFC (Projecto, Seminário, ou Trabalho Final de Curso), durante o estágio no INESC-Porto, sendo realizado no DEEC (Departamento Engenharia Electrotécnica e Computadores), departamento da FEUP (Faculdade de Engenharia da Universidade do Porto). O projecto foi orientado pelo Prof. Isidro Vila Verde (orientador da FEUP) e pelo Prof. Mário Jorge Leitão (orientador do INESC-ID Porto).

O projecto teve como base a Tese de Mestrado do Prof. Ricardo Costa, "Plataforma Genérica de Votação Electrónica" (PGVE), que apresenta um sistema seguro de votação electrónica. O nosso objectivo principal era implementação da plataforma em serviços web. À medida que fomos analisando a Tese de Mestrado, verificamos que a plataforma necessitava de melhores meios de operacionalização:

- configuração semi-automática dos componentes da plataforma;
- integração de um sistema de validação de eleitores;
- disponibilidade para mais do que uma eleição (*Always ON*);
- integração com componentes de diferentes entidades;
- disponibilização de uma fase de teste da Eleição;
- disponibilização de uma fase de ambientação dos Eleitores à Eleição.

Para corresponder a estes requisitos foram criados dois componentes, sendo remodelados alguns módulos.

Assim foram desenvolvidos 5 serviços web em duas máquinas independentes, sendo possível ser acedidos a partir do Campus da FEUP. Também foram desenvolvidas 3 aplicações cliente que permitem interagir com os serviços web.

Abstract

The project "Serviços Web de Votação Electrónica" presented in this report, was developed in the scope of PSTFC (Projecto, Seminário, ou Trabalho Final de Curso) during the period of training in INESC-ID Porto,it was developed in DEEC (Departamento Engenharia Electrotécnica e Computadores) FEUP´s department (Faculdade de Engenharia da Universidade do Porto).

The project was oriented by Prof. Isidro Vila Verde (FEUP) and Prof. Mário Jorge Leitão (INESC-ID Porto).

This project is based in the Master's Thesis of Prof. Ricardo Costa,"Plataforma Genérica de Votação Electrónica" (PGVE), that presents a security e-voting system. The principal porpuse of this project was developmented a PGVE based in Web Services.

Looking at PGVE, we verify that needs better ways of funcionality:

- Semi-automatic configuration of the plataform components;
- Integration of validation system of voters;
- Disponibility for more than once election (System Always ON);
- integration with components of diferent entities;
- Offers a testing period of the election;
- offers a adaptation period for the voters.

To satisfize that requisits it was created two components and changed some modules.

So, it was developed five web services in two independents systems. This systems can be accessed from FEUP Campus. To permit access to the web services, it was developed three client applications.

Agradecimentos

Ao longo da realização deste trabalho beneficiamos do apoio de diversas pessoas e instituições a quem nós neste momento gostariamos de agradecer. Em particular, quero expressar o nosso reconhecimento:

- Ao nosso orientador de estágio da faculdade, o Prof. Isidro Vila Verde, pela sua disponibilidade para nos orientar e apoio dado durante o projecto.
- Ao nosso orientador de estágio da empresa (INESC-ID Porto), o Prof. Mário Jorge Leitão, pela possibilidade de estágio no INESC-ID Porto e integração num conjunto de projectos relacionados com Votação Electrónica.
- Ao Prof. Ademar Aguiar, pela possibilidade de inclusão do projecto no âmbito da disciplina de Engenharia do Software, ajudando a melhor documentar o projecto.
- Ao grupo de reunião de projectos relacionados com Votação Electrónica pela troca de ideias. Desse grupo, as pessoas intervenientes foram Prof. Mário Jorge Leitão, Prof. Isidro Vila Verde, Prof. Jorge Mamede, Prof. Ricardo Costa, Engª Mariana Costa e Joaquim.
- Ao Departamento de Engenharia Electrotécnica e de Computadores (DEEC), pelo local de trabalho e material disponibilizado, nomeadamente o Prof. Luís Corte Real e o técnico Carlos Graf.

Contents

List of Figures

List of Tables

Lista de Abreviaturas

API - Application Programming Interface

AS - Authorization System

BEEP - Blocks Extensible Exchange Protocol

BS - Ballot System

DEEC - Departamento de Engenharia Electrotécnica e de Computadores

ECR - Electoral Circumscription

ECM - Electoral Commission Member

ECRI - Electoral Circumscription Identifier

EDS - Electoral Definition System

EI - Election Identifier

FEUP - Faculdade de Engenharia da Universidade do Porto

FTP - File Transfer Protocol

HP - Hewlett-Packard

HTML - HyperText Markup Language

HTTP - HyperText Transfer Protocol

IP - Internet Protocol

JDBC - Java Database Connectivity

JSP - Java Server Pages

JVM -Java Virtual Machine

INESC - Instituto Nacional de Engenharia em Sistemas de Computadores do Porto

LAN - Local Area Network

LDAP - Lightweight Directory Acess Protocol

MAN - Metropolitan Area Network

MIME - Multipurpose Internet Mail Extensions

REST - Representational State Transfer

PGVE - Plataforma Genérica de Votação Electrónica

PKCS#7 - Public Key Crytography

REVS - Robust Electronic Voting System

RPC - Remote Procedure Call

SAML - Security Assertion Markup Language

SGBD - Sistema de Gestão de Base de Dados

SGML - Standard Generalized Markup Language

SMTP - Simple Mail Transfer Protocol

SOAP - Simple Object Access Protocol

SQL - Structured Query Language

TCP - Transmission Control Protocol

TS - Trust System

UDDI - Universal Description, Discovery and Integration

VC - Vote Collector

WAN - Wide Area Network

WSDD - Web Service Deployment Descriptor

WSDL - Web Services Description Language

WSS4J - Web Services Security 4 Java

XML -eXtensible Markup Language

Chapter 1

Introdução

1.1 Votação Electrónica

Numa democracia, um direito politico muito importante é a posibilidade de votar. Sendo hoje em dia esse direito utilizado não só em politica como torna-se comum em muitas áreas: Inquéritos, Sondagens, Referendos, Votações em programas televisivos (exemplo: *Reality Shows*), Eleições de cargos (exemplo: eleição do reitor da Universidade do Porto) .

A Internet tem conhecido nos últimos anos uma forte expansão, sendo cada vez menos as pessoas que ainda não tiveram contacto. Assim surge a ideia de votação pela Internet (Votação Electrónica), trazendo vantagens e desvantagens em relação ao método tradicional.

Como a Internet é um ambiente hostil, traz a desvantagem de existirem problemas nas comunicações, sendo necessário mecanismos para a recuperar de erros provocados pela comunicação. E ainda está sujeita a vírus informáticos, *hackers*, *spyware*, entre outros factores, tornando-se necessário uma plataforma de voto segura. Uma vantagem que Votação Electrónica traz é a mobilidade, sendo possível o voto em qualquer lado, desde que o eleitor conecte-se à Internet e aceda ao serviço. Contribuindo para a diminuição da abstenção. Outra vantagem é reduzir os custos de uma votação, nomeadamente em papel e alugar espaços para realização da votação.

1.2 Estado da Arte

A Estónia e Suíça são exemplos de países em que se utiliza Votação Electrónica em eleições. Para além destes exemplos, existem sites na Internet onde são fornecidos serviços de votação electrónica, ficam aqui alguns serviços:

- iBallot [3]
- Safevote [4]
- SCYTL - Secure Electronic Voting [5]
- VoteHere [6]
- EveryCounts [7]

Em Portugal, já se tem elaborados vários projectos em diversas entidades, sendo feitos alguns testes piloto. As conclusões a que se chegam é que existe *a necessidade de realizar mais experiências de voto electrónico não vinculativo, sendo essencial ter como objectivo garantir a segurança, a transparência e a acessibilidade do processo para a participação de todos os eleitores*[1].

Em 29 Março de 2006 realizou-se um *workshop* em Lisboa, no INESC-ID, em que foram apresentados modelos de votação electrónica. Dentro dos quais *Robust Eletronic Voting System* (REVS), e algumas evoluções, assim como o modelo *Plataforma Genérica de Votação Electrónica* (PGVE), assim como o nosso modelo aqui apresentado (evolução da PGVE).

1.3 Introdução à *Plataforma Genérica de Votação Electrónica*

A Plataforma Genérica de Votação Electrónica (PGVE) [8] é um sistema distribuído com diferentes componentes interligados por um rede, seja esta uma LAN, MAN ou WAN como apresentado na figura 1.1. A PGVE satisfaz os requisitos

gerais de uma eleição: exactidão, democracia, privacidade, verificação, mobilidade, auditabilidade (estes requisitos serão detalhados na secção 2.1).

Neste sistema de votação, o eleitor acede à PGVE através da rede, embora a sua presença física seja opcional, interagindo com os diversos componentes da plataforma.

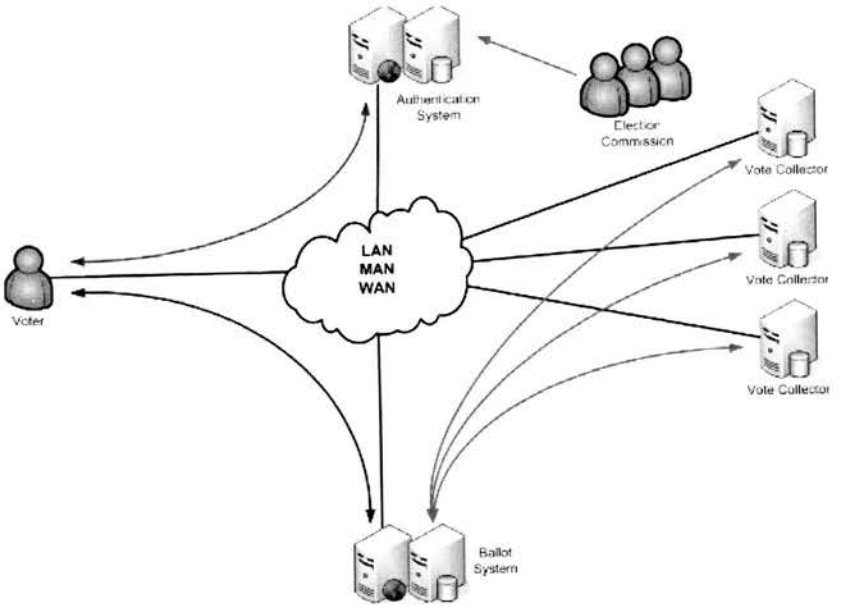


Figure 1.1: Cenário Plataforma Genérica de Votação Electrónica

Os componentes da PGVE são os seguintes:

- O "*Authorization System*" (AS) é responsável pela autenticação do eleitor durante a eleição e por lhe entregar a credencial de votação anónima e o boletim de voto.
- O "*Ballot System*" (BS) é responsável por receber os votos cifrados, validar as respectivas credenciais, verificar que não foram usadas previamente e distribuir os votos validados pelos vários "*Vote Collectors*" (VC).
- O(s) "*Vote Collector(s)*" aceitam os votos provenientes apenas do BS, guardando-nos aleatoriamente e permitindo a sua contagem após finalizada a eleição. Neste mesmo momento os votos podem ser tornados públicos para recontagem.

A opção da utilização de múltiplos VC é uma característica essencial desse sistema, para garantir uma redundância geográfica para os votos já recolhidos.

1.3.1 Questões de Operacionalização

A PGVE, embora seja uma plataforma robusta e cumpre os requisitos gerais de uma eleição, tem certos problemas de operacionalização, em que para cada eleição é necessário:

- Eventualmente instalar hardware e software. Embora seja possível ter estas máquinas já preparadas e prontas a usar, pode ser necessário instalá-las numa nova infra-estrutura, sendo necessário a mudança de IP's, configuração de *firewalls*, entre outros procedimentos de configuração.
- Adaptar o AS a cada sistema particular de validação de eleitores. Cada entidade que organiza umas eleições (comissão eleitoral) pode e terá certamente sistemas distintos de autenticação (LDAP, base de dados, sistemas de contas Unix, ficheiro de utilizador/password, etc). Em cada caso o AS terá de ser adaptado a essa eleição.
- Criar e distribuir as chaves. O processo de criação e distribuição de chaves obriga a procedimentos humanos passiveis de erros.
- Controlar o período de votação. O controlo do período de tempo durante o qual é possível o acesso não é definido na PGVE (embora seja possível), mas deve ser controlado por procedimentos manuais ou semi-automáticos a nível de *firewalls* ou então a nível de configurações adicionais nas aplicações do AS, BS e VC.

1.3.2 Solução proposta

Para solucionarmos as questões de operacionalização propomos a disponibilização dos sistemas por serviços web com serviços de auto-configuração dos serviços web envolvidos na eleição, assim como um simples serviço de confiança (em que uma entidade disponibiliza para autenticar os utilizadores). Esta solução para além de colmatar os problemas de operacionalização da PGVE, possibilita a integração em futuras aplicações de votação electrónica.

1.4 Organização e Temas Abordados no Presente Relatório

Análise do Problema

Neste capítulo iremos analisar o problema que foram encontrados na PGVE e a ideia do nosso projecto. Para finalizar este capítulo serão apresentados os casos de utilização para cada sistema (AS, BS e VC) e para as aplicações (aplicação de voto e aplicação de contagem), os novos sistemas e aplicação criados serão apresentados com os seus respectivos diagramas de utilização na no capítulo Arquitectura.

Este capítulo

Tecnologias

Neste capitulo será descrito as tecnologias usadas, assim como a eventual razão para a sua utilização.

Arquitectura

No capítulo Arquitectura inicialmente serão apresentados os novos sistemas criados. Neste capítulo tambem serão apresentadas a arquitectura lógica e física do projecto bem como a classes de dominio.

Funcionamento

Neste capítulo serão apresentados inicialmente os vários estados de uma eleição(na configuração desta e nas fases dos testes e votação oficial). Na secção funcionamento iremos dar enface aos vários passos que acontecem no acto da votaçaõ.

Resultados

Aqui apresentamos todos os passos necessários para chegar aos resultados, assim como os resultados. Será apresentado a instalação do software necessário, dos serviços web e depois é apresentado as aplicações cliente.

Conclusões

Neste capítulo serão apresentadas as várias conclusões a que chegamos depois da implementação do nosso sistema de votação.

Perspectivas de trabalho futuro

As ideias que foram aparecendo a medida que a execução do projecto ia avançando e que não foi possivel implementar-las (por motivos de falta de tempo ou por escolha de outras tecnologias mais estaveis na altura de algumas decisões), foram registradas neste capítulo.

Chapter 2

Análise do Problema

A Plataforma Genérica de Votação Electrónica (PGVE) na qual este projecto foi baseado, possuí algumas questões de operacionalização, as quais já foram anteriormente apresentadas na secção 1.3.

Neste capítulo serão apresentados os requisitos gerais de uma eleição na secção 2.1. Para finalizar mostraremos a nossa ideia de projecto(ver secção2.2)que encontramos para responder as questões de operacionalização que a PGVE possuí bem como a apresentação dos vários sistemas desta plataforma (PGVE).

Os vários sistemas da PGVE foram agrupados nos Módulos de Votação (ver secção dos sistemas 2.3). Os sistemas que a PGVE possuí são:

- Authorization System
- Ballot System
- Trust System

Estes sistemas entram directamente no funcionamento(ver secção 4.2 onde o funcionamento é apresentado mais pormenorizadamente).

A autenticação do eleitor, no nosso sistema é feita de forma diferente em relação a PGVE, pois para fazer a autenticação criamos um novo sistema que será apresentado na secção3.9).Os passos próximos do eleitor são praticamente os mesmos descritos na PGVE. Em que o eleitor se identifica no AS e obtém a credencial de voto. Neste passo o AS só devolve a credencial de votante se o potencial eleitor tiver uma credencial que o qualifica como um eleitor (*credencial de eleitor*). Depois o eleitor, através da aplicação de votação, preencheria o boletim de voto e submetia ao BS juntamente com a credencial de votante. Depois o BS depositaria os votos nos VCs.

Após a votação procederia-se à contagem dos votos. Neste último passo utilizamos como intermediário um outro sistema (ver secção 3.8) onde a aplicação de contagem obteria os endereços do VCs e a chave da Eleição que permitiria abrir os votos. Depois solicita os votos a um VC para fazer a contagem.

Esta aplicação é executada pela comissão eleitoral que tem que apresentar os resultados. Mas pode estar acessível a todos para verificação da contagem.

2.1 Requisitos Gerais

2.1.1 Precisão

Não deve ser possível alterar ou eliminar um voto válido, bem como contar um voto inválido. A contagem dos votos deve ser reprodutível em sucessivas contagens dos votos. Na realidade pode acontecer erros humanos na contagem dos votos mas num sistema electrónico o erro pode acontecer inúmeras vezes visto que pode ser um erro sistemático. Para eliminar estes erros sistemáticos seria vantajoso utilizar diferentes tipos de aplicações de contagem para eliminar o erro sistemático.

2.1.2 Democracia

Cada eleitor válido tem o direito de depositar um e apenas um voto válido. Dando assim o poder à eleitores válidos de tomar decisões.

2.1.3 Privacidade

Nem o eleitor, nem mais ninguém, consegue provar a sua orientação de voto. O eleitor é o único que pode saber a sua estratégia de voto/opção. Cada eleitor para cumprir o seu direito de voto deve depositar um boletim especificando a sua opção. Na prática um eleitor renega o seu direito de votar não depositando o voto.

2.1.4 Verificabilidade

Deve ser possível recontar os votos de forma independente.

2.1.5 Mobilidade

Um eleitor deve ser capaz de votar independentemente da sua localização. Logo não deve ser imposto ao eleitor qualquer restrições quanto à sua localização.

2.1.6 Auditabilidade

O sistema de cotação deve ser validado por observadores externos. Qualquer sistema de hardware ou software bem como toda a documentação, dever ser disponibilizado para inspecções aleatórias mesmo que os fornecedores se oponham. Um sistema *open source* seria o ideal pelas características que possuí.

2.2 Apresentação *Serviços Web de Votação Electrónica*

A ideia que trazemos neste trabalho é disponibilizar os sistemas por serviços web (*Web Services*):

- Para possibilitar integrações em aplicações (futuras) de votação electrónica.
- Para possibilitar o rápido desenvolvimento de aplicações de e-vote.
- Sendo os sistemas orientados para web.
- Possibilitando a existência de vários sistemas, que permite a distribuição de círculos eleitorais por vários sistemas.
- Tornando o seu funcionamento visível aos utilizadores, sendo também mais fácil a configuração dos sistemas.

2.3 Sistemas

A Plataforma Genérica de Votação Electrónica apresenta na sua arquitectura três sistemas: *Authorization System, Ballot System* e o *Vote Collector*. Estes sistemas fazem parte dos Módulos de Votação que serão apresentados a continuação.

2.3.1 Módulos de Votação

Authorization System

Este sistema cria credenciais de votação para os votantes que estejam devidamente identificados. É responsável por fazer com que a credencial de votação seja anónima para garantir a privacidade do eleitor. Para além de ter a função de criar credenciais de votação é responsável por fazer também a entrega do boletim de voto.

- Obter Chave Pública -Um actor obtém a chave pública do par de chaves do sistema.
- Obter Credencial de Votação -Um *Voter* obtém um credencial de votação única, que certefica o Voter como um votante para um Circulo Eleitoral.
- Definir Configurações -O EDS configura o sistema para um Circulo Eleitoral.
- Visualizar Configurações -O EDS obtém as configurações do sistema

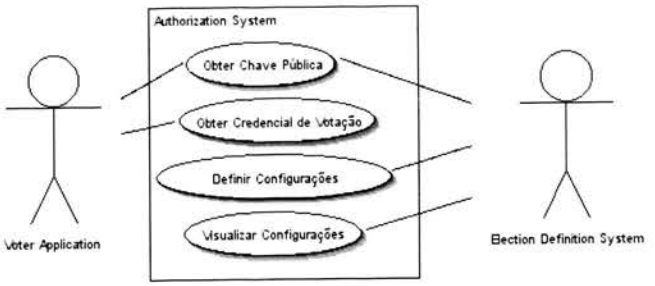


Figure 2.1: Diagrama dos modelos de casos de utilização para o AS

Ballot System

O *Ballot System* recebe o voto do votante cifrado e verifica as respectivas credenciais, verificando se estas já não foram usadas previamente. Se os votos forem validos são reencaminhados para os vários VCs.

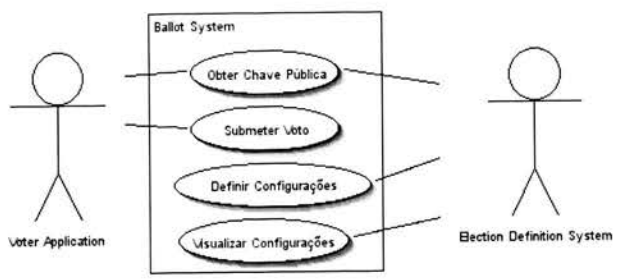


Figure 2.2: Diagrama dos modelos de casos de utilização para o BS

- Obter Chave Pública -Um actor obtém a chave pública do par de chaves do sistema.
- Submeter Voto -O *Vote Application* submete o voto ao sistema.
- Definir Configurações -O EDS configura o sistema para um Circulo Eleitoral.
- Visualizar Configurações -O EDS obtém as configurações do sistema

Vote Collector

Este sistema tem como funcionalidade armazenar os votos que lhe são enviados apenas pelo *Ballot System.* Depois de a eleição ter finalizado e da chave da eleição ter sido fornecida, podes-se efectuar a contagem.

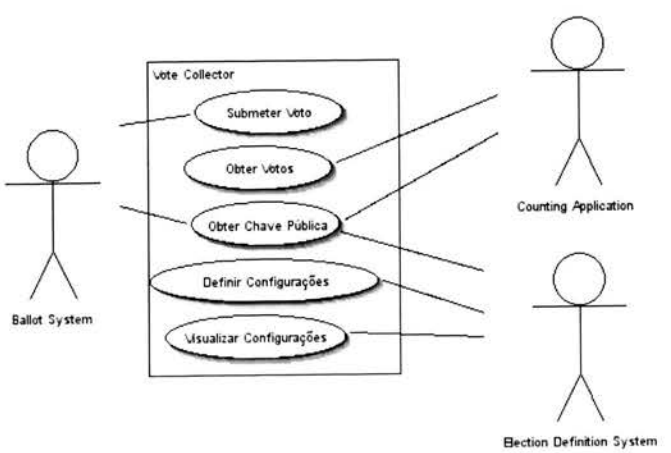


Figure 2.3: Diagrama dos modelos de casos de utilização para o VC

- Submeter Voto -O BS submete o voto ao sistema.

- Obter Votos -O *Counting Application* obtém votos de um ECR.
- Obter Chave Pública -Um actor obtém a chave pública do par de chaves do sistema.
- Definir Configurações -O EDS configura o sistema para um Circulo Eleitoral.
- Visualizar Configurações -O EDS obtém as configurações do sistema

2.3.2 Módulos de Aplicação

A Plataforma Genérica de Votação Electrónica tem duas aplicações. Estas aplicações são:

- Aplicação de voto -É a interface entre o eleitor e os vários sistemas da PGVE.
- Aplicação de Contagem -Estas aplicação serve para contar os votos previamente recolhidos do *Vote Collector.*

Para além das aplicações usadas pela PGVE usamos uma aplicação extra para ser utilizada pela comissão eleitoral.

Aplicação de Voto

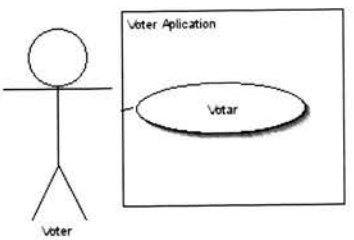


Figure 2.4: Diagrama dos modelos de casos de utilização para à aplicação de voto

- Votar -O votante vota.

Aplicação de Contagem

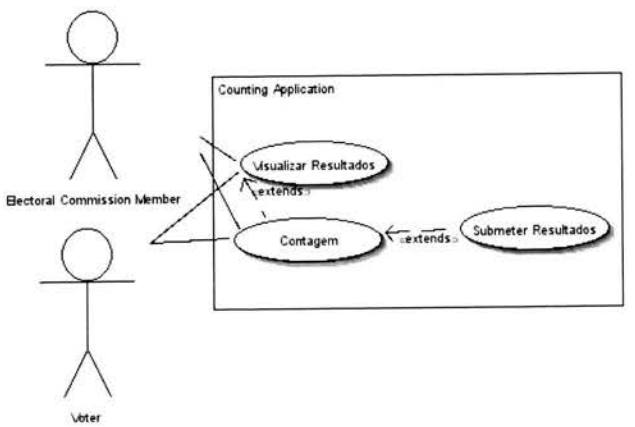


Figure 2.5: Diagrama dos modelos de casos de utilização para à aplicação de contagem

- Visualizar Resultados -O *Electoral Commission Member* ou o *Voter* visualizam os resultados da eleição pretendida.
- Contagem -O *Electoral Commission Member* ou o *Voter* podem fazer a sua propria contagem dos votos.
- Submeter Resultados -O *Electoral Commission Member* submete os resultados da eleição.

2.3.3 Actores

Nesta secção iremos abrangir os vários actores que a PGVE possuí. Para além dos actores que serão apresentados a seguir o nosso sistema tem um actor extra. Os motivos para ter-mos mais um actor em relação a PGVE serão explicados na secção **??**.

Votante

Pessoa que pretende votar, interagindo com a Aplicação de Voto.

Membro da Comissão Eleitoral

Pessoa, membro da comissão eleitoral (entidade promotora da eleição), que cria e configura uma eleição.

Aplicação de Voto

Esta aplicação serve de intermediário entre os serviços web e o votante.

Aplicação da Comissão Eleitoral

Esta aplicação serve de intermediário entre os serviços web e o um membro da comissão eleitoral.

Aplicação de Contagem

Esta aplicação serve de intermediário entre os serviços web e alguém que pretenda fazer a contagem dos votos de uma determinada eleição.

Ballot System

O *Ballot System* é o único que pode submeter votos no *Vote Collector*, logo funciona como um actor.

Chapter 3

Tecnologias

3.1 Sistema Operativo

O Sistema Operativo utilizado para implementação e desenvolvimento foi a distribuição de Linux: Gentoo Linux.

O Linux é um sistema operativo, derivado do UNIX e compatível com o standard POSIX. Tecnicamente, Linux é o kernel (o núcleo) do sistema operativo.

As vantagens pela qual escolhemos usar Linux face a outros sistemas operactivos foram:

- Robustez e fiabilidade. O seu código é analizado por milhares de programadores, e as correções de vulnerabilidades disponíveis num curto espaço de tempo. É modular, sendo apenas necessário reiniciar o sistema para trocar o kernel ou modificar hardware.
- Segurança. Códigos escritos 100% abertamente tem a segurança redobrada, pois quando o código é lido, mesmo asssim não é possivel invadi-lo.
- Rapidez. O protocolo TCP/IP do Linux foi reescrito do zero, usando novas técnicas de conexão e mais segurança, assim o protocolo do linux sendo 30% mais rápido do que o Windows NT/2000.
- Estabilidade. A estabilidade dos sistemas Unix já é conhecida mundialmente, por esta razão empresas como Google, Yahoo, HP estão migrando para Linux.

A distribuição Gentoo Linux permite uma melhor perfomanece face às outras distribuições. Para além disso possui suporte a várias aplicações, utilizando a aplicação portage como gestor de aplicações.

Para instalar o Gentoo Linux basta seguir o handbook disponível em [9].

3.2 Linguagem Programação

A linguagem de programação usada para desenvolver este rpojecto foi Java, para assim satisfazer os requisitos de portabilidade e segurança (ver secção), e devido a existirem variadas bibliotecas relativas ao serviços web (como por exemplo: axis) e segurança.

A linguagem de programação Java foi inicialmente lançada em meados de 1990, tendo sido desenvolvida por uma equipa de engenheiros da *SUN Microsystems*.

Neste momento a tecnologia Java é simultâneamente uma linguagem de programação e uma plataforma.

Linguagem de Programação

A linguagem Java é uma linguagem de alto-nível que tem as seguintes características:

- Programação orientada a objectos, o que torna a programação muito simples;
- O ciclo de desenvolvimento é mais rápido, pois o Java é uma linguagem interpretada. Assim basta compilar e executar.
- As aplicações são portáveis sobre diversas plataformas. Possibilita a escrita de codigo e compilação de código numa plataforma e execução noutra.

- As aplicações são robustas, porque é a *Java Runtime Environment* que maneja a memória.
- As aplicações gráficas interactivas tem boa performance devido ao Java suporte *multithreading*.
- As aplicações são seguras, mesmo fazendo *downloads* por toda a Internet, porque a *Java Runtime Environment* tem protecções contra vírus e modificação da memória.

Na linguagem de programação Java, todo código fonte primeiramente é escrito em texto em ficheiros com a extensão .java. Estes ficheiros são posteriormente compilados em ficheiros .class pelo compilador *javac*. Um ficheiro do tipo .class não contém código binário para o processador, mas sim *bytecodes* - a linguagem maquina da *Java Virtual Machine*, sob a qual a aplicação executa, mediante o uso da aplicação *java* (este aspecto constata que a tecnologia java é uma plataforma, ver melhor na secção 3.2).

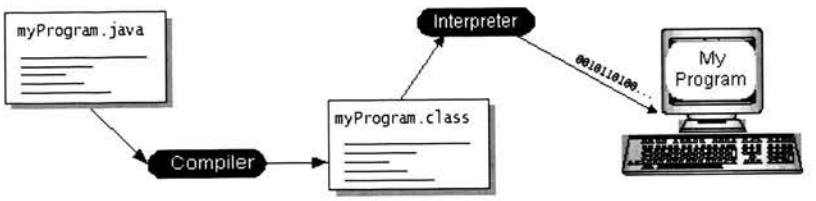


Figure 3.1: Funcionamento da Linguagem de Programação Java

Devido à *Java Virtual Machine* estar disponível para vários sistemas operativos, o mesmo ficheiro .class pode ser executado em plataformas diferentes.

A Plataforma Java

Uma plataforma é o *hardware* ou *software* no qual um programa é executado. Algumas plataformas muito conhecidas são Microsoft Windows, Linux, Solaris OS e MacOS. A maioria das plataformas podem ser descritas como uma combinação de sistema operativo e *hardware*.

A plataforma Java difere da maioria das plataformas devido a ser uma plataforma de apenas *software*, que funciona sobre plataformas baseadas em *hardware*. Esta plataforma tem dois componentes:

- *Java Virtual Machine*
- *Java Application Programing Interface (API)*

A figura 3.2 ilustra como a API e a JVM suportam o programa a partir de uma plataforma com base no hardware.

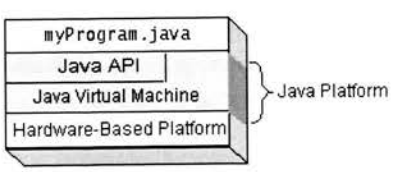


Figure 3.2: Arquitectura da plataforma

A API é uma grande colecção de componentes de software compilados que disponibilizam muitas capacidades, como o *Swing* (bibliotecas para interface gráfica). É um grupo de bibliotecas de classes e interfaces. Essas bibliotecas são conhecidas como *packages*.

3.3 Serviços Web

De acordo com o W3C, um serviço web é um sistema de software desenhado para suportar interação entre máquinas, podendo sendo sistemas diferentes com softwares diferentes, sobre uma rede. A sua interface pode ser descrita num formato processável por máquinas como o WSDL. A sua informação (incluido o WSDL) e publicada usando o protocolo UDDI. Os outros sistemas interagem com o serviço web de acordo com a sua interface usando mensagens, que podem ser incluidas num envelope SOAP. Essas mensagens são tipicamente trocadas usando o protocolo HTTP, e normalmente comprometendo XML em conjunção com outros standards da Web. As aplicações de software escritas em várias linguagens de programação e funcionando em várias plataformas podem usar serviços web para trocar informação pela rede.

As vantagens do uso de serviços web são:

- Possibilitam a troca de informação entre máquinas diferentes com diferentes softwares
- Uso de standards e protocolos abertos. Os protocolos e formato de informação são baseados em texto quando possiveis, tornando mais fácil de desenvolvedores compreenderem.
- Utilizando HTTP, os serviços web pode trabalhar através das firewalls sem ser necessário modificações na firewall.
- Permite software e serviços de diferentes entidades e locais ser combinados facilmente possibilitando serviços integrados.
- Permite o reuso de serviços e componentes dentro de uma infra-estrutura.

3.3.1 Standards Usados

Os standards são usados para defenir, localizar, implementar e fazer serviços web.
Eles distribuiem-se por 4 áreas:

- Transporte do serviço: é responsavel por transportar mensagens entre aplicações ligadas à rede, e inclui protocolos tais como HTTP, SMTP, FTP, e mais recentemente *Blocks Extensible Exchange Protocol* (BEEP).
- Troca de mensagens XML: é responsavel por codificar as mensagens no formato XML, para que as mensagens possam ser entendidas em uma ou outra extremidade da conexão da rede. Recentemente, esta área inclui protocolos como XML-RPC, SOAP e REST.
- Descrição do serviço: é usada para descrever para descrever a interface pública para um serviço web específico. A interface WSDL é usada para este propósito.
- Descoberta do serviço: centraliza serviços num registo comum tal que os serviços web podem pubilicar a sua localização e descrição e tornar fácil encontrar os serviços disponíveis na rede. No nomento, a API UDDI é usada normalmente para encontrar serviços. (no presente projecto não iremos usar esta área)

XML

XML (eXtensible Markup Language) é uma recomendação da W3C para gerar linguagens de marcação para necessidades especiais. É um subtipo de SGML (Standard Generalized Markup Language - Linguagem Padronizada de Marcação Genérica) capaz de descrever diversos tipos de dados. Seu propósito principal é a facilitar a partilha de dados entre diferentes sistemas, particularmente sistemas ligados via Internet.

Em baixo nivel, num documento XML toda a informação manifesta-se como o texto, entremeado com marcas que indicam a separação da informação em uma hierarquia de dados de caráteres (elementos), e atributos daqueles elementos. Essas marcas (elementos) codificam a descrição da disposição (*layout*) e estrutura lógica do documento. O XML possibilita um mecanismo para impor condições na disposição (*layout*) e estrutura lógica.

Algumas vantagens do XML são:

- simultaneamente um formato legível por máquinas e humanos;
- suporte para Unicode (padrão de codificação de caracteres), permitindo qualquer informação em qualquer lingua humana ser comunicada;
- abilidade para representar a maoria da estrutura de dados da informática (registos, listas e árvores);
- um formato auto-documentado que descreve a estrutura e o nome dos campos assim como os valores específicos;
- a sintaxe estrita e os requesitos de análise que permitem os necessário algoritmos de análise se tornarem simples, efícientes e consistentes;
- robusto, o formato é verificavel logicamente baseado em padrões internacionais;
- a estrutura hierárquica é apropriada para a maioria (mas não todos) tipos de originais;
- manifesta-se em ficheiros de texto, desprovidos de licenças e restrições;
- independente da plataforma, assim relativamente imune às mudanças na tecnologia;
- o XML e seu predecessor, SGML, estiveram no uso desde 1986, assim há úma experiência extensiva e um conjunto alargado de *software* disponíveis.

Em seguida, vemos um exemplo de um documento XML.

```
<?xml version="1.0" encoding="UTF-8"?>

<Receita nome="pão" tempo_de_preparo="5 minutos" tempo_de_cozimento="3 horas">
   <título>Pão simples</título>
   <ingrediente quantidade="3" unidade="xicaras">Farinha</ingrediente>
   <ingrediente quantidade="7" unidade="gramas">Fermento</ingrediente>
   <ingrediente quantidade="1.5" unidade="xicaras" estado="morna">Àgua</ingrediente>
   <ingrediente quantidade="1" unidade="colheres de chá">Sal</ingrediente>
   <Instruções>
      <passo>Misture todos os ingredientes, e dissolva bem.</passo>
      <passo>Cubra com um pano e deixe por uma hora em um local morno.</passo>
      <passo>Misture novamente, coloque numa bandeja e asse num forno.</passo>
   </Instruções>
</Receita>
```

Onde temos na primeira linha:

```
<Receita nome="pão" tempo_de_preparo="5 minutos" tempo_de_cozimento="3 horas">
```

"Receita" é o nome principal para o seu documento. Note que a semelhança entre XML e HTML é grande, na 1ª linha abrimos a tag Receita e na última linha fechamos a mesma, como em HTML e assim se estende por todo o exemplo.

"Receita" é um elemento e "nome", "tempo_de_preparo", "tempo_de_cozimento" são atributos.

XML Schema

O XML Schema é uma descrição de um tipo de documento XML, tipicamente expresso em restrições na estrutura e conteúdo de documentos daquele tipo. O XML Schema permite um vista do tipo de documento a um alto nível de abstração.

Existem linguagens desenvolvidas especificamente para descrever XML Schemas. Entre as quais se destaca o XML Schema (XSD).

O processo de verificação se um ficheiro XML está de acordo com um XML Schema é chamado de validação. Assim os documentos serão considerados válidos se eles satifazerem os requerimentos do esquema com que estejam associados.

As restrições de um XML Schema são do tipo:

- Tem que ter certos Elementos e atributos, com uma certa estrutura
- A forma como os caracteres são interpretados (exemplo: número, data, URL)

SOAP

O SOAP (Simple Object Access Protocol) é um protocolo leve para troca de informação estruturada em um ambiente descentralizado, distribuído. Usa tecnologias de XML (ver secção 3.3.1) para definir uma estrutura extensível de troca de mensagens que fornece uma construção da mensagem que possa ser trocada sobre uma variedade de protocolos subjacentes. A framework foi projectada para ser independente de todo o modelo de programação particular e da outra semântica de implementação. Duas principais caracteristicas do SOAP é simplicidade e extensíbilidade.

Em seguida temos um exemplo de uma troca de mensagens SOAP. No presente exemplo, um cliente pede informação de um produto de um serviço web fictício da warehouse. O cliente necessita saber que produto corresponde com o ID 827635:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getProductDetails xmlns="http://warehouse.example.com/ws">
      <productID>827635</productID>
    </getProductDetails>
  </soap:Body>
</soap:Envelope>
```

Aqui está apresentado como o serviço da warehouse pode formatar sua mensagem de resposta:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
     <getProductDetailsResponse xmlns="http://warehouse.example.com/ws">
       <getProductDetailsResult>
         <productName>Toptimate 3-Piece Set</productName>
         <productID>827635</productID>
         <description>3-Piece luggage set. Black Polyester.</description>
         <price>96.50</price>
         <inStock>true</inStock>
       </getProductDetailsResult>
     </getProductDetailsResponse>
   </soap:Body>
 </soap:Envelope>
```

WSDL

WSDL

Enquanto os protocolos de comunicações e os formatos da mensagem são standards na comunidade web, torna-se cada vez mais possível e importante a descrição das comunicações de forma estruturada. WSDL (Web Service Description Language) dirige-se a esta necessidade definindo uma gramática de XML para descrever serviços de rede como coleções de endpoints de uma comunicação capazes de trocar mensagens. As definições do serviço de WSDL fornecem a documentação para sistemas distribuídos e servem como uma receita para automatizar os detalhes envolvidos em uma comunicação das aplicações.

Um documento WSDL usa os seguintes elementos para descrever um serviço web:

- Types - um recipiente para o tipo de dados usa num sistema (tal como XSD);
- Message - uma definição abstrata dos dados a serem comunidados;
- Operation - uma definição abstrata de uma acção suportada por um serviço;
- Port Type - um conjunto abstrato de acções suportado por um ou mais portos (endpoints);
- Binding - um protocolo concreto (SOAP, HTTP, MIME) e especificação do tipo de dados para um *Port Type* particular;
- Port - um porto definido como a combinação de *Binding* e um endereço;
- Service - uma coleção de portos relacionados.

Aqui fica um exemplo de um documento WSDL, que pode ser a descrição do serviço web que trocou a mensagens SOAP num exemplo na secção 3.3.1:

```
<?xml version="1.0"?>
<definitions name="WareHouse" targetNamespace="http://example.com/warehouse.wsdl" xmlr
xmlns:xs="http://www.w3.org/2000/10/XMLSchema"
xmlns:xsd1="http://example.com/warehouse.xsd" xmlns:soap="http://schemas.xmlsoap.org/w
  <types>
    <schema targetNamespace="http://example.com/warehouse.xsd"                xmlns="ht
      <element name="productDetails">
        <complexType>
          <sequence>
            <element name="productName" type="string"/>
            <element name="productID" type="int"/>
            <element name="description" type="string"/>
            <element name="price" type="float"/>
            <element name="inStock" type="string"/>
          </sequence>
        </complexType>
      </element>
    </schema>
  </types>
```

```
<message name="getProductDetailsInput">
  <part name="productID" type="xs:int"/>
</message>
<message name="getProductDetailsOutput">
  <part name="getProductDetailsResult" type="productDetails"/>
</message>
<portType name="WareHousePortType">
  <operation name="getProductDetails">
    <input message="tns:getProductDetailsInput"/>
    <output message="tns:getProductDetailsOutput"/>
  </operation>
</portType>

<binding name="WareHouseSoapBinding" type="tns:WareHousePortType">
  <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="getProductDetails">
    <input>
      <soap:body use="literal" namespace="http://warehouse.example.com/ws"/>
    </input>
    <output>
      <soap:body use="literal" namespace="http://warehouse.example.com/ws"/>
    </output>
  </operation>
</binding>

<service name="WareHouseService">
  <port name="WareHousePort" binding="tns:WhareHouseSoapBinding">
    <soap:address location="http://example.com/warehouse"/>
  </port>
</service>
<definitions>
```

3.4 Servidores

3.4.1 Postgresql

Os serviços web, em correcto funcionamento, necessitam de armazenar, consultar, actualizar e remover dados sobre informações relativas às eleições. O AS, BS, VC gerem dados de configuração para o seu correcto funcionamento numa votação. O TS gere dados relativos aos eleitores. O EDS gere toda a informação de uma eleição.

O uso de base de dados neste sistema é necessário para que o sistema suporte grande volume de dados com uma boa eficiência, assim como consiga guarda-los de forma estruturada. Neste projecto, decidimos usar o sistema de gestão de base de dados *Postgresql*.

Uma base de dados guarda uma colecção de informação associada entre si, constituída por subconjuntos ordenados e estruturados, de tal forma que os conteúdos armazenados possam ser facilmente acedidos e manipulados.

Devido à estrutura e organização das base de dados, qualquer infomação pode ser retornada rápida e facilmente.

O Sistema de Gestão de Base de Dados (SGBD) integra todo o software de criação, acesso e manutenção da Base de Dados. Este sistema pode suportar várias Base de Dados. As aplicações recorrem ao SGBD. O SGBD suporta dados persistentes, acesso arbitrário e eficiente a grandes quantidades de dados, gestão de transacções (concorrência), controle de acesso (segurança), robustez (integridade e recuperação), etc.

Quando se pretende organizar informação a maneira mais simples e comum de o fazer é recorrendo ao uso de tabelas. As tabelas contém várias entidades com informação associada entre si partilham atributos comuns. As colunas das tabelas referem-se a atributos comuns entre os vários elementos, ao passo que as linhas das tabelas serão preenchidas pelas várias propriedades de cada elemento particular.

Apesar das base de dados oferecerem todos os mecanismos necessários a uma gestão eficiente de grandes quantidades de dados é da responsabilidade de quem cria o esquema da base de dados são consistentes e que a actualização de uma qualquer entrada poderá ser feita de forma rápida e sem possibilidade de inserção de erros. Para garantir eficiência e integridade dos dados a informação deve respeitar o modelo relacional.

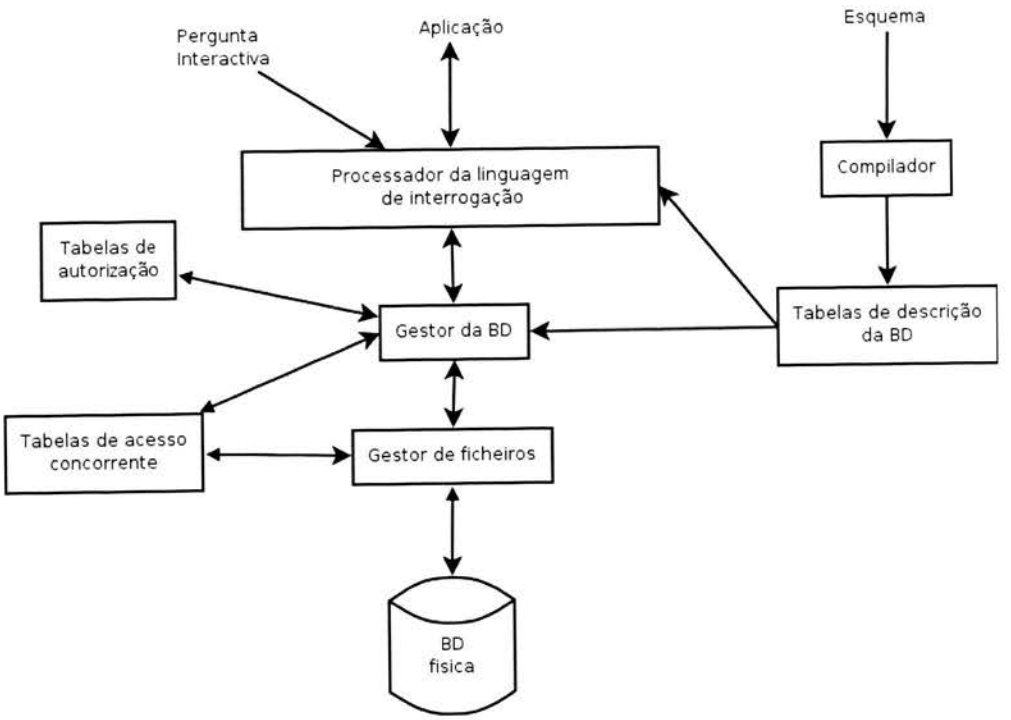


Figure 3.3: Diagrama de SGBD

Postgresql é um sistema de gestão de base de dados relacional livre, disponível sobre a licença *BSD*. Este sistema é uma boa alternativa aos mais variados sistemas de base de dados, entre os quais sistemas *open-source*, como *Ingres*, *MySQL* e *Firebird*, e os sistemas proprietários, como Oracle, Sybase, IBM's DB2 and Microsoft SQL Server. A sua linguagem de interrogação usada é SQL, que vai ser descrita a seguir.

SQL

SQL (Linguagem de Consulta Estruturada) é a mais popular linguagem usada para criar, modificar, obter e manipular dados de sistemas de gestão de base de dados relacional. A linguagem evoluiu além de sua finalidade original, que era suportar base de dados objecto-relacional. Agora é um padrão de ANSI/ISO. Muitas das características originais do SQL foram inspiradas na álgebra relacional.

3.4.2 Apache Tomcat

O Tomcat é um contentor de servlets. Na tecnologia Java, um contentor de servlets é responsável por receber pedidos Web e passá-los às aplicações Web em Java (ver figura). Este contentor de serverlets é aquele que é usado como implementação de referência para as tecnologias Servlet de Java e JSP.

Devido ao Tomcat incluir o seu próprio servidor de HTTP internamente, ele também é considerado um servidor web.

Tomcat é uma implementação livre e aberta, desenvolvida sob o projecto Jakarta na fundação de Software da Apache e está disponível para uso comercial sob a licença ASF no site web da Apache.

No actual momento, o Tomcat está na versão 5.5, mas nós utilizamos a versão 5.0 por considerarmos mais estável. Neste trabalho não trabalhamos directamente com o Tomcat, mas servimo-nos dele para usar a aplicação web Axis (ver secção **??**), e esta vai-nos permitir publicar serviços web.

3.5 Aplicações Web

Neste trabalho usado a aplicação web Apache Axis, que para além de nos ter permitido publicar os serviços, permitiu desenvolver os serviços web, fornecendo uma API. Neste momento existe a versão Axis1 1.4 e o Axis2 versão 1.0, que se apresenta estável. No momento que iniciamos o projecto o Axis 2 ainda estava numa versão testing, por isso decidimos utilizar o Axis1.

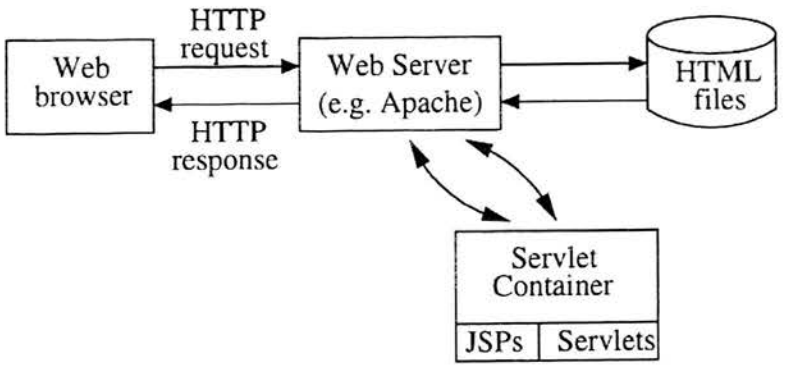


Figure 3.4: Contentor de Servlets interagindo com um servidor web.

O Axis essensialmente é um motor de SOAP - uma framework para construir processadores de SOAP, como clientes, servidores (serviços web), gateways, etc.

O Axis inclui também

- um simples servidor stand-alone
- aplicação web que integra no Tomcat
- suporte a WSDL
- ferramentas para gerar classes Java a partir do WSDL e vice-versa

Usando Apache Axis, os programadores podem criar aplicações de computação distribuída, podendo-se ligar a máquinas com software diferente.

Para publicar um serviço em existe dois caminhos distintos.

Um caminho, é desenvolver um class Java, e alterar o a extensão .java para *.jws e colocar no directorio da aplicação web Axis, cada método publica da class é uma operação. Este caminho é o mais fácil, mas o mais limitativo, por exemplo não permite novos mapeamentos de XML, assim como não permite utilização de packages. Um problema é que se baseia apenas na class desenvolvida, não se preocupando em um serviço padronizado.

Outro caminho é descrever o serviço a partir de Web Service Deployment Descriptor (WSDD) format, que permite descrever de maneira simples o serviço para o Axis o poder disponibilizar.

Para o desenvolvimento do serviço web também existe dois caminhos. O primeiro desenvolver as classes do serviço e depois descrever o serviço para publicar. Outro é criar um WSDL com a descrição do serviço e usar o WSDL2Java (classe disponibilizada pelo Axis) para gerar as classes necessária para implementar as interfaces do serviço, assim como o ficheiro WSDD para publicar

No presente trabalho decidimos usar um caminho que passa-se pelo desenvolvimento da descrição do serviço para que seja mais fácil a interação entre diversas aplicações cliente (exactamente foi o segundo caminho tanto no caso da publicação como no caso do desenvolvimento).

3.6 Segurança

Num sistema de voto electrónico a segurança é muito importante, tendo que atender aos seguintes requisitos:

- Integridade do sistema
- Confidencialidade, integridade e fiabilidade dos dados
- Anonimato do eleitor
- Abertura do sistema (o sistema deve ser open-source)

O último requisito é necessário para tornar o sistema de voto mais transparente e fiavel.

O 3º requisito consegue-se com a dissimulação da identidade do eleitor.

Quanto aos 2 primeiros podem ser garantidos utilizando criptografia assimétrica.

3.6.1 XML Signature

XML Signature é uma recomendação W3C que define a sintaxe xml para assinaturas digitais. Funcionalmente, é muito comum com PKCS#7 mas é mais extensivel e feito para assinar documentos XML. É usado por várias tecnologias web como SOAP, SAML entre outras.

XML Signature pode ser usado para assinar informação de qualquer tipo, tipicamente documentos xml, mas qualquer informação acessivel via URL pode ser assinada.

3.6.2 XML Encryption

XML Encryption é uma recomendação do W3C que define como encriptar informação e representar em XML.

A informação pode ser um tipo arbritrário (incluindo documento xml), um elemento xml, ou o conteúdo dum elemento xml. O resultado da encriptação é um elemento XML Encryption *EncryptedData*, que contém ou identifica a informação cifrada.

Na encriptação de um elemento XML ou seu conteúdo, o elemento *EncryptedData* substitui o elemento ou seu conteúdo (respectivamente) pela versão encriptada do documento xml.

Na encriptação de um tipo arbitário de dados (incluindo documentos xml inteiros), o elemento *EncryptedData* pode se tornar a raiz do novo documento xml ou tornar-se um elemento filho numa aplicação documento xml.

3.7 API's

API, de Application Programming Interface (ou Interface de Programação de Aplicativos) é um conjunto de rotinas e padrões estabelecidos por um software para utilização de suas funcionalidades por programas aplicativos.

3.7.1 Apache Axis

O Apache Axis é uma aplicação web que fornece uma API para desenvolvimento de serviços web baseados em troca de mensagens SOAP (para mais informação ver secção 3.5).

3.7.2 Swing

Swing é uma API Java para interfaces gráficas. Ela é compatível com a API AWT, mas trabalha de uma maneira totalmente diferente. A API Swing procura renderizar\desenhar por contra própria todos os componentes, ao invés de delegar essa tarefa ao sistema operacional, como a maioria das outras APIs de interface gráfica trabalham.

Por ser uma API de mais alto nível, ou seja, mais abstracção, menor aproximação das APIs do sistema operacional, ela tem bem menos performace que outras APIs gráficas e consome mais memória RAM em geral. Porém, ela é bem mais completa, e os programas que usam Swing tem uma aparência muito parecida, independente do Sistema Operacional utilizado.

3.7.3 JDBC

Java Database Connectivity ou JDBC é um conjunto de classes e interfaces (API) escritas em Java que faz o envio de instruções SQL para qualquer base de dados relacional. É uma API de baixo nível e base para API's de alto nível.

Para cada Banco de dados há um driver JDBC.

3.7.4 XML Security

XML Security é uma ferramenta que possibilita características de segurança, tais como, assinatura digital, encriptação e controlo de acesso a documentos XML. Estas características estão por detrás da capacidade de protocolos de segurança na camada de transporte como o Secure Sockets Layer (SSL). O objectivo na criação desta tecnologia são contribuir para o desenvolvimento de padrões permitindo simples implementações e pedir tecnologias avançadas para aos sócios e aos colaboradores e para recolher a sua entrada.

Arquitectura

Para além do *AS, BS* e *VC* decidimos que seriam necessários mais dois sistemas para responder aos requisitos apresentados anteriormente. Esses novos sistemas sistemas são: *Trust System* e *Election Definition System*. Os motivos para tal decisão serão apresentados a medida que estes serviços forem descritos.

Depois serão apresentadas a arquitectura do projecto (lógica e física) e para finalizar este capitulo serão apresentadas os modelos das classes de domínio.

Os WSDLs dos sistemas apresentados (*Authorization System, Ballot System, Vote Collector, Election Definition System* e *Trust System*) podem ser consultados através da nossa página de projecto[23].

3.8 Módulo de Configuração

3.8.1 Electoral Definition System

Depois de termos analisado a Plataforma Genérica de Votação Electrónica decidimos que seria necessário criar um sistema que permiti-se a criação e configuração uma eleição, visto que o objectivo do projecto era implementar os vários sistemas da PGVE em *webservices*. Como os sistemas aqui apresentados são disponibilizados em serviços web, a sua configuração remota requeria um sistema de configuração adicional. O *Election Definition System* (EDS).

O acesso a estes serviços web tem que ser necessariamente por aplicações cliente. Para além da aplicação do votante e da aplicação de contagem (que já existem na PGVE) existe agora uma terceira aplicação, a da comissão eleitoral. Esta aplicação vai ser a responsável pela criação / configuração de novas eleições.

Para disponibilizar um serviço web a esta nova aplicação foi criado o *Election Definition System* (EDS). Este serviço permite:

- Modificar os membros da comissão eleitoral
- Modificar informação das eleições (nome da eleição, datas de inicio e fim, circulos eleitorais,endereço dos serviços web envolvidos)
- Modificar informação dos circulos eleitorais (nome do circulo eleitoral, lista de votantes, boletim de voto, endereço dos serviços web envolvidos)

Assim a comissão eleitoral cria e configura a eleição e depois solicita ao EDS a configuração dos sistemas (AS, BS e VC), em que o EDS envia-lhes a informação que necessitam.

Depois da configuração, procede-se à votação. O eleitor para iniciar a sua votação solicita informação sobre a eleição ao EDS, por exemplo os endereços do AS e BS e boletim de voto.

Como o EDS necessita de configurar os sistemas e obter as chaves públicas destes, terá assim o papel de actor pelo interacção entre o EDS e os restantes sistemas.

Modelos de Casos de Utilização do Election Definition System

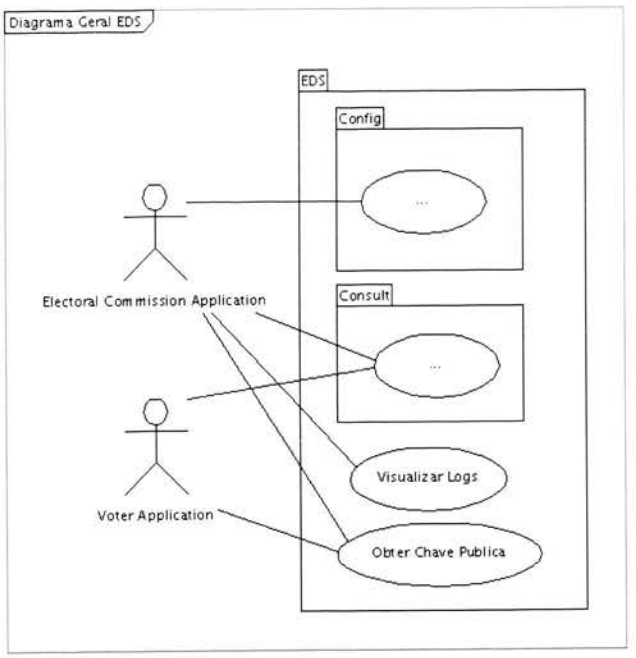


Figure 3.5: Diagrama geral de casos de utilização para o EDS

Para o caso geral da figura temos:

- Obter Chave Pública -Um actor obtém a chave pública do par de chaves do sistema.
- Visualizar Logs -O *Electoral Commission Application* obtém os acessos que determinado membro da comissão eleitoral teve ao sistema. Um membro apenas pode ver os registos relativos à eleição em que se encontra inserido.

Para os restantes módulos(Config e o Consult) temos:

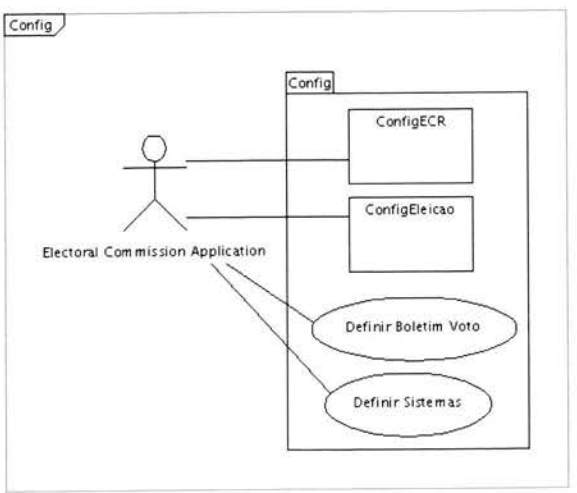


Figure 3.6: Diagrama geral de casos de utilização para o Config

- Definir Boletim de Voto -O *Electoral Commission Application* submete o boletim de voto para um ou mais circulos eleitorais.
- Definir Sistemas -O *Electoral Commission Application* define os endereços dos sistemas intervenientes em cada circulo eleitoral.

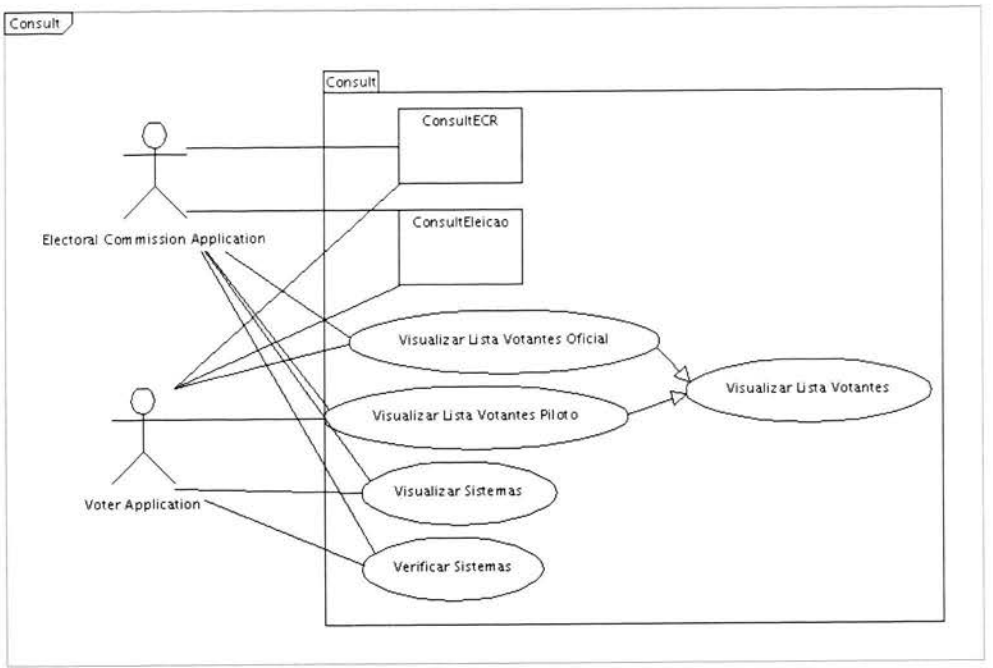


Figure 3.7: Diagrama geral de casos de utilização para o Consult

- Visualizar Lista de Votantes Oficial -O Actor visualiza a lista de votantes referente a um ou mais circulos eleitorais ou a uma eleição. Esta lista de votantes é diferente da lista de votantes piloto.
- Visualizar Lista de Votantes Piloto -O Actor visualiza a lista de votantes referente a um ou mais circulos eleitorais ou a uma eleição.
- Visualizar Lista de Votantes -O Actor visualiza a lista de votantes referente a um circulo eleitoral ou a uma eleição.
- Visualizar Sistemas -O Actor visualiza sistemas referente a um ou mais circulos eleitorais ou a uma eleição.
- Verificar Sistemas -O Actor verifica se os sistemas estão online e prontos para iniciar o processo de votação.

Aplicação da Comissão Eleitoral

Para além da aplicação de voto e de contagem implementamos uma aplicação para a comissão eleitoral. Os casos de uso para este aplicação serão apresentados a seguir:

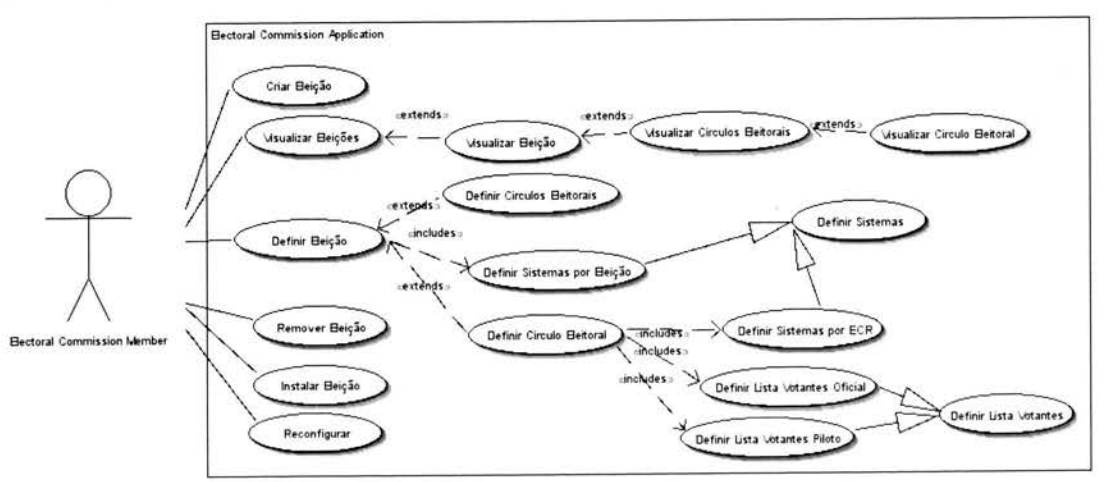


Figure 3.8: Diagrama de casos de utilização para à aplicação da comissão eleitoral

Para o caso da aplicação eleitoral temos:

- Criar Eleição - O *Electoral Commission Member* cria uma eleição
- Visualizar Eleições - O ECM visualiza as eleições
  - Visualizar Eleição - O ECM visualiza a eleição pretendida
    - Visualizar Círculos Eleitorais - O ECM visualiza uma lista de círculos eleitorais
      - Visualizar Círculo Eleitoral - O ECM visualiza as características de um círculo eleitoral
- Definir Eleição - O ECM define as características de uma eleição
  - Definir Círculos Eleitorais - O ECM define círculos eleitorais para uma eleição.
  - Definir Círculo Eleitoral - O ECM define um Círculo Eleitoral.
    - Definir Sistemas por ECR - O ECM define os sistemas para um círculo eleitoral.
      - Definir Lista Votantes Oficias - O ECM define a lista de votantes para a eleição oficial e teste público.
      - Definir Lista Votantes Piloto - O ECM define lista de votantes para o teste piloto.
    - Definir Sistemas por Eleição - O ECM define os sistemas para uma eleição
- Instalar Eleição - O ECM instala e configura os sistemas (AS, BS, VCs) para uma eleição.
- Reconfigurar - O ECM tem a possibilidade de reconfigurar os parâmetros da eleição.
- Remover Eleição - O ECM remove a eleição.

3.9 Módulo de Autenticação

3.9.1 Trust System

Visto que cada comissão eleitoral (ou entidade promotora das eleições) terá diferentes métodos de validação de eleitores, o nosso sistema delega um *Trust System* (TS) externo que pode eventualmente ser disponibilizado por serviços web. A função deste TS é a autenticar os eleitores e fornecer uma credencial de eleitor que identifica esse eleitor. Este serviço pode ser simplesmente substituído por um método mais clássico de identificação como por exemplo: *Smart Card*.

Com estes dois novos sistemas: *Election Definition System* e o *Trust System* passamos a ter um novo cenário (ver a figura3.9).

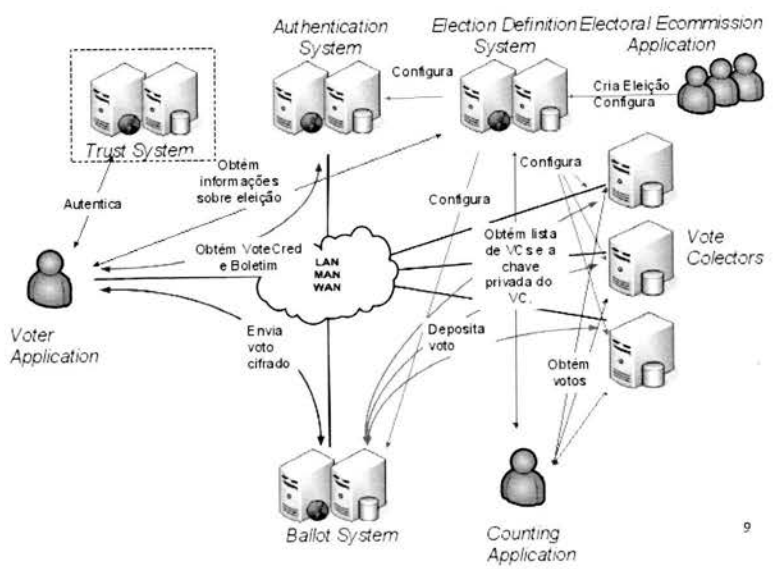


Figure 3.9: Cenário do Serviços Web de Votação Electrónica

Modelos de Casos de Utilização do Trust System

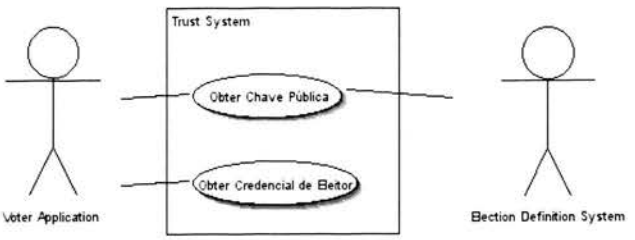


Figure 3.10: Diagrama de casos de utilização para o TS

Para o caso do Trust System temos:

- Obter Chave pública -Um actor obtém a chave pública do par de chaves do sistema
- Obter Credencial Eleitor -Um *Voter* obtém um credencial de autenticação única, que identifica um eleitor.

3.10 Arquitectura Lógica

A arquitectura lógica pode ser descomposta em duas partes: decomposição horizontal e decomposição vertical.

A decomposição horizontal abrange as quatro camadas que são comuns aos vários módulos apresentados anteriormente. Estas camadas serão posteriormente apresentadas.

Os módulos (módulos de votação, módulo de configuração, módulo de autenticação e os módulos de aplicação) referidos anteriormente são incluidos na decomposição vertical.

3.10.1 Decomposição Horizontal

Divisão em Pacotes

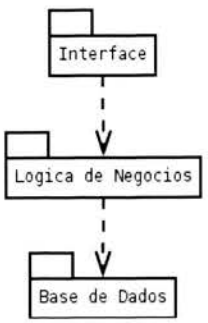


Figure 3.11: Diagrama da divisão de pacotes.

Interface

Esta é a camada responsável pela interacção entre o Utilizador e o Sistema. É implementada com o auxilio da API SWING no caso dos módulos das Aplicações. No caso dos módulos de votação, módulo de configuração e módulo de autenticação a interface são webservices.

Lógica de Negócio

Esta camada processa os pedidos do utilizador e faz a conversão destes pedidos para formatos adequados. É aqui que se estabelece a ligação entre o utilizador (Cliente/Votante) e o sistema (Serviço). A troca de mensagens entre o cliente e o servidor é feita em SOAP. Cada sistema é descrito pelo seu respectivo WSDL.

Base de Dados

Esta é a camada pela interacção com a base de dados relacional PostgreSQL. Nesta Camada são feitas as conversões necessárias para o posterior armazenamento na base de dados relacional.

A conexão e acesso a base de dados é feito com recurso ao JDBC e utilizando as instruções SQL.

3.10.2 Decomposição Vertical

Aqui será apresentada a Decomposição Vertical da nossa Arquitectura Lógica.

No diagrama podemos observar os vários módulos que constituem esta decomposição.

Estes módulos serão apresentados individualmente de uma forma mais detalhada, dando maior importância aos módulos de votação (*AS, BS* e *VC*), módulo de configuração (*EDS*) e módulo de autenticação (*TS*).

Authorization System

As operações que este sistema oferece são as seguintes:

- *getPublickey* - Este serviço retorna a chave pública do *AS*
- *getVoteCred* - Retorna uma credencial de votação
- *setSystem* - Configura o servidor para um *ECR*

Ballot System

O *Ballot System* possui as seguintes operações:

- *getPublickey* - Este serviço retorna a chave pública do *BS*
- *postdCiB*- Envia o voto encriptado e a credencial
- *setSystem* - Configura o servidor para um *ECR*

Vote Collector

Para o *Vote Collector* temos as seguintes operações:

- *getPublickey* - Este serviço retorna a chave pública do *VC*
- *postVote* - Recebe a opção de voto
- *setSystem* - Configura o servidor para um *ECR*
- *getVotes* - Retorna os votos da eleição

Trust System

O *Trust System* possui as seguintes operações:

- *getPublickey* - Este serviço retorna a chave pública do *TS*
- *getAuthCred* - Retorna uma credencial de autenticação.

Election Definition System

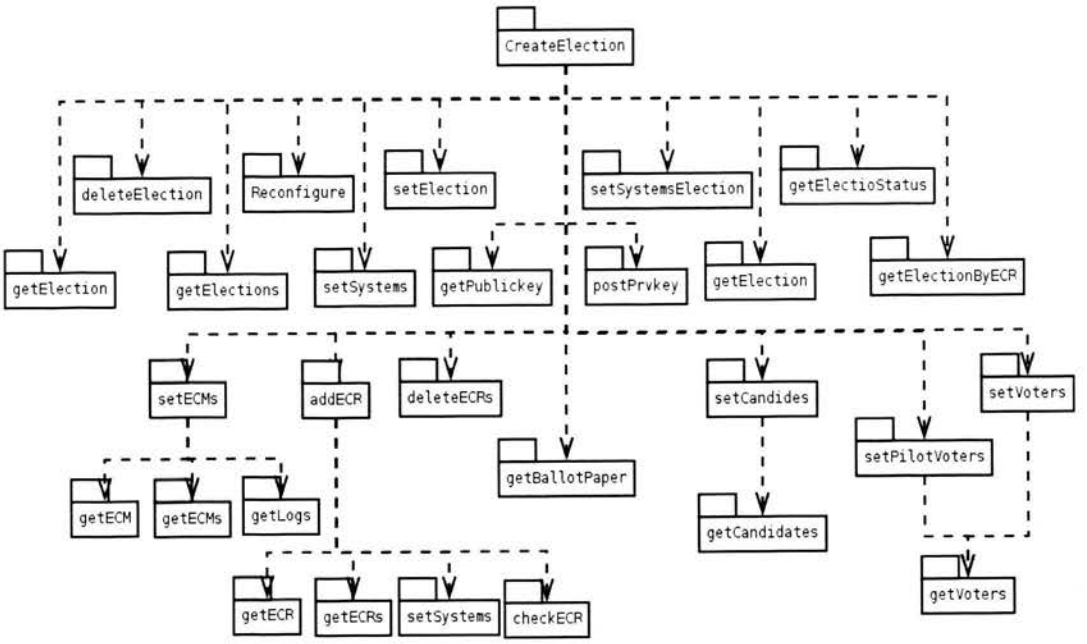


Figure 3.12: Election Definition System

As operações que o *Election Definition System* possui :

- *getPublickey* - Este serviço retorna a chave pública do *EDS.*
- *createElection* - Cria uma Eleição
- *deleteElection* - Apaga uma Eleição
- *getElections* - Mostra eleições do *EDS*, conforme os parâmetros
- *installSystems* - Configura os servidores dos sistemas
- *setElection* - Modifica a Eleição
- *setSystemsElection* - Atribui Sistemas a Eleição
- *getElection* - Retorna informação sobre a Eleição
- *getElectionStatus* - Verifica Estado da Eleição
- *getElectionByECR* - Mostra Eleição pelo *ECR*
- *setECMs* - Modificar Lista de *ECMs*
- *addECR* - Adiciona Lista de *ECRs*
- *deleteECRs* - Apaga Lista de Círculos Eleitorais
- *setCandidates* - Modifica Candidatos
- *setVoters* - Modifica Votantes
- *setPilotVoters* - Modificar Votantes Pilotos
- *getVoters* - Retorna lista de votantes segundo parâmetros
- *getCandidates* - Retorna os Candidatos
- *getLogs* - Retorna os acessos de um *ECM*
- *getECR* - Retorna Informações do *ECR*

- *getECRs* - Visualiza os *ECRs*
- *setSystems* - Atribui Sistemas aos *ECRs*
- *checkECR* - Verifica Configuração do *ECR*
- *getECM* - Retorna informação sobre o *ECM*
- *getECMs* - Retorna *ECMs* de uma eleição
- *getBallotPaper* - Retorna Boletim de voto
- *postPrvkey* -Retorna resultados da eleição, tornando pública a chave privada da eleição.

3.11 Arquitectura Física

A Máquina Cliente pode estar equipada com uma ou mais Aplicações Cliente (Aplicação de Voto, Aplicação da Comissão Eleitoral e Aplicação de Contagem) dependendo do tipo de utilizador.

A Máquina Cliente possibilitará a conexão via HTTP a um ou mais *Web Services* (*Trust System, Authorization System, Ballot System, Vote Collector, Election Definition System*) dependendo da aplicação cliente que o utilizador está a usar.

Cada Sistema (*TS, AS, BS, VC* e *EDS*) tem a sua máquina e estas serão apresentadas em pormenor nas próxima secções. As várias máquinas existentes têm componentes comuns que serão apresentados a continuação:

- axis - Este componente é a API Axis
- xmlsec - Este componente é a API XMLSecurity.
- swve.serializer - Este componente é uma extensão dos *Serializers* e *Deserializers*. Ele permite a serialização objectos java e deserialização de elementos (XML) específicos deste projecto.
- swve.types - Este componente é um package Java com Objectos Java específicos deste projecto.
- swve.utils - Este componentes é um package Java que contém Objectos para optimizar e concentrar processos envolvidos nos serviços.

Os componentes que são exclusivos de determinadas máquinas serão apresentados nas seguintes secções nos respectivos sistemas.

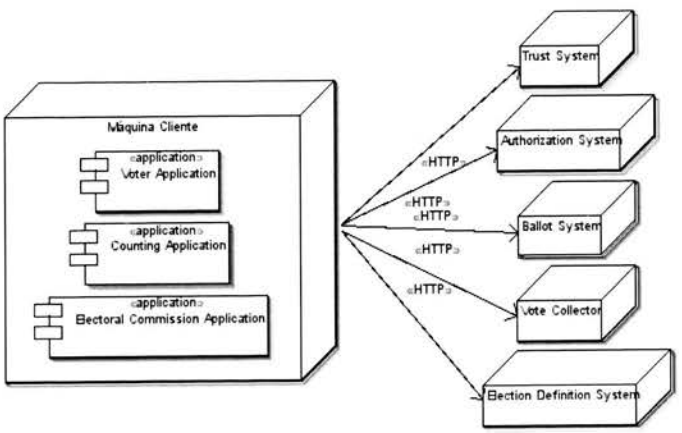


Figure 3.13: Visão Geral

3.11.1 Trust System

Representa a máquina, equipada para suportar o módulo do *Trust System*.
Esta máquina tem para além dos componentes apresentados anteriormente os seguintes:

- TSdb - Este componente representa a Base de Dados do sistema *Trust System*.
- swve.ts - Este componente é um package Java, tipo kernel do serviço. É em volta deste package que se desenvolve a implementação do cliente e do servidor relativamente ao serviço *TS*
- swve.ts.impl - Este componente é um package Java que contém a implementação dos serviços Web do Sistema *TS*.

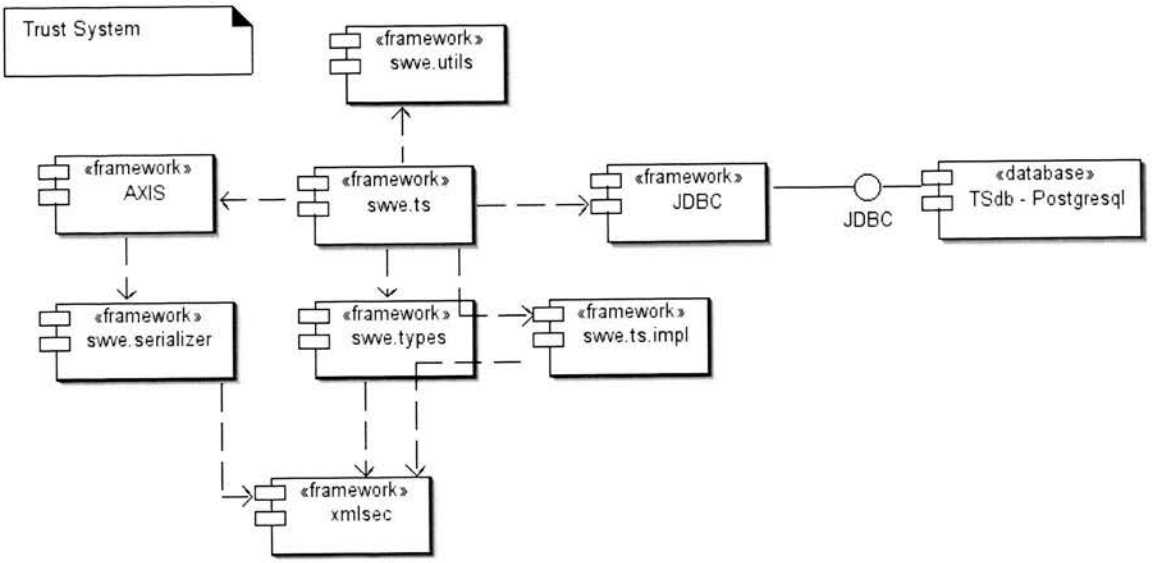


Figure 3.14: Diagrama de Componentes do Trust System

3.11.2 Authorization System

Representa a máquina, equipada para suportar o módulo do *Authorization System*.
Esta máquina tem para além dos componentes apresentados anteriormente os seguintes:

- ASdb - Este componente representa a Base de Dados do sistema *Authorization System*.
- swve.as - Este componente é um package Java, tipo kernel do serviço. É em volta deste package que se desenvolve a implementação do cliente e do servidor relativamente ao serviço *AS*.
- swve.as.impl - Este componente é um package Java que contém a implementação dos serviços Web do Sistema *AS*.

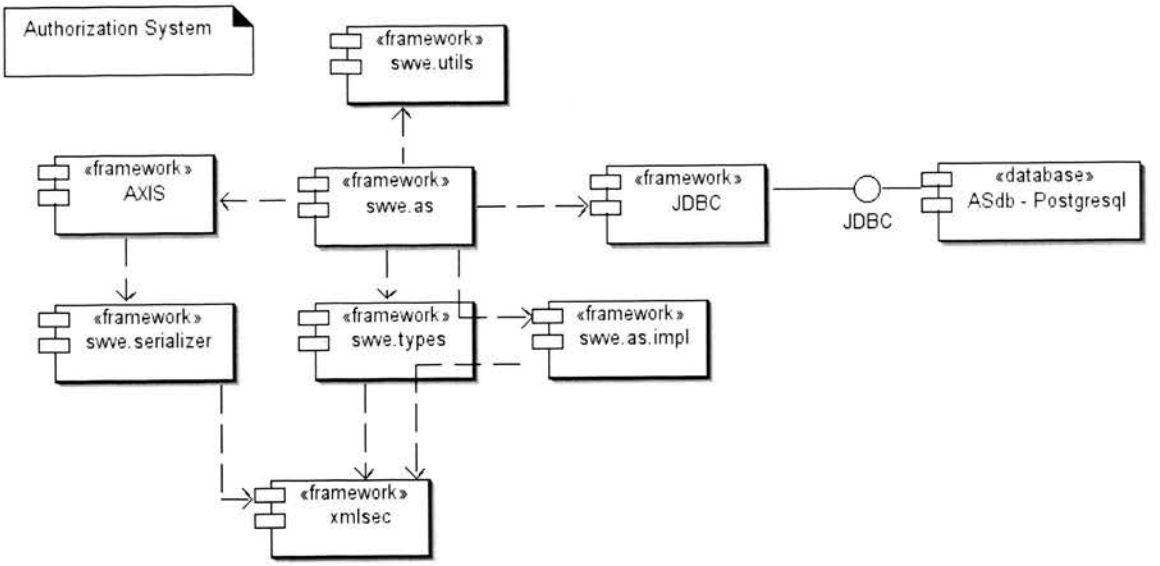


Figure 3.15: Diagrama de Componentes do Authorization System

3.11.3 Ballot System

Representa a máquina, equipada para suportar o módulo do *Ballot System*.
Esta máquina tem para além dos componentes apresentados anteriormente os seguintes:

- BSdb - Este componente representa a Base de Dados do sistema *Ballot System*.
- swve.bs - Este componente é um package Java, tipo kernel do serviço. É em volta deste package que se desenvolve a implementação do cliente e do servidor relativamente ao serviço *BS*.
- swve.bs.impl - Este componente é um package Java que contém a implementação dos serviços Web do Sistema *BS*.

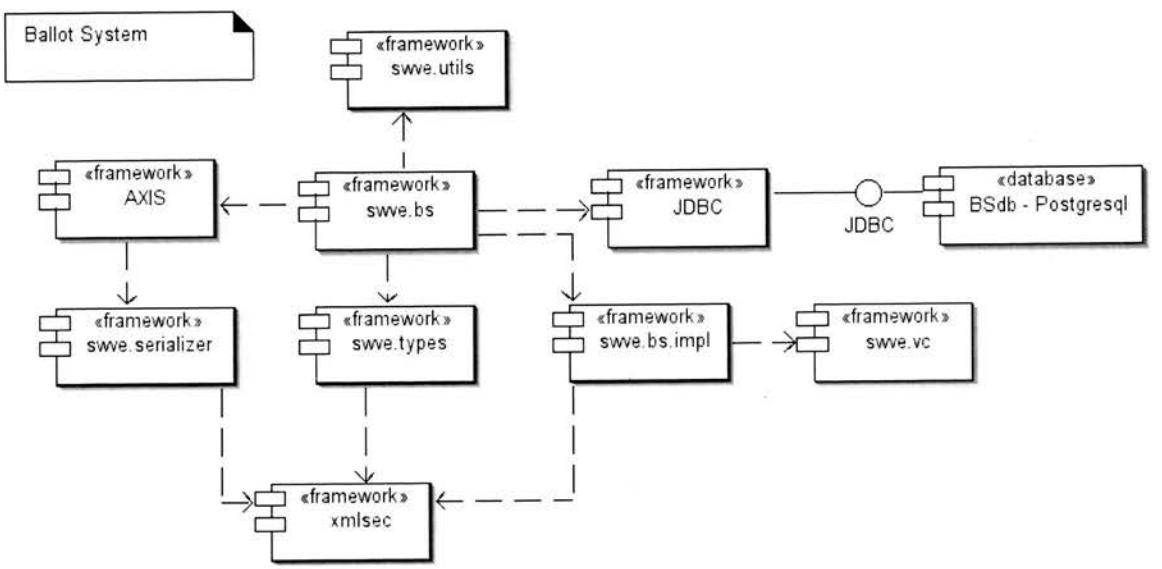


Figure 3.16: Diagrama de Componentes do Ballot System

3.11.4 Vote Collector

Representa a máquina, equipada para suportar o módulo do *Vote Collector*.
Esta máquina tem para além dos componentes apresentados anteriormente os seguintes:

- VCdb - Este componente representa a Base de Dados do sistema *Vote Collector.*
- swve.vc - Este componente é um package Java, tipo kernel do serviço. É em volta deste package que se desenvolve a implementação do cliente e do servidor relativamente ao serviço *VC.*
- swve.vc.impl - Este componente é um package Java que contém a implementação dos serviços Web do Sistema *VC.*

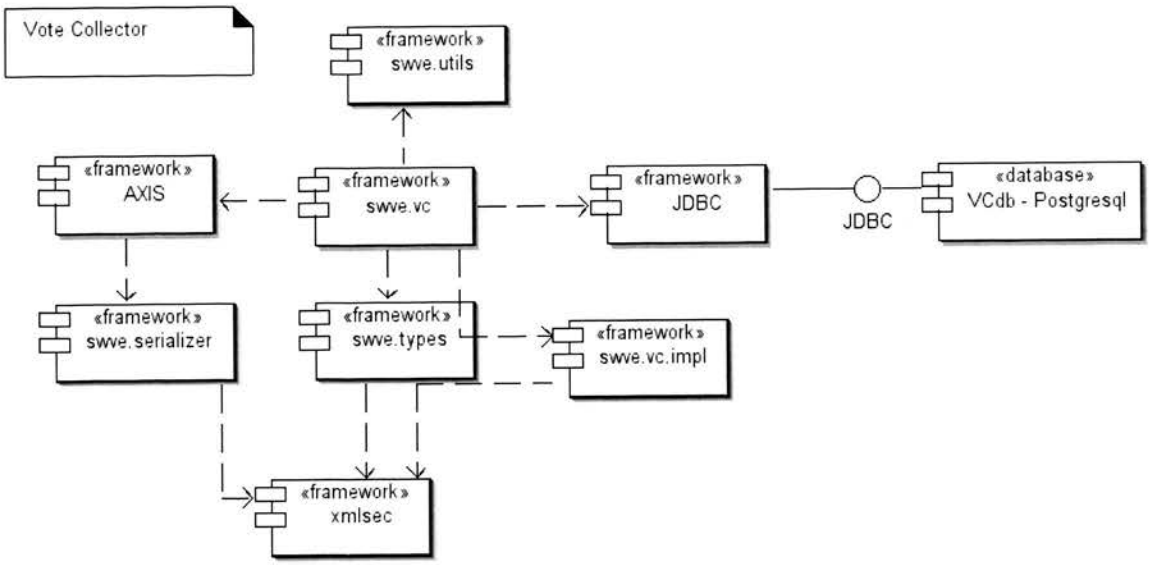


Figure 3.17: Diagrama de Componentes do Vote Collector

3.11.5 Election Definition System

Representa a máquina, equipada para suportar o módulo do *Election Definition System.*
Esta máquina tem para além dos componentes apresentados anteriormente os seguintes:

- EDSdb - Este componente representa a Base de Dados do sistema *Election Definition System.*
- swve.eds - Este componente é um package Java, tipo kernel do serviço. É em volta deste package que se desenvolve a implementação do cliente e do servidor relativamente ao serviço *EDS.*
- swve.eds.impl - Este componente é um package Java que contém a implementação dos serviços Web do Sistema *EDS.*
- swve.eds.impl.config - Este componente é um package que pertence ao package swve.eds.impl. O package contém classes e packages que permitem configurar eleições e circulos eleitorais.
- swve.eds.impl.consult - Este componente é um package que pertence ao package swve.eds.impl. O package contém classes e packages que permitem consultar informação relativa a eleições e circulos eleitorais.
- swve.eds.impl.config.election - Este componente é um package Este componente é um package que pertence ao package swve.eds.impl.config. O package contém classes que permitem configurar as eleições.
- swve.eds.impl.config.ecr - Este componente é um package que pertence ao package swve.eds.impl.config. O package contém classes que permitem configurar os Círculos Eleitorais.
- swve.eds.impl.consult.election - Este componente é um package que pertence ao package swve.eds.impl.consult. O package contém classes que permitem consultar informação relativo às eleições.
- swve.eds.impl.consult.ecr - Este componente é um package que pertence ao package swve.eds.impl.consult. O package contém classes que permitem consultar informação relativa aos Circulos Eleitorais.

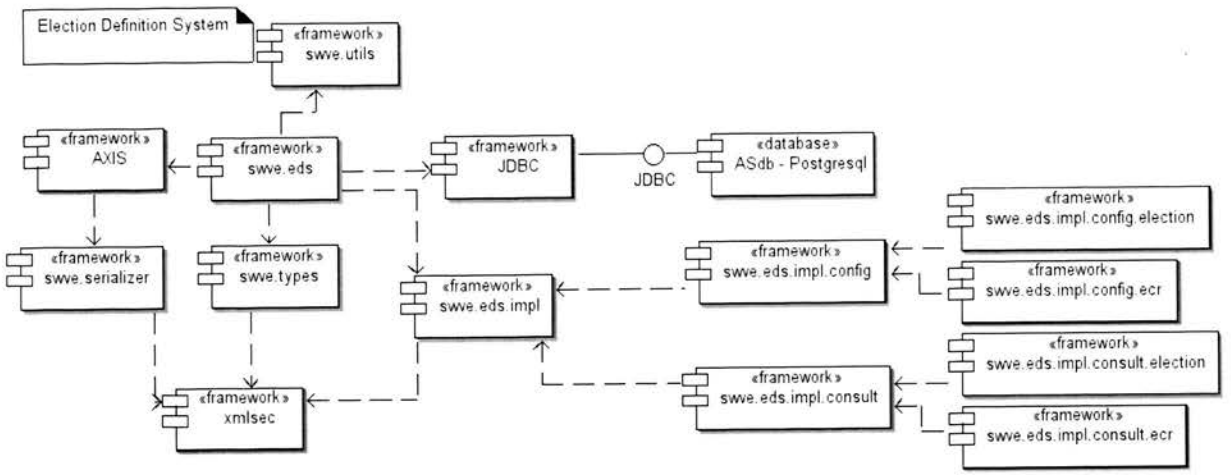


Figure 3.18: Diagrama de Componentes do Election Definition System

3.11.6 Máquina Cliente

Aplicação de Voto

Representa a aplicação cliente do Votante. Esta é executada na Máquina Cliente.
A Máquina Cliente tem para além dos componentes apresentados anteriormente os seguintes:

- swing - Este componente é a API swing.
- swve.as - Este componente é um package Java, tipo kernel do serviço. É em volta deste package que se desenvolve a implementação do cliente e do servidor relativamente ao serviço *AS*.
- swve.bs - Este componente é um package Java, tipo kernel do serviço. É em volta deste package que se desenvolve a implementação do cliente e do servidor relativamente ao serviço *BS*.
- swve.eds - Este componente é um package Java, tipo kernel do serviço. É em volta deste package que se desenvolve a implementação do cliente e do servidor relativamente ao serviço *EDS*.
- VoterApplication - Este componente é a aplicação cliente de votação.

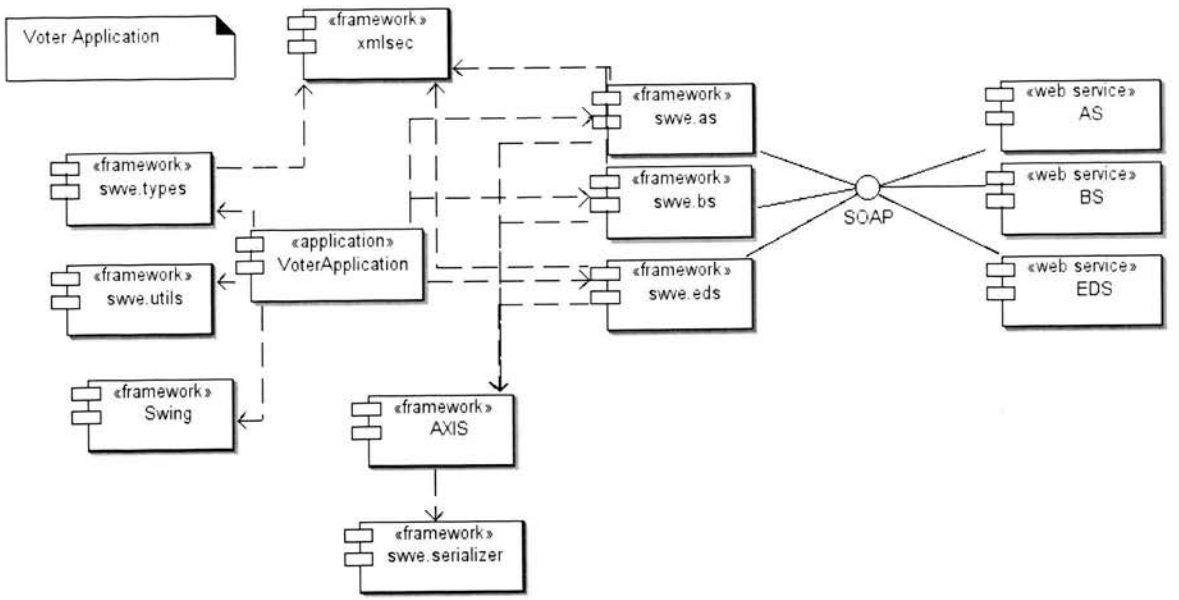


Figure 3.19: Diagrama de Componentes da Aplicação de Voto

Aplicação de Contagem

Representa a aplicação de contagem. Esta é executada na Máquina Cliente.
A Máquina Cliente tem para além dos componentes apresentados anteriormente os seguintes:

- swing - Este componente é a API swing.
- swve.vc - Este componente é um package Java, tipo kernel do serviço. É em volta deste package que se desenvolve a implementação do cliente e do servidor relativamente ao serviço *VC*.
- swve.eds - Este componente é um package Java, tipo kernel do serviço. É em volta deste package que se desenvolve a implementação do cliente e do servidor relativamente ao serviço *EDS*.
- CountingApplication - Este componente é a aplicação cliente de contagem.

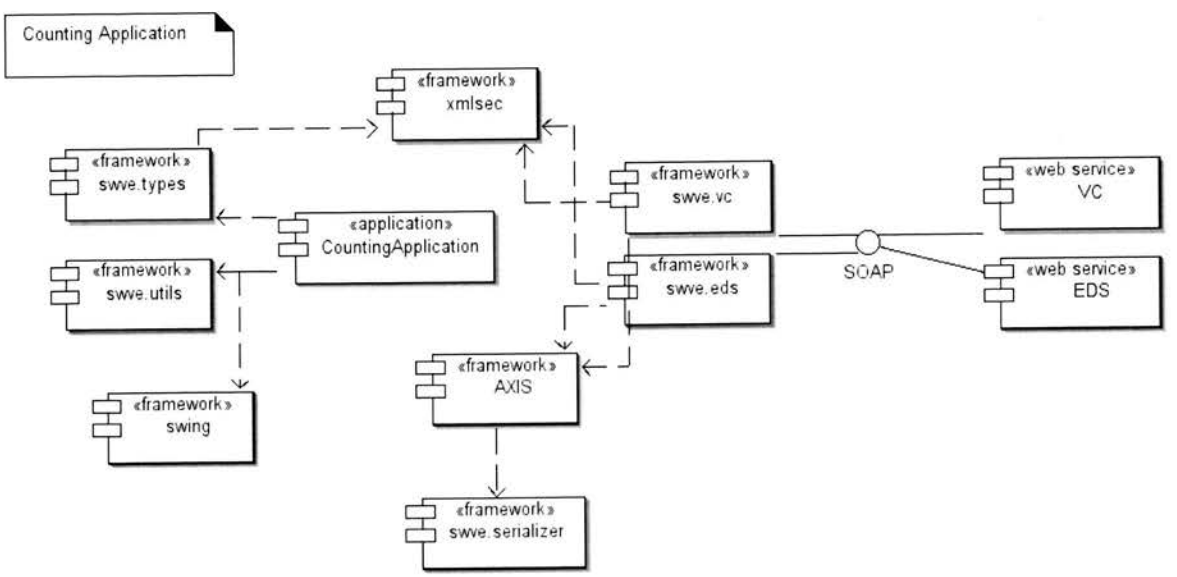


Figure 3.20: Diagrama dos Componentes da Aplicação de Contagem

Aplicação da Comissão Eleitoral

Representa a aplicação da comissão eleitoral. Esta é executada na Máquina Cliente.
A Máquina Cliente tem para além dos componentes apresentados anteriormente os seguintes:

- swing - Este componente é a API swing.
- swve.eds - Este componente é um package Java, tipo kernel do serviço. É em volta deste package que se desenvolve a implementação do cliente e do servidor relativamente ao serviço *EDS*.

ElectoralCommissionApplication - Este componente é a aplicação cliente da comissão eleitoral.

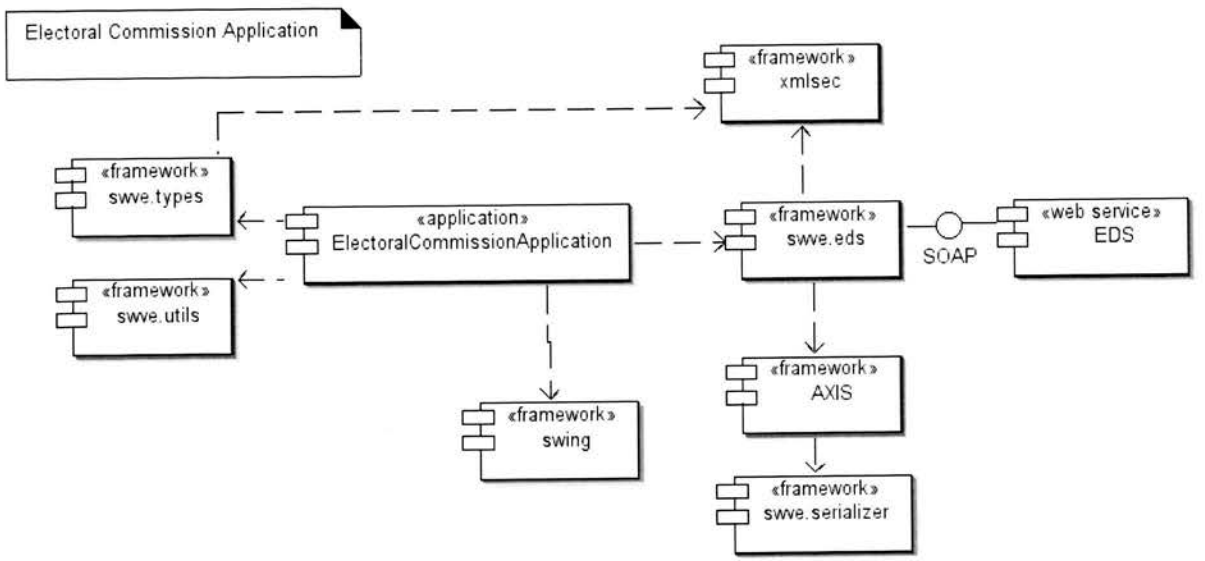


Figure 3.21: Diagrama dos Componentes da Aplicação da Comissão Eleitoral

3.12 Modelos de Classes do Domínio

Nesta secção serão apresentados os modelos de classes do domínio para os módulos existentes (módulos de votação, módulo configuração e módulo autenticaçã).

Para cada modelo será apresentado o seu respectivo diagrama de classes acompanhado com uma descrição das classes e as respectivas restrições e relações destas.

3.12.1 Módulos de Votação

Authorization System

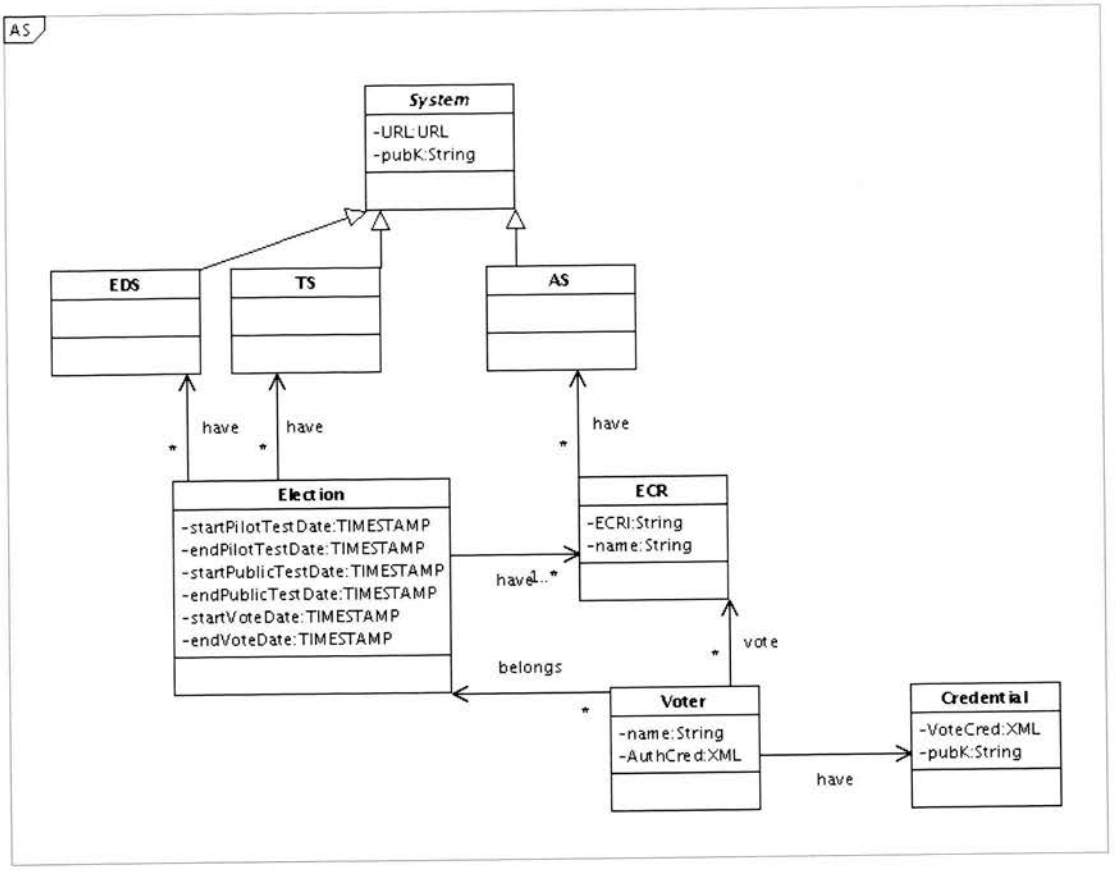


Figure 3.22: Diagrama de classes do Authorization System

Como podemos ver no diagrama de classes do *Authorization System* existe as seguintes classes:

- *Election*
- *ECR(Círculo Eleitoral)*
- *Voter*
- *System*
  - *EDS*
  - *AS*
  - *TS*
- *Credential*

A continuação serão apresentados para cada classe os seus atributos, restrições e relações como se tinha referido anteriormente.

A classe *Election* pretende representar uma eleição. Os atributos, relações e restrições desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
startPilotTestDate	Data de início da Votação de Teste Piloto
endPilotTestDate	Data de fim da Votação de Teste Piloto
startPublicTestDate	Data de início da Votação de Teste Público
endPublicTestDate	Data de fim da Votação de Teste Público
startVoteDate	Data de início da Votação Oficial
endVoteDate	Data de fim da Votação Oficial

Table 3.1: Atributos da classe Election

Atributo	Restrição
startPilotTestDate	Data posterior à data actual
endPilotTestDate	Data posterior a startPilotTestDate
startPublicTestDate	Data posterior a endPilotTestDate
endPublicTestDate	Data posterior a startPublicTestDate
startVoteDate	Data posterior a endPublicTestDate
endVoteDate	Data posterior a startVoteDate

Table 3.2: Restrições da classe Election

Classe associada	Descrição
ECR	Uma Eleição tem vários Circulos Eleitorais
TS	Uma Eleição tem um Trust System associado
Voter	Uma Eleição tem vários Eleitores

Table 3.3: Relações da classe Election

A classe *ECR* pretende representar um Círculo Eleitoral pertencente a uma eleição. Os atributos, relações e restrições desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
ECRI	Identificador do Circulo Eleitoral
name	Designação do Circulo Eleitoral

Table 3.4: Atributos da classe ECR

Atributo	Restrição
ECRI	Tem que estar assinado pelo EDS

Table 3.5: Restrições da classe ECR

Classe associada	Descrição
Election	Um Circulo Eleitoral pertence a uma Eleição
Voter	Vários eleitores votam num Circulo Eleitoral
AS	Um Circulo Eleitoral tem um Authorization System

Table 3.6: Relações da classe ECR

A classe *Voter* pretende representar um Eleitor da eleição. Os atributos e relações desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
name	Nome do Eleitor
AuthCred	Credencial de eleitor

Table 3.7: Atributos da classe Voter

Classe associada	Descrição
Election	Um Eleitor de uma Eleição pertence apenas a uma eleição
ECR	Um Eleitor de uma Eleição pertence apenas a um Círculo Eleitoral
Credential	Um Eleitor de uma Eleição tem uma credencial de votação

Table 3.8: Relações da classe Voter

A classe *System* pretende representar as características comuns entre os Sistemas. Os atributos e restrições desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
URL	Endereço do Sistema
pubK	Chave Pública do Sistema

Table 3.9: Atributos da classe System

Atributo	Restrição
URL	Tem que ser um url absoluto

Table 3.10: Restrições da classe System

A classe *EDS* pretende representar o *Election Definition System*. As relações desta classe serão apresentadas na seguinte tabela.

classe associada	Descrição
Election	Um EDS pode suportar várias Eleições

Table 3.11: Relações da classe EDS

A classe *AS* pretende representar o *Authorization System*. As relações desta classe serão apresentadas na seguinte tabela.

classe associada	Descrição
ECR	Um AS pode suportar várias Círculos Eleitorais

Table 3.12: Relações da classe AS

A classe *TS* pretende representar o *Trust System*. As relações desta classe serão apresentadas na seguinte tabela.

classe associada	Descrição
Election	Um TS suporta várias Eleições

Table 3.13: Relações da classe TS

A classe *Credential* pretende representar a credencial de votaçao. Os atributos, restrições e relações desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
pubK	Chave Pública do Eleitor
VoteCred	Credencial de votação do Eleitor

Table 3.14: Atributos da classe Credential

Atributo	Restrição
VoteCred	Tem que estar assinada pelo AS

Table 3.15: Restrições da classe Credential

Classe associada	Descrição
Voter	Um Eleitor de uma Eleição tem uma Credencial de Votação

Table 3.16: Relações da classe Credential

Ballot System

Ballot System

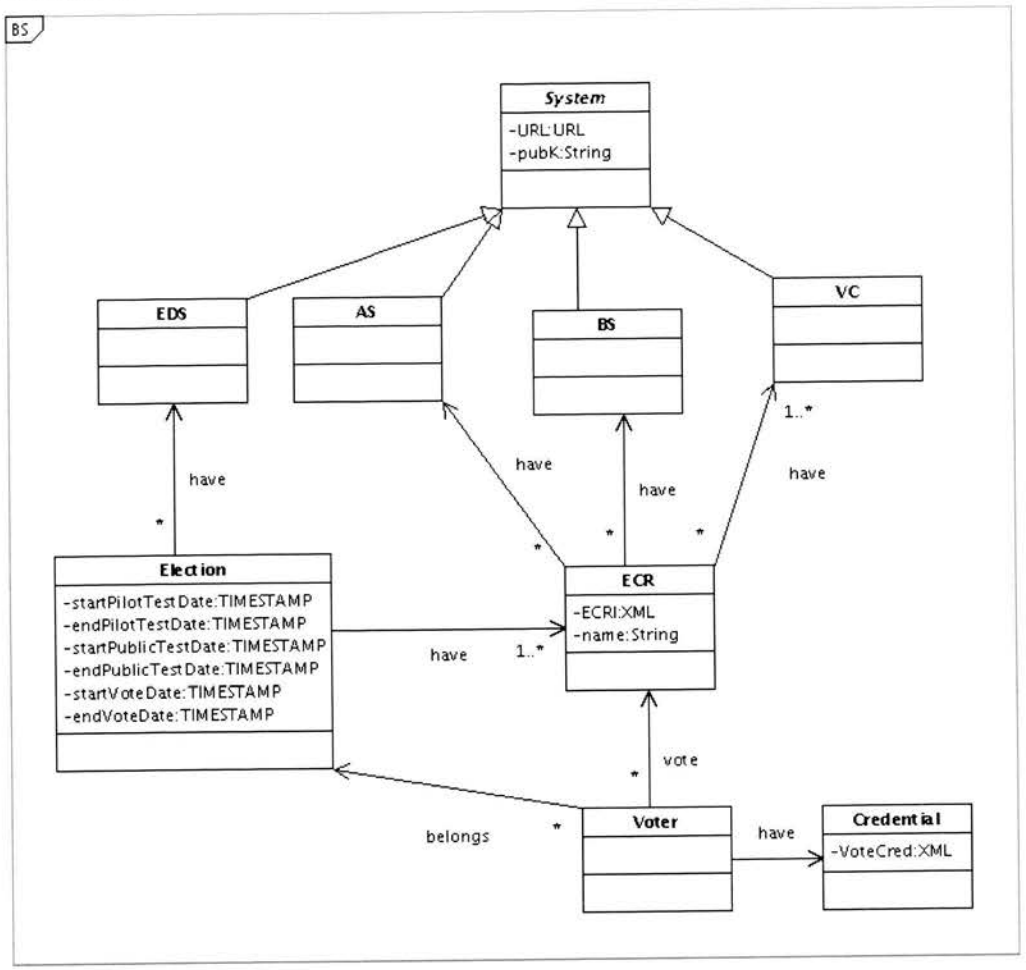


Figure 3.23: Diagrama de classes do Ballot System

Como podemos ver no diagrama de classes do *Ballot System* existe as seguintes classes:

- *Election*
- *ECR(Círculo Eleitoral)*
- *Voter*
- *System*
  - *EDS*
  - *AS*
  - *BS*
  - *VC*

A classe *Election* pretende representar uma eleição. Os atributos, relações e restrições desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
startPilotTestDate	Data de início da Votação de Teste Piloto
endPilotTestDate	Data de fim da Votação de Teste Piloto
startPublicTestDate	Data de início da Votação de Teste Público
endPublicTestDate	Data de fim da Votação de Teste Público
startVoteDate	Data de início da Votação Oficial
endVoteDate	Data de fim da Votação Oficial

Table 3.17: Atributos da classe Election

Atributo	Restrição
startPilotTestDate	Data posterior à data actual
endPilotTestDate	Data posterior a startPilotTestDate
startPublicTestDate	Data posterior a endPilotTestDate
endPublicTestDate	Data posterior a startPublicTestDate
startVoteDate	Data posterior a endPublicTestDate
endVoteDate	Data posterior a startVoteDate

Table 3.18: Restrições da classe Election

Classe	Descrição
ECR	Uma Eleição tem vários Circulos Eleitorais
Voter	Uma Eleição tem vários Eleitores

Table 3.19: Relações da classe Election

A classe *ECR* pretende representar um Círculo Eleitoral pertencente a uma eleição. Os atributos, relações e restrições desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
ECRI	Identificador do Circulo Eleitoral
name	Designação do Circulo Eleitoral

Table 3.20: Atributos da classe ECR

Atributo	Restrição
ECRI	Tem que estar assinado pelo EDS

Table 3.21: Restrições da classe ECR

Classe associada	Descrição
Election	Um Circulo Eleitoral pertence a uma Eleição
Voter	Vários eleitores votam num Circulo Eleitoral
candidate	Um Circulo Eleitoral tem vários candidatos
AS	Um Circulo Eleitoral tem um Authorization System
BS	Um Circulo Eleitoral tem um Ballot System
VC	Um Circulo Eleitoral tem vários Vote Collectors

Table 3.22: Relações da classe ECR

A classe *Voter* pretende representar um Eleitor da eleição. AS relações desta classe serão apresentados na seguinte tabela.

Classe associada	Descrição
Election	Um Eleitor de uma Eleição pertence apenas a uma eleição
ECR	Um Eleitor de uma Eleição pertence apenas a um Círculo Eleitoral
Credential	Um Eleitor de uma Eleição tem uma credencial de votação

Table 3.23: Relações da classe Voter

A classe *System* pretende representar as características comuns entre os Sistemas. Os atributos e restrições desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
URL	Endereço do Sistema
pubK	Chave Pública do Sistema

Table 3.24: Atributos da classe System

Atributo	Restrição
URL	Tem que ser um url absoluto

Table 3.25: Restrições da classe System

A classe *EDS* pretende representar o *Election Definition System*. As relações desta classe serão apresentadas na seguinte tabela.

classe associada	Descrição
Election	Um EDS pode suportar várias Eleições

Table 3.26: Relações da classe EDS

A classe *AS* pretende representar o *Authorization System*. As relações desta classe serão apresentadas na seguinte tabela.

classe associada	Descrição
ECR	Um AS pode suportar várias Círculos Eleitorais

Table 3.27: Relações da classe AS

A classe *BS* pretende representar o *Ballot System*. As relações desta classe serão apresentadas na seguinte tabela.

classe associada	Descrição
ECR	Um BS suporta várias Círculos Eleitorais

Table 3.28: Relações da classe BS

A classe *VC* pretende representar o *Vote Collector*. As relações desta classe serão apresentadas na seguinte tabela.

classe associada	Descrição
ECR	Um VC suporta vários Círculos Eleitorais

Table 3.29: Relações da classe VC

Vote Collector

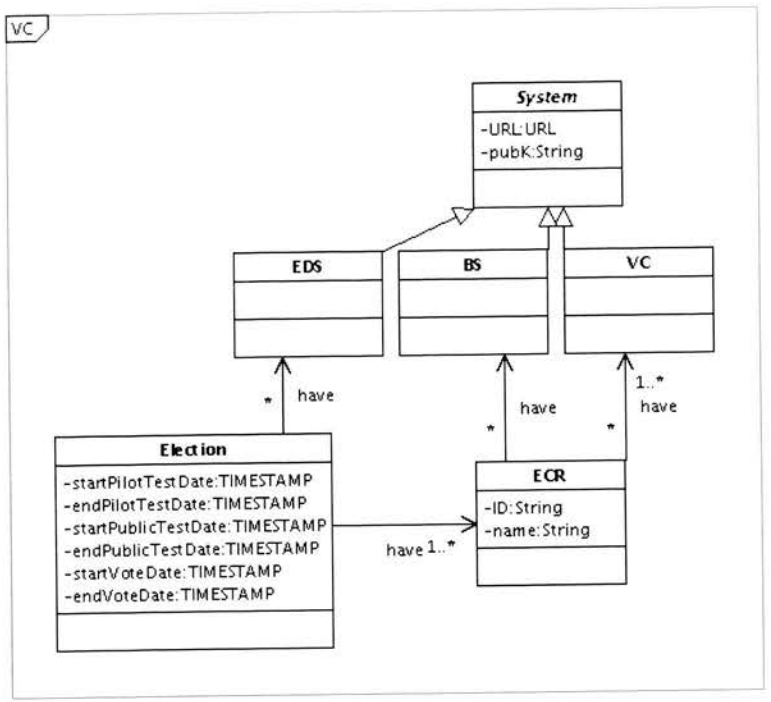


Figure 3.24: Diagrama de classes do Vote Collector

Como podemos ver no diagrama de classes do *Vote Collector* existe as seguintes classes:

- *Election*
- *ECR(Círculo Eleitoral)*
- *System*
  - *EDS*
  - *BS*
  - *VC*

A classe *Election* pretende representar uma eleição. Os atributos, relações e restrições desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
startPilotTestDate	Data de início da Votação de Teste Piloto
endPilotTestDate	Data de fim da Votação de Teste Piloto
startPublicTestDate	Data de início da Votação de Teste Público
endPublicTestDate	Data de fim da Votação de Teste Público
startVoteDate	Data de início da Votação Oficial
endVoteDate	Data de fim da Votação Oficial

Table 3.30: Atributos da classe Election

Classe associada	Descrição
ECR	Uma Eleição tem vários Circulos Eleitorais

Table 3.32: Relações da classe Election

Classe associada	Descrição
Election	Um Circulo Eleitoral pertence a uma Eleição
BS	Um Circulo Eleitoral tem um Ballot System
VC	Um Circulo Eleitoral tem vários Vote Collectors

Table 3.35: Relações da classe ECR

Atributo	Restrição
startPilotTestDate	Data posterior à data actual
endPilotTestDate	Data posterior a startPilotTestDate
startPublicTestDate	Data posterior a endPilotTestDate
endPublicTestDate	Data posterior a startPublicTestDate
startVoteDate	Data posterior a endPublicTestDate
endVoteDate	Data posterior a startVoteDate

Table 3.31: Restrições da classe Election

A classe *ECR* pretende representar um Círculo Eleitoral pertencente a uma eleição. Os atributos, relações e restrições desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
ECRI	Identificador do Circulo Eleitoral
name	Designação do Circulo Eleitoral

Table 3.33: Atributos da classe ECR

Atributo	Restrição
ECRI	Tem que estar assinado pelo EDS

Table 3.34: Restrições da classe ECR

A classe *System* pretende representar as características comuns entre os Sistemas. Os atributos e restrições desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
URL	Endereço do Sistema
pubK	Chave Pública do Sistema

Table 3.36: Atributos da classe System

Atributo	Restrição
URL	Tem que ser um url absoluto

Table 3.37: Restrições da classe System

A classe *EDS* pretende representar o *Election Definition System*. As relações desta classe serão apresentadas na seguinte tabela.

classe associada	Descrição
Election	Um EDS pode suportar várias Eleições

Table 3.38: Relações da classe EDS

A classe *BS* pretende representar o *Ballot System*. As relações desta classe serão apresentadas na seguinte tabela.

classe associada	Descrição
ECR	Um BS suporta várias Círculos Eleitorais

Table 3.39: Relações da classe BS

A classe *VC* pretende representar o *Vote Collector*. As relações desta classe serão apresentadas na seguinte tabela.

classe associada	Descrição
ECR	Um VC suporta vários Círculos Eleitorais

Table 3.40: Relações da classe VC

3.12.2 Módulo de Configuração

Election Definition System

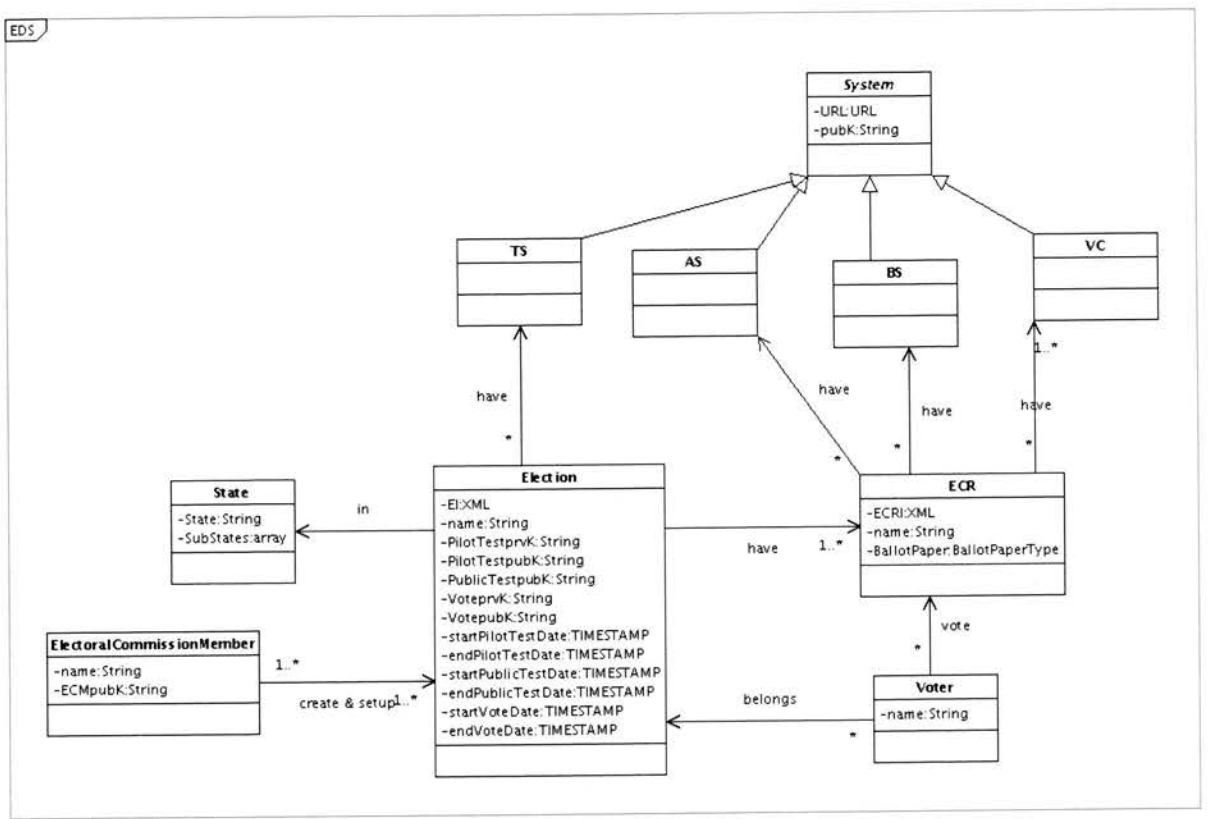


Figure 3.25: Diagrama de classes do Election Definition System

Como podemos ver no diagrama de classes do *Election Definition System* existe as seguintes classes:

- *Election*
- *ECR(Círculo Eleitoral)*
- *ElectoralCommissionMember*
- *Voter*
- *State*
- *System*

Classe associada	Descrição
ECR	Uma Eleição tem vários Circulos Eleitorais
ElectoralCommissionMember	Uma Eleição pode ser gerida por vários membros da Comissão Eleitoral
State	Uma Eleição pode estar num Estado
TS	Uma Eleição tem um Trust System associado
Voter	Uma Eleição tem vários Eleitores

Table 3.43: Relações da classe Election

- *AS*
- *BS*
- *VC*
- *TS*

A classe *Election* pretende representar uma eleição. Os atributos, relações e restrições desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
EI	Identificador da Eleição
name	Designação da Eleição
PilotTestprvK	Chave Privada da Votação de Teste Piloto
PilotTestpubK	Chave Pública da Votação de Teste Piloto
PublicTestpubK	Chave Pública da Votação de Teste Público
VoteprvK	Chave Privada da Votação Oficial
VotepubK	Chave Pública da Votação Oficial
startPilotTestDate	Data de início da Votação de Teste Piloto
endPilotTestDate	Data de fim da Votação de Teste Piloto
startPublicTestDate	Data de início da Votação de Teste Público
endPublicTestDate	Data de fim da Votação de Teste Público
startVoteDate	Data de início da Votação Oficial
endVoteDate	Data de fim da Votação Oficial

Table 3.41: Atributos da classe Election

Atributo	Restrição
EI	Tem que estar assinado pelo EDS
startPilotTestDate	Data posterior à data actual
endPilotTestDate	Data posterior a startPilotTestDate
startPublicTestDate	Data posterior a endPilotTestDate
endPublicTestDate	Data posterior a startPublicTestDate
startVoteDate	Data posterior a endPublicTestDate
endVoteDate	Data posterior a startVoteDate

Table 3.42: Restrições da classe Election

A classe *ECR* pretende representar um Círculo Eleitoral pertencente a uma eleição. Os atributos, relações e restrições desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
ECRI	Identificador do Círculo Eleitoral
name	Designação do Circulo Eleitoral
BallotPaper	Boletim de voto associado ao Circulo Eleitoral

Table 3.44: Atributos da classe ECR

Atributo	Restrição
ECRI	Tem que estar assinado pelo EDS
BallotPaper	Tem que estar assinado pelo EDS

Table 3.45: Restrições da classe ECR

Classe associada	Descrição
Election	Um Círculo Eleitoral pertence a uma Eleição
Voter	Vários eleitores votam num Circulo Eleitoral
AS	Um Circulo Eleitoral tem um Authorization System
BS	Um Circulo Eleitoral tem um Ballot System
VC	Um Circulo Eleitoral tem vários Vote Collectors

Table 3.46: Relações da classe ECR

A classe *ElectoralCommissionMember* pretende representar um membro da Comissão Eleitoral, responsavel por gerir a eleição a que pertence. Os atributos, e relações desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
name	Nome do membro da comissão eleitoral
ECMpubK	Chave pública do membro da commissão eleitoral(tambem funciona como identificador)

Table 3.47: Atributos da classe ElectoralCommissionMember

Classe associada	Descrição
Election	Um membro de uma comissão eleitoral, só pode gerir uma eleição

Table 3.48: Relações da classe ElectoralCommissionMember

A classe *Voter* pretende representar um Eleitor da eleição. Os atributos e relações desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
name	Nome do Eleitor

Table 3.49: Atributos da classe Voter

Classe associada	Descrição
Election	Um Eleitor de uma Eleição pertence apenas a uma eleição
ECR	Um Eleitor de uma Eleição pertence apenas a um Circulo Eleitoral

Table 3.50: Relações da classe Voter

A classe *State* pretende representar o estado de uma eleição. Os atributos, restrições e relações desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
State	Estado de uma Eleição
SubStates	SubEstados de uma Eleição

Table 3.51: Atributos da classe State

Atributo	Restrição
State	Tem que pertencer a uma lista de estados possíveis
SubStates	Tem que pertencer a uma lista de subestados possíveis

Table 3.52: Restrições da classe State

Classe associada	Descrição
Election	Um Estado está associado a uma Eleição

Table 3.53: Relações da classe State

A classe *System* pretende representar as características comuns entre os Sistemas. Os atributos e restrições desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
URL	Endereço do Sistema
pubK	Chave Pública do Sistema

Table 3.54: Atributos da classe System

Atributo	Restrição
URL	Tem que ser um url absoluto

Table 3.55: Restrições da classe System

A classe *AS* pretende representar o *Authorization System*. As relações desta classe serão apresentadas na seguinte tabela.

classe associada	Descrição
ECR	Um AS pode suportar várias Círculos Eleitorais

Table 3.56: Relações da classe AS

A classe *BS* pretende representar o *Ballot System*. As relações desta classe serão apresentadas na seguinte tabela.

classe associada	Descrição
ECR	Um BS suporta várias Círculos Eleitorais

Table 3.57: Relações da classe BS

A classe *VC* pretende representar o *Vote Collector*. As relações desta classe serão apresentadas na seguinte tabela.

classe associada	Descrição
ECR	Um VC suporta vários Círculos Eleitorais

Table 3.58: Relações da classe VC

A classe *TS* pretende representar o *Trust System*. As relações desta classe serão apresentadas na seguinte tabela.

Classe associada	Descrição
Election	Um TS suporta várias Eleições

Table 3.59: Relações da classe TS

3.12.3 Módulo de Autenticação

Trust System

Como podemos ver no diagrama de classes do *Trust System* existe as seguintes classes:

- *Voter*
- *System*
  - *TS*

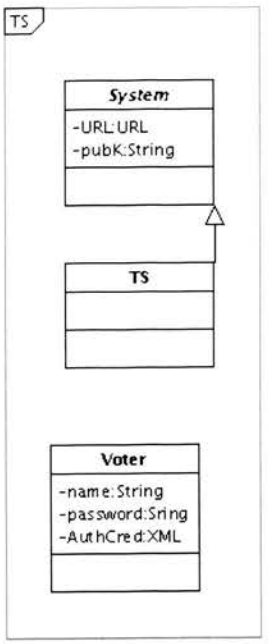


Figure 3.26: Diagrama de classes do Trust System

Nome	Restrição
AuthCred	Tem que estar assinada pelo TS

Table 3.61: Restrições da classe Voter

A classe *Voter* pretende representar um Eleitor da eleição. Os atributos e restrições desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
name	Nome do Eleitor
password	SHA1 da password do Eleitor
AuthCred	credencial de eleitor do Eleitor

Table 3.60: Atributos da classe Voter

A classe *System* pretende representar as características comuns entre os Sistemas. Os atributos e restrições desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
URL	Endereço do Sistema
pubK	Chave Pública do Sistema

Table 3.62: Atributos da classe System

Atributo	Restrição
URL	Tem que ser um url absoluto

Table 3.63: Restrições da classe System

A classe *TS* pretende representar o *Trust System*.

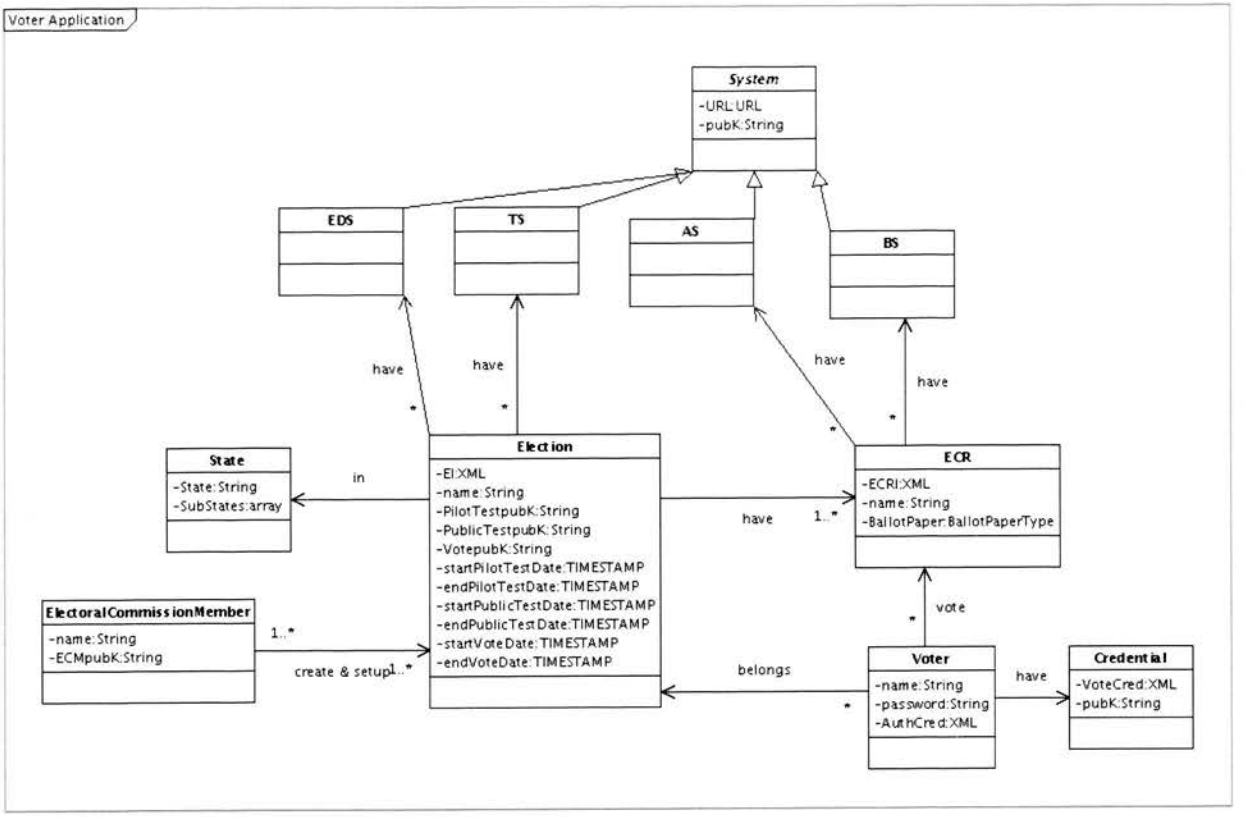


Figure 3.27: Diagrama de classes da aplicação de voto

3.12.4 Módulos de Aplicação

Voter Application

Como podemos ver no diagrama de classes do *Voter Application* existe as seguintes classes:

- *Election*
- *ECR*
- *ElectoralCommissionMember*
- *Voter*
- *State*
- *System*
  - *EDS*
  - *AS*
  - *BS*
  - *TS*
- *Credential*

A classe *Election* pretende representar uma eleição. Os atributos, relações e restrições desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
EI	Identificador da Eleição
name	Designação da Eleição
PublicTestpubK	Chave Pública da Votação de Teste Público
PilotTestpubK	Chave Pública da Votação de Teste Piloto
VotepubK	Chave Pública da Votação Oficial
startPilotTestDate	Data de início da Votação de Teste Piloto
endPilotTestDate	Data de fim da Votação de Teste Piloto
startPublicTestDate	Data de início da Votação de Teste Público
endPublicTestDate	Data de fim da Votação de teste Público
startVoteDate	Data de início da votação Oficial
endVoteDate	Data de fim da Votação Oficial

Table 3.64: Atributos da classe Election

Classe associada	Descrição
ECR	Uma Eleição tem vários Circulos Eleitorais
ElectoralCommissionMember	Uma Eleição pode ser gerida por vários membros da Comissão Eleitoral
State	Uma Eleição pode estar num Estado
TS	Uma Eleição tem um Trust System associado
Voter	Uma Eleição tem vários Eleitores

Table 3.66: Relações da classe Election

Atributo	Restrição
EI	Tem que estar assinado pelo EDS
startPilotTestDate	Data posterior à data actual
endPilotTestDate	Data posterior a startPilotTestDate
startPublicTestDate	Data posterior a endPilotTestDate
endPublicTestDate	Data posterior a startPublicTestDate
startVoteDate	Data posterior a endPublicTestDate
endVoteDate	Data posterior a startVoteDate

Table 3.65: Restrições da classe Election

A classe *ECR* pretende representar um Círculo Eleitoral pertencente a uma eleição. Os atributos, relações e restrições desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
ECRI	Identificador do Círculo Eleitoral
name	Designação do Circulo Eleitoral
BallotPaper	Boletim de voto associado ao Circulo Eleitoral

Table 3.67: Atributos da classe ECR

Atributo	Restrição
ECRI	Tem que estar assinado pelo EDS
BallotPaper	Tem que estar assinado pelo EDS

Table 3.68: Restrições da classe ECR

A classe *ElectoralCommissionMember* pretende representar um membro da Comissão Eleitoral, responsavel por gerir a eleição a que pertence. Os atributos e relações desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
name	Nome do membro da comissão eleitoral
ECMpubK	Chave pública do membro da commissão eleitoral(tambem funciona como identificador)

Table 3.70: Atributos da classe ElectoralCommissionMember

Classe associada	Descrição
Election	Um Circulo Eleitoral pertence a uma Eleição
Voter	Vários eleitores votam num Circulo Eleitoral
Candidate	Um Circulo Eleitoral tem vários candidatos
AS	Um Circulo Eleitoral tem um Authorization System
BS	Um Circulo Eleitoral tem um Ballot System

Table 3.69: Relações da classe ECR

Classe associada	Descrição
Election	Um membro de uma comissão eleitoral, só pode gerir uma eleição

Table 3.71: Relações da classe ElectoralCommissionMember

A classe *Voter* pretende representar um Eleitor da eleição. Os atributos, restrições e relações desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
name	Nome do Eleitor
password	Password do Eleitor
AuthCred	Credencial de autorização

Table 3.72: Atributos da classe Voter

Atributo	Restrição
AuthCred	Tem que estar assinada pelo TS

Table 3.73: Restrições da classe Voter

Classe associada	Descrição
Election	Um Eleitor de uma Eleição pertence apenas a uma eleição
ECR	Um Eleitor de uma Eleição pertence apenas a um Circulo Eleitoral
Credential	Um Eleitor de uma Eleição tem apenas uma credencial

Table 3.74: Relações da classe Voter

A classe *State* pretende representar o estado de uma eleição. Os atributos, restrições e relações desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
State	Estado de uma Eleição
SubStates	SubEstados de uma Eleição

Table 3.75: Atributos da classe State

Atributo	Restrição
State	Tem que pertencer a uma lista de estados possíveis
SubStates	Tem que pertencer a uma lista de subestados possíveis

Table 3.76: Restrições da classe State

Classe associada	Descrição
Election	Um Estado está associado a uma Eleição

Table 3.77: Relações da classe State

A classe *System* pretende representar as características comuns entre os Sistemas. Os atributos e restrições desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
URL	Endereço do Sistema
pubK	Chave Pública do Sistema

Table 3.78: Atributos da classe System

Atributo	Restrição
URL	Tem que ser um url absoluto

Table 3.79: Restrições da classe System

A classe *EDS* pretende representar o *Election Definition System*. As relações desta classe serão apresentadas na seguinte tabela.

Classe associada	Descrição
Election	Um EDS pode suportar várias Eleições

Table 3.80: Relações da classe EDS

A classe *AS* pretende representar o *Authorization System*. As relações desta classe serão apresentadas na seguinte tabela.

classe associada	Descrição
ECR	Um AS pode suportar várias Círculos Eleitorais

Table 3.81: Relações da classe AS

A classe *BS* pretende representar o *Ballot System*. As relações desta classe serão apresentadas na seguinte tabela.

classe associada	Descrição
ECR	Um BS suporta várias Círculos Eleitorais

Table 3.82: Relações da classe BS

A classe *TS* pretende representar o *Trust System*. As relações desta classe serão apresentadas na seguinte tabela.

Classe associada	Descrição
Election	Um TS suporta várias Eleições

Table 3.83: Relações da classe TS

A classe *Credential* pretende representar a credencial de votaçao. Os atributos, restrições e relações desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
pubK	Chave Pública do Eleitor
VoteCred	Credencial de votação do Eleitor

Table 3.84: Atributos da classe Credential

Atributo	Restrição
VoteCred	Tem que estar assinada pelo AS

Table 3.85: Restrições da classe Credential

Classe associada	Descrição
Voter	Um Eleitor de uma Eleição tem uma Credencial de Votação

Table 3.86: Relações da classe Credential

Electoral Commission Application

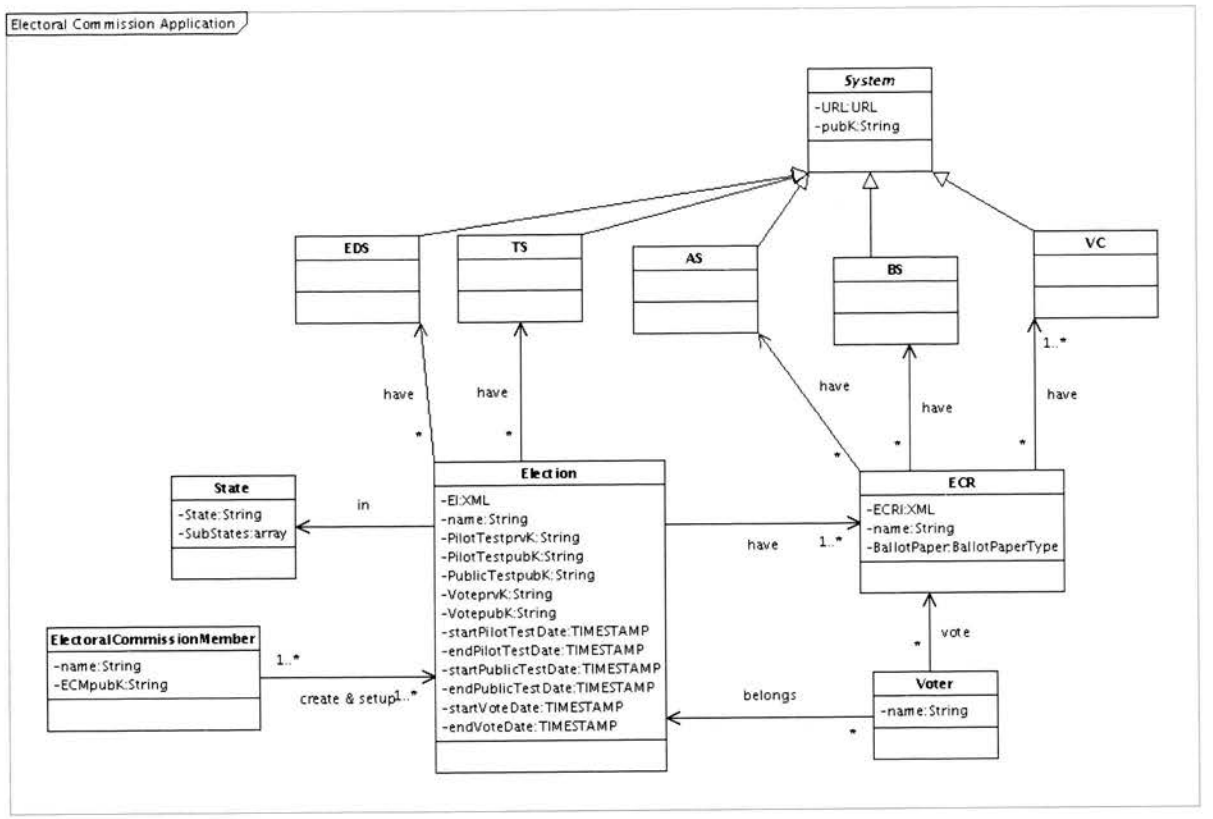


Figure 3.28: Diagrama de classes da aplicação da comissão

Como podemos ver no diagrama de classes do *Electoral Commission Application* existe as seguintes classes:

- *Election*
- *ECR*
- *ElectoralCommissionMember*
- *Voter*
- *State*
- *System*
  - *EDS*
  - *AS*
  - *BS*
  - *VC*
  - *TS*

A classe *Election* pretende representar uma eleição. Os atributos, relações e restrições desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
EI	Identificador da Eleição
name	Designação da Eleição
PilotTestprvK	Chave Privada da Votação de Teste Piloto
PilotTestpubK	Chave Pública da Votação de Teste Piloto
PublicTestpubK	Chave Pública da Votação de Teste Público
VoteprvK	Chave Privada da Votação Oficial
VotepubK	Chave Pública da Votação Oficial
startPilotTestDate	Data de início da Votação de Teste Piloto
endPilotTestDate	Data de fim da Votação de Teste Piloto
startPublicTestDate	Data de início da Votação de Teste Público
endPublicTestDate	Data de fim da Votação de Teste Público
startVoteDate	Data de início da Votação Oficial
endVoteDate	Data de fim da Votação Oficial

Table 3.87: Atributos da classe Election

Atributo	Restrição
EI	Tem que estar assinado pelo EDS
startPilotTestDate	Data posterior à data actual
endPilotTestDate	Data posterior a startPilotTestDate
startPublicTestDate	Data posterior a endPilotTestDate
endPublicTestDate	Data posterior a startPublicTestDate
startVoteDate	Data posterior a endPublicTestDate
endVoteDate	Data posterior a startVoteDate

Table 3.88: Restrições da classe Election

Classe associada	Descrição
ECR	Uma Eleição tem vários Circulos Eleitorais
ElectoralCommissionMember	Uma Eleição pode ser gerida por vários membros da Comissão Eleitoral
State	Uma Eleição pode estar num Estado
TS	Uma Eleição tem um Trust System associado
Voter	Uma Eleição tem vários Eleitores

Table 3.89: Relações da classe Election

A classe *ECR* pretende representar um Círculo Eleitoral pertencente a uma eleição. Os atributos, relações e restrições desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
ECRI	Identificador do Círculo Eleitoral
name	Designação do Circulo Eleitoral
BallotPaper	Boletim de voto associado ao Circulo Eleitoral

Table 3.90: Atributos da classe ECR

Atributo	Restrição
ECRI	Tem que estar assinado pelo EDS
BallotPaper	Tem que estar assinado pelo EDS

Table 3.91: Restrições da classe ECR

Classe associada	Descrição
Election	Um Eleitor de uma Eleição pertence apenas a uma eleição
ECR	Um Eleitor de uma Eleição pertence apenas a um Circulo Eleitoral

Table 3.96: Relações da classe Voter

Classe associada	Descrição
Election	Um Circulo Eleitoral pertence a uma Eleição
Voter	Vários eleitores votam num Circulo Eleitoral
AS	Um Circulo Eleitoral tem um Authorization System
BS	Um Círculo Eleitoral tem um Ballot System
VC	Um Círculo Eleitoral tem vários Vote Collectors

Table 3.92: Relações da classe ECR

A classe *ElectoralCommissionMember* pretende representar um membro da Comissão Eleitoral, responsavel por gerir a eleição a que pertence. Os atributos e relações desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
name	Nome do membro da comissão eleitoral
ECMpubK	Chave pública do membro da commissão eleitoral

Table 3.93: Atributos da classe ElectoralCommissionMember

Classe associada	Descrição
Election	Um membro de uma comissão eleitoral, só pode gerir uma eleição

Table 3.94: Relações da classe ElectoralCommissionMember

A classe *Voter* pretende representar um Eleitor da eleição. Os atributos e relações desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
name	Nome do Eleitor

Table 3.95: Atributos da classe Voter

A classe *State* pretende representar o estado de uma eleição. Os atributos, restrições e relações desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
State	Estado de uma Eleição
SubStates	SubEstados de uma Eleição

Table 3.97: Atributos da classe State

Atributo	Restrição
State	Tem que pertencer a uma lista de estados possíveis
SubStates	Tem que pertencer a uma lista de subestados possíveis

Table 3.98: Restrições da classe State

Classe associada	Descrição
Election	Um Estado está associado a uma Eleição

Table 3.99: Relações da classe State

A classe *System* pretende representar as características comuns entre os Sistemas. Os atributos e restrições desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
URL	Endereço do Sistema
pubK	Chave Pública do Sistema

Table 3.100: Atributos da classe System

Atributo	Restrição
URL	Tem que ser um url absoluto

Table 3.101: Restrições da classe System

A classe *EDS* pretende representar o *Election Definition System*. As relações desta classe serão apresentadas na seguinte tabela.

Classe associada	Descrição
Election	Um EDS pode suportar várias Eleições

Table 3.102: Relações da classe EDS

A classe *AS* pretende representar o *Authorization System*. As relações desta classe serão apresentadas na seguinte tabela.

classe associada	Descrição
ECR	Um AS pode suportar várias Círculos Eleitorais

Table 3.103: Relações da classe AS

A classe *BS* pretende representar o *Ballot System*. As relações desta classe serão apresentadas na seguinte tabela.

classe associada	Descrição
ECR	Um BS suporta várias Círculos Eleitorais

Table 3.104: Relações da classe BS

A classe *VC* pretende representar o *Vote Collector*. As relações desta classe serão apresentadas na seguinte tabela.

classe associada	Descrição
ECR	Um VC suporta vários Círculos Eleitorais

Table 3.105: Relações da classe VC

A classe *TS* pretende representar o *Trust System*. As relações desta classe serão apresentadas na seguinte tabela.

Classe associada	Descrição
Election	Um TS suporta várias Eleições

Table 3.106: Relações da classe TS

Counting Application

Como podemos ver no diagrama de classes do *Counting Application* existe as seguintes classes:

- *Election*
- *ECR*
- *ElectoralCommissionMember*

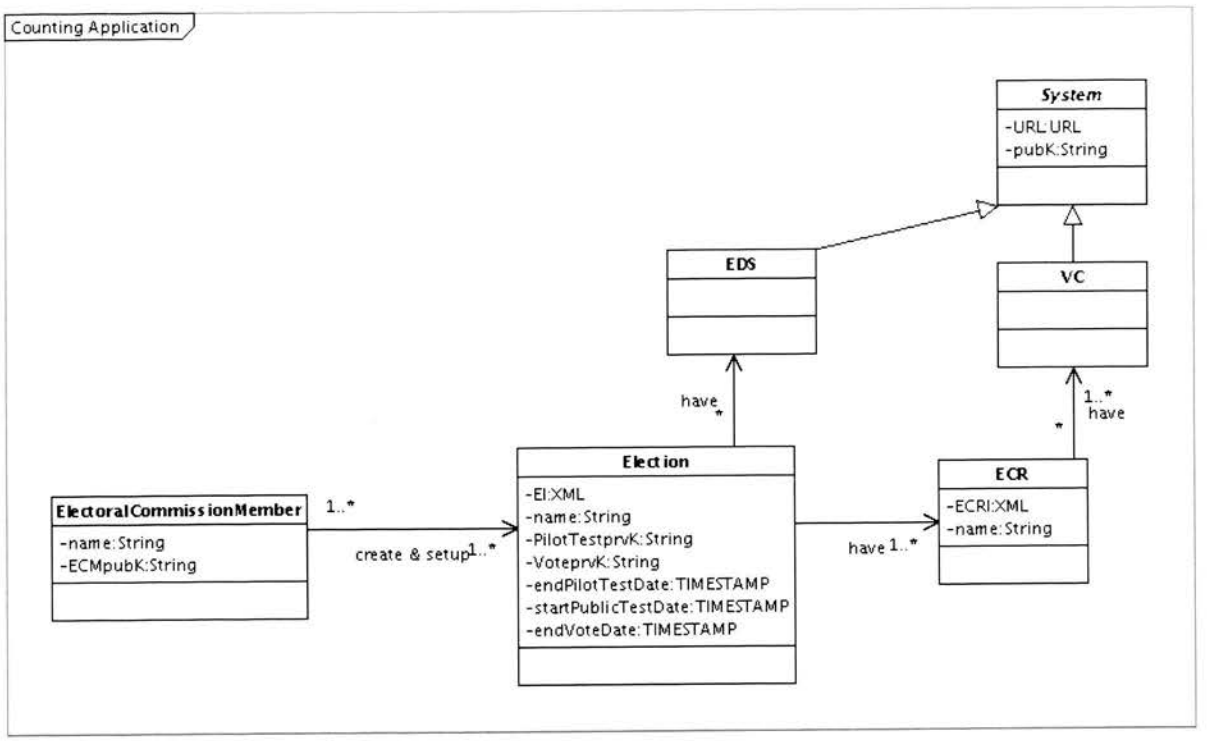


Figure 3.29: Diagrama de classes da aplicação de contagem

- *System*
  - *EDS*
  - *VC*

A classe *Election* pretende representar uma eleição. Os atributos, relações e restrições desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
EI	Identificador da Eleição
name	Designação da Eleição
PilotTestprvK	Chave Privada da Votação de Teste Piloto
VoteprvK	Chave Privada da Votação Oficial
endPilotTestDate	Data de fim da Votação de Teste Piloto
startPublicTestDate	Data de início da Votação de Teste Público
endVoteDate	Data de fim da Votação Oficial

Table 3.107: Atributos da classe Election

Atributo	Restrição
EI	Tem que estar assinado pelo EDS
startPilotTestDate	Data posterior à data actual
endPilotTestDate	Data posterior a startPilotTestDate
startPublicTestDate	Data posterior a endPilotTestDate
endPublicTestDate	Data posterior a startPublicTestDate
startVoteDate	Data posterior a endPublicTestDate
endVoteDate	Data posterior a startVoteDate

Table 3.108: Restrições da classe Election

Classe associada	Descrição
ECR	Uma Eleição tem vários Circulos Eleitorais
ElectoralCommissionMember	Uma Eleição pode ser gerida por vários membros da Comissão Eleitoral

Table 3.109: Relações da classe Election

A classe *ECR* pretende representar um Círculo Eleitoral pertencente a uma eleição. Os atributos, relações e restrições desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
ECRI	Identificador do Circulo Eleitoral
name	Designação do Circulo Eleitoral

Table 3.110: Atributos da classe ECR

Atributo	Restrição
ECRI	Tem que estar assinado pelo EDS

Table 3.111: Restrições da classe ECR

Classe associada	Descrição
Election	Um Círculo Eleitoral pertence a uma Eleição
VC	Um Círculo Eleitoral tem vários Vote Collectors

Table 3.112: Relações da classe ECR

A classe *ElectoralCommissionMember* pretende representar um membro da Comissão Eleitoral, responsável por gerir a eleição a que pertence. Os atributos e relações desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
name	Nome do membro da comissão eleitoral
ECMpubK	Chave pública do membro da commissão eleitoral

Table 3.113: Atributos da classe ElectoralCommissionMember

Classe associada	Descrição
Election	Um membro de uma comissão eleitoral, só pode gerir uma eleição

Table 3.114: Relações da classe ElectoralCommissionMember

A classe *System* pretende representar as características comuns entre os Sistemas. Os atributos e restrições desta classe serão apresentados nas seguintes tabelas.

Nome	Descrição
URL	Endereço do Sistema
pubK	Chave Pública do Sistema

Table 3.115: Atributos da classe System

Atributo	Restrição
URL	Tem que ser um url absoluto

Table 3.116: Restrições da classe System

A classe *EDS* pretende representar o *Election Definition System*. As relações desta classe serão apresentadas na seguinte tabela.

Classe associada	Descrição
Election	Um EDS pode suportar várias Eleições

Table 3.117: Relações da classe EDS

A classe *VC* pretende representar o *Vote Collector*. As relações desta classe serão apresentadas na seguinte tabela.

classe associada	Descrição
ECR	Um VC suporta vários Círculos Eleitorais

Table 3.118: Relações da classe VC

Chapter 4

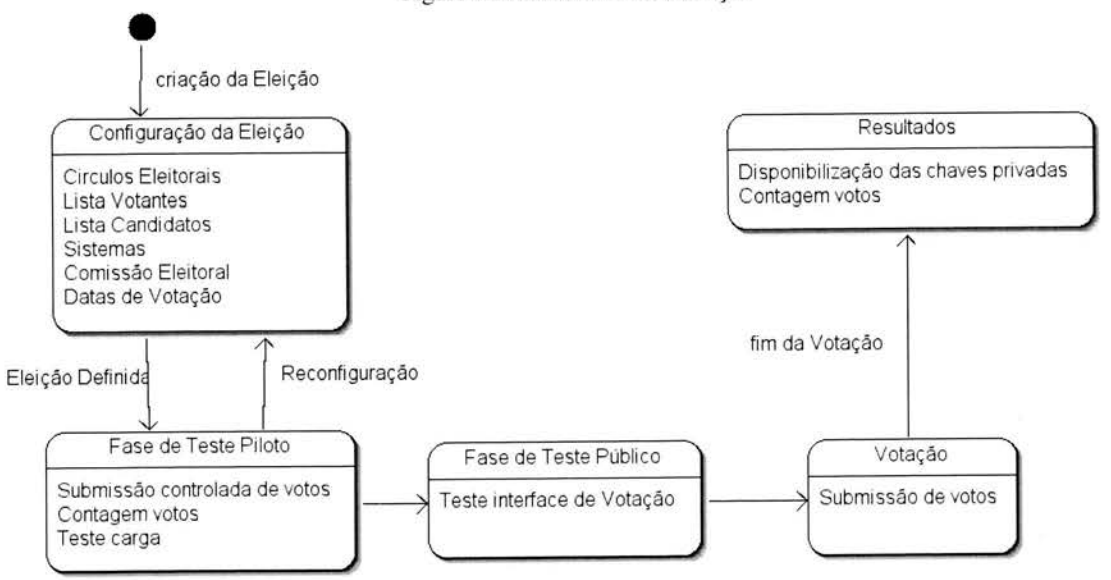
Funcionamento

4.1 Configuração da Eleição

Nestas secção serão apresentados os vários estados de uma eleição. Começaremos por apresentar o diagrama de estados 4.1 para ter uma ideia geral.

Com um sistema *Always-on* e é gerido por um EDS é possível controlarmos estado a estado a oferta de serviços. Assim será possível para uma eleição dispor dos estados apresentados na figura 4.1.

Figure 4.1: Estados de uma eleição



Num 1º estado a eleição é configurada:

- Definir Eleição
- Definir Circulo Eleitoral
- Definir membros da comissão eleitoral

Apesar de o sistema poder funcionar correctamente, no seguinte estado uma pequena lista de votantes (lista de votantes piloto) utiliza o sistema permitindo verificar se não existe nenhuma falha. Neste estado existe a possibilidade de visualizar os votos enquanto se procede à votação.

Na Fase de Teste Público, a finalidade é apresentar a eleição aos eleitores para que estes se acostumem com a votação, por exemplo o boletim de voto. Nesta fase, os votos são secretos e não são contados para evitar influências nos votantes.

No Estado Votação é o estado em que decorre a votação oficial.

No Estado Resultados a chave privada da Eleição é disponibilizada assim como os votos chifrados, sendo possível a contagem.

A continuação será apresentado o diagrama de estados mais pormenorizado e por questões de espaço e por forma a manter uma qualidade de resolução aceitável para o diagrama, decidimos separar-lo em quatro secções. Estas secções serão apresentadas a continuação com o seu respectivo diagrama de estados e daremos o devido encadeamento entre as várias secções que compõem o mesmo.

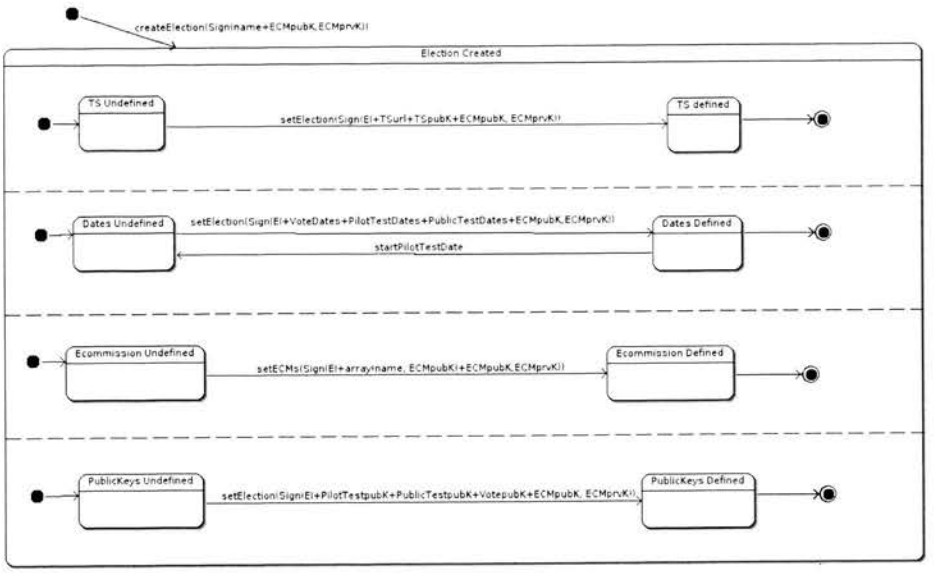


Figure 4.2: Macro estado referente a criação da eleição

Este macro estado apresentado na figura anterior, apresenta-nos o estado inicial de uma eleição. Para passar para o macro estado *Election Created* é necessário enviar a operação *createElection* e os respectivos parâmetros como podes-se ver na figura. Este estado fica completamente definido quando os sub-estados *TS, Dates, Ecommission e Publickeys* sejam definidos e para que isso seja possível terão que ser efectuadas determinadas operações.

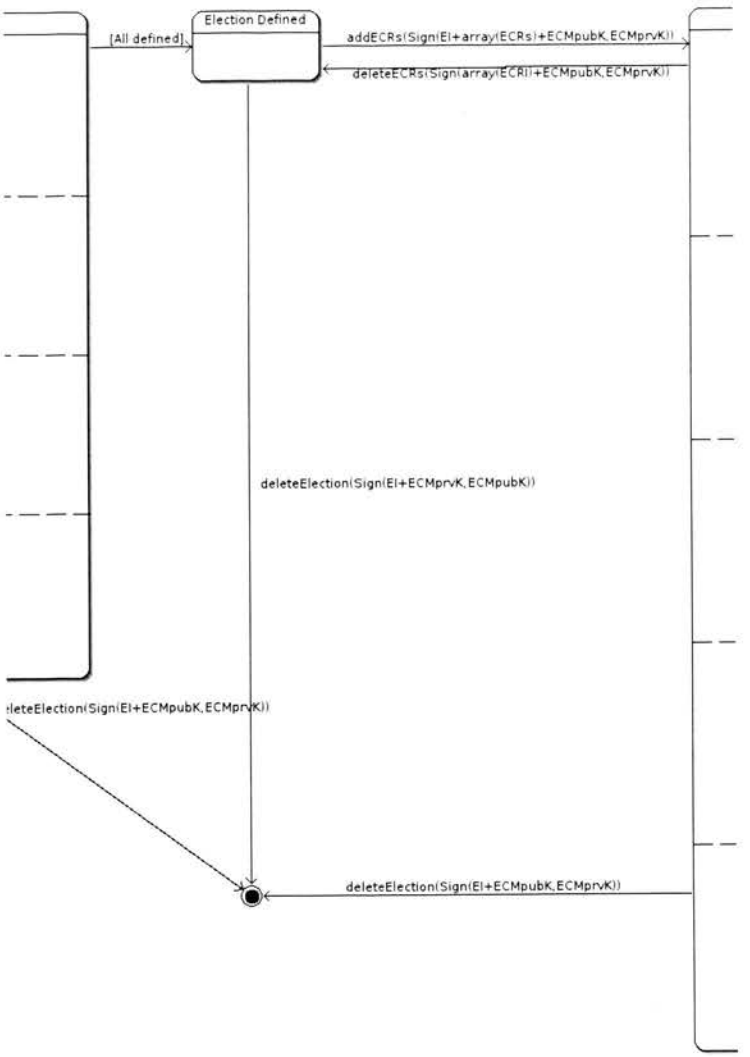


Figure 4.3: Estado intermediário entre os macros estados da eleição criada e ECR criado

Quando a eleição já tiver sido definida o sistema pode passar para um outro estado (*Election Defined*) ou a eleição pode também ser apagada desde que a operação deleteElection for requerida.

Estando no estado *Election Defined* somos capazes de criar os círculos eleitorais (*addECRs*) ou apagar a eleição (*delete-Election*).

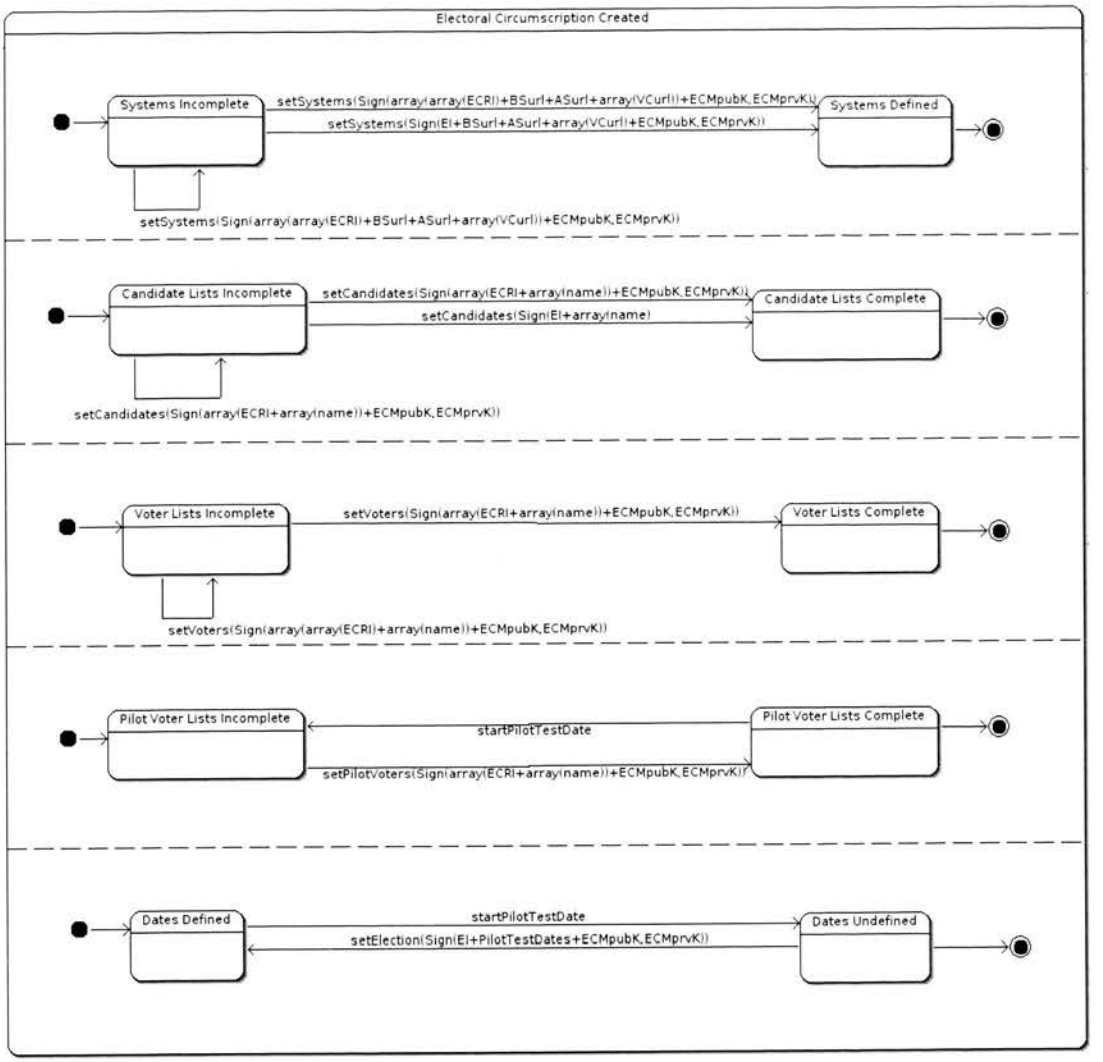


Figure 4.4: Macro estado referente a criação do ECR

O macro estado da figura anterior é onde os Círculos Eleitorais(*ECR*) são definidos. Para que este estado do sistema fique totalmente definido é necessário que a lista de votantes(Pilotos e Oficias), candidatos, as datas e a configuração dos sistemas estejam completamente definidas(é possível fazer alterações nos sub-estados que compõem o *Electoral Circumscription*).

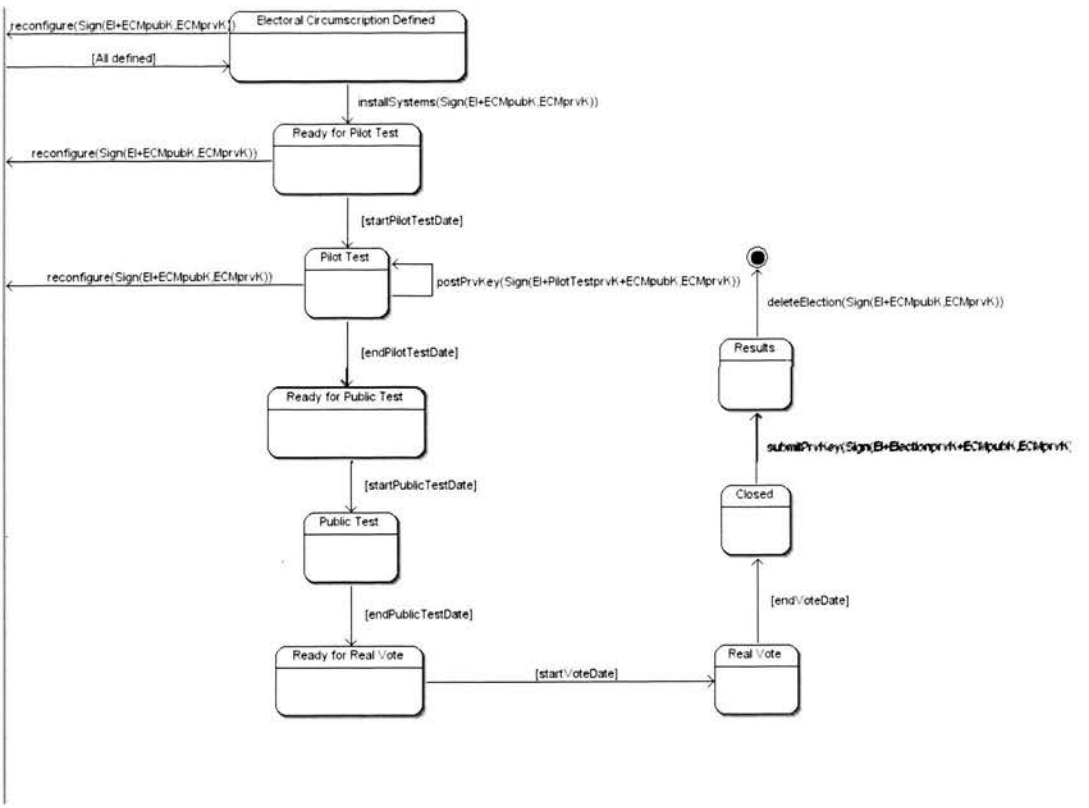


Figure 4.5: Os vários estados de testes e votação oficial de uma eleição.

Depois que o *ECR* estiver completamente defenido o sistema passa automaticamente para um novo estado(*Electoral Circumscription Defined*) sendo possível voltar ao macro estado anterior caso seja necessário alguma alteração no *ECR*.
Os estados seguintes dizem respeito a realização dos testes pilotos, públicos e a eleição oficial. Nos testes pilotos é possível fazer reconfigurações caso a comissão eleitoral assim o desejar.
No caso de finalizar-se a votação oficial o sistema passa para o estado onde serão mostrados os resultados de determinada eleição.

4.2 Funcionamento da Eleição

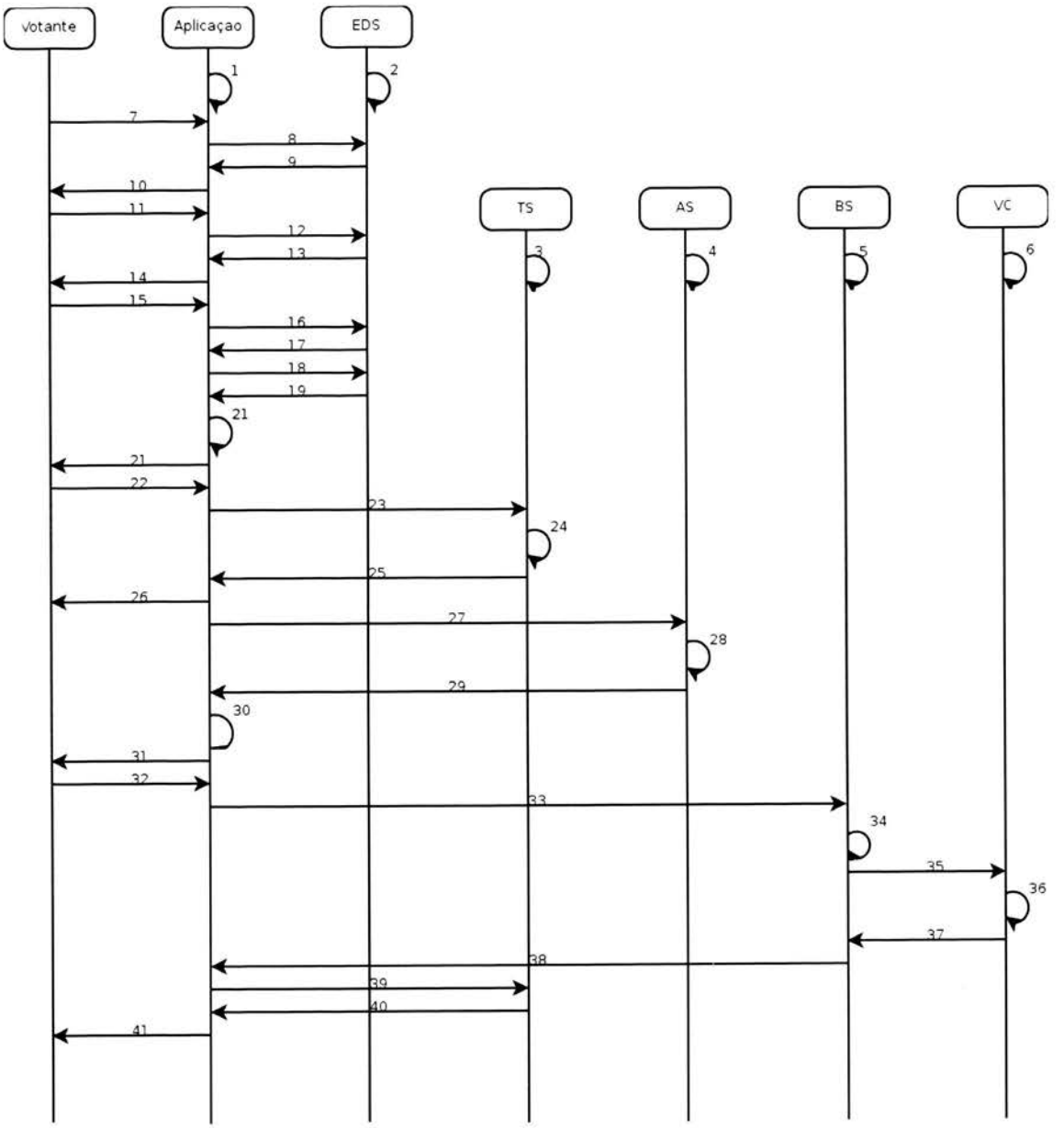


Figure 4.6: Diagrama do funcionamento da eleição.

Descrição do diagrama do funcionamento da eleição:

1. É gerada a chave pública e privada do Votante (VpubK,VprvK)
2. É gerada a chave pública e privada do EDS (Election Description System): (EDSpubK, EDSprvK)
3. É gerada a chave pública e privada do AS (Authorization System): (ASpubK, ASprvK)
4. É gerada a chave pública e privada do TS (Trust System): (TSpubK, TSprvK)
5. É gerada a chave pública e privada do BS (Ballot System) (BSpubK, BSprvK)
6. É gerada a chave pública e privada de cada VC (Vote Colector) (VCpubK, VCprvK)
7. O votante acede a interface
8. A Interface invoca o serviço getElections ao EDS

9. EDS responde com um arrayof (name, EI)
10. Interface mostra a informação obtida
11. O Votante seleciona uma eleição
12. A interface invoca o serviço getECRs (EI)
13. EDS retorna um arrayof (name+ECRI) da eleição EI
14. A interface mostra os Círculos eleitorais
15. O Votante seleciona um circulo eleitoral
16. A interface invoca o getElection (EI)
17. EDS retorna (EI, name, ASurl, ASpubK, startPilotTestDate, endPilotTestDate, startPublicTestDate, endPublicTestDate, startVoteDate, endVoteDate)
18. A interface invoca o getECR (ECRI)
19. O EDS retorna (name, ECRI, BSurl, BSpubK, TSurl, TSpubK, array (VCurl,VCpubK), array (candidate))
20. A interface guarda todos os dados retornados
21. A interface envia formulário de autenticação ao Votante
22. Votante autentica-se, enviando: user e password
23. A interface autentica o Votante no TS enviando: Ciph (username+password, TSpubK)
24. O TS desencripta e verifica se o user e password estão correctos: Cria uma credencial de autenticação que identifica o utilizador. Sendo esta credencial criada o user não pode mudar a password, enquanto durar o tempo de vida da AuthCred. AuthCred=Sign (SHA-1 (username,password)+username+endValidDate,TSprvK). CiAuthCred=Ciph (AuthCred,VoterpubK)
25. O TS retorna o certificado de autenticação (CiAuthCred)
26. Interface informa ao votante que foi autenticado
27. A interface pede uma credencial de votação ao AS para votar: Interface envia CiAuthCred + ECRI ao AS. AuthCred=DeCiph(CiAuthCred,VprvK). CiAuthCred=Ciph(AuthCred,ASpubK)
28. O AS verifica AuthCred, verifica se esse votante está habilitado para votar nesse ECRI. Cria a credencial de votação e guarda o sha1 do AuthCred associando com a credencial de votação (VoteCred=Sign (SHA-1 AuthCred+rand)+ECRI+endValidDate,ASprvK)). CiVoteCred=Ciph (VoteCred,VpubK)
29. AS retorna a credencial de votação: CiVoteCred
30. A interface apresenta um boletim de voto ao Votante.
31. Utilizador efectua a escolha do candidato
32. A Interface cria o dCiB. CiB=Ciph ((Ballot),VCpubK). VoteCred=DeCiph (CiVoteCred,VprvK). dCiB=Ciph ((CiB,VoteCred+ECI
33. A interface envia o dCiB ao BS
34. O BS desencripta o dCiB, e valida a credencial de votação. Depois assina o CiB mais a credencial de votação. (CiB,VoteCred)=DeCiph(dCiB,BSprvK), SiCiB=Sign(CiB+rand(),BSprvK)
35. Envia o SiCiB aos VCs
36. VC valida o SiCiB. Validate(SiCiB,BSpubK)
37. VC confirma ao BS a recepção do voto
38. BS confirma a interface a recepção do voto
39. A interface confirma ao AS que já votou
40. AS confirma a interface que o Votante já completou o processo de votação
41. A interface informa ao Votante que o voto foi submetido com sucesso

Chapter 5

Resultados

5.1 Instalação do Software

Na instalação do sistema operativo (*Gentoo Linux*, ver secção **??**) seguimos os procedimentos descritos na página oficial da distribuição de Linux [9].

5.1.1 Java Virtual Machine (JVM)

No presente projecto, usamos a JVM com Java 1.4, porque, apesar do Java 1.5 já estar disponível, consideramos a versão 1.4 mais estável, e consegue satifazer os requisitos pretendidos.
Na instalação do JVM, usamos a aplicação portage, executando o seguinte comando:

```
# emerge sun-jre-bin sun-jdk
```

Em seguida para utilização de encriptação e assinaturas RSA, adicionamos o *provider* bouncycastle [**?**]. Utilizamos o seguintes procedimentos:

- Download da biblioteca do *provider BouncyCastle*. Como se utilizou a Java 1.4, o *provider* tem que ser correspondente à versão do Java (no presente caso, o *provider BouncyCastle* estava na *release 1.33*), assim a biblioteca utilizada foi bcprov-jdk14-133.jar.
- Copiar bcprov-jdk14-133.jar para o directorio do *JVM*

```
# export JRE_HOME=/opt/blackdown-jre-blackdown-jre-1.4.2.03
# export JDK_HOME=/opt/blackdown-jre-blackdown-jdk-1.4.2.03
# cp bcprov-jdk14-133.jar $JRE_HOME/lib/ext
# cp bcprov-jdk14-133.jar $JDK_HOME/jre/lib/ext
```

- Modificar os ficheiros

```
$JRE_HOME/lib/security/java.security
$JDK_HOME/lib/security/java.security
```

- Adicionar o *provider BouncyCastle* na lista que aparece no ficheiro (como se pode observar no provider 6)

```
...
security.provider.1=sun.security.provider.Sun security.provider.2=com.sun.net.ssl.inte
...
```

5.1.2 Apache Tomcat

O contentor de servlets Tomcat utilizado foi o Tomcat 5.0, que é a última versão estável. Para a sua instalação utilizamos a aplicação portage, executando o seguinte comando:

```
# emerge tomcat
```

Para iniciar o tomcat executar o seguinte comando:

```
# /etc/init.d/tomcat5 start
```

Para verificar se o Tomcat está correctamente instalado e a funcionar visualizar http://localhost:8080.

5.1.3 Axis

A instalação do Axis realiza-se em dois passos, primeiro instalação da aplicação web, e o segundo instalação das bibliotecas necessárias para funcionamento do Axis e para suporte das tecnologias adjacentes.

A versão do Axis 1 utilizada é 1.4, pois é a última versão até ao momento, e consequentemente mais estável.

Na instalação da aplicação web seguimos os seguintes procedimentos:

- Fazer download do Axis (disponivel em [15]) e descompactar;
- Dos arquivos descompactados, copiar o directório *axis* que está dentro do directório *webapps* para o directório *webapps* onde está instalado o Tomcat ($TOMCAT_HOME);

Concluída a instalação, é preciso instalar as bibliotecas, para isso colocou-se as seguintes biliotecas no directório $TOMCAT_HOME/wel INF/lib (a este directório vamos passar a chamar $AXIS_LIB):

- JavaBeans Activation Framework (activation.jar, disponivel em [16])
- Apache XML Security (xmlsec.jar, disponivel em [17])

Um possível testar para verificar se o Axis está correctamente instalado é visualizar a página `http://localhost:8080/axis/happyaxis.jsp` (supondo que o servidor é local, e o Tomcat está a ser executado na porta TCP 8080). Este teste é inconclusivo, pois não permite verificar se o Axis está a funcionar correctamente, mas pode-se detectar alguns problemas.

Após a instalação do Axis e antes de fazer o teste, convém reiniciar o Tomcat.

5.1.4 Postgresql

A versão de Postgresql utilizada foi 8.1, que é a ultima versão estável (disponivel em [18]).

Para a sua instalação, usou-se a aplicação portage, executando o seguinte comando:

```
# emerge postgresql
```

O portage disponibiliza um sistema de configuração automática do postgresql que nós utilizamos

```
# emerge --config postgresql
```

Para iniciar o postgresql usamos o seguinte comando:

```
# /etc/init.d/postgresql start
```

Para configurar a criar as base de dados e utilizadores utilizamos o phppgadmin (em que tivemos que instalar o Apache2):

```
# emerge phppgadmin
# /etc/init.d/apache2 start
```

JDBC Driver

Para a aplicação java interagir com a Base de Dados, instalamos o driver *jdbc3-postgresql*. Usamos a aplicação portage, executando os comandos:

```
# emerge jdbc3-postgresql
```

Depois copiamos a biblioteca criada (pode ser criado um link simbolico) para o directório $AXIS_LIB.

A configuração da váriavel de ambiente $CLASSPATH vai ser tratada na secção .

5.1.5 Configuração das Variáveis de Ambiente

Para desenvolvimento das aplicações é necessário que a varíavel de ambiente $CLASSPATH aponte para as bibliotecas dentro do directorio $AXIS_LIB. Também pode é necessário incluir bibliotecas que desenvolvidas neste projecto (AS.jar, BS.jar, TS.jar, VC.jar e common.jar). Nesse caso é necessário adicionar essas bibliotecas no directório $AXIS_LIB.

Para automatizar o processo de configuração da váriavel $CLASSPATH usamos o seguinte script (sh):

```
#!/bin/sh
export AXIS_LIB=/opt/tomcat5/webapps/axis/WEB-INF/lib
for f in $AXIS_LIB/*.jar
do CLASSPATH=$CLASSPATH:$f
done
export CLASSPATH
```

5.1.6 Instalação de serviços

Depois de ter instalado o software (referido na secção anterior), seguir os seguintes passos, na máquina pretendida:

- Fazer download do AS.jar em http://192.168.103.23:8668/ (disponível apenas a partir do Campus da FEUP);
- Copiar o AS.jar para a directoria $TOMCAT_HOME/webapps/axis/WEB-INF/classes/;
- Extrair AS.jar

  ```
  jar -xf AS.jar
  ```

- Pubilicar o serviço web

  ```
  java org.apache.axis.wsdl.client.AdminClient $TOMCAT_HOME/webapps/axis/WEB-INF/classes
  ```

Para os restantes serviços (BS,TS,VC,EDS), o processo é identico.

5.1.7 Instalação das base de dados

Para instalar as base de dados é necessário ir a pagina do nosso projecto[23] onde esta disponibilizado o SQL das várias base de dados para tal tem que seguir este caminho Documentação > Protótipo > Concretização do Caso de Uso > Parte Dinâmica > Pré-Manual.

5.2 Utilização das Aplicações

Depois dos serviços estarem instalados, passemos à sua utilização.

5.2.1 Voter Application

- Executemos a aplicação VoterApplication (disponivel em [23])

  ```
  jar -jar VoterApplication.jar
  ```

- O primeiro painel é o login, onde se pode fazer a autenticação:

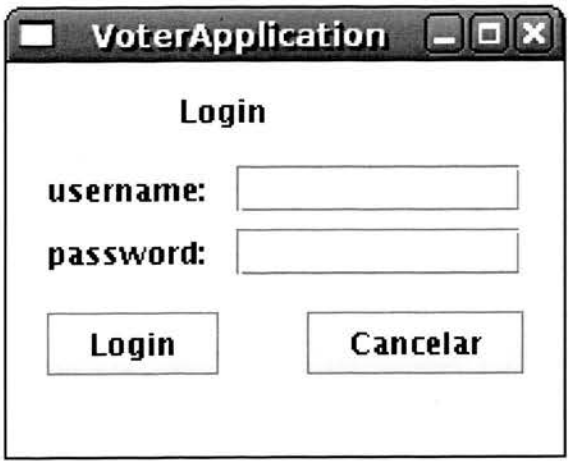


Figure 5.1: Painel gráfico da Voter Application - Login

- Depois de feito o login, é altura de votar:

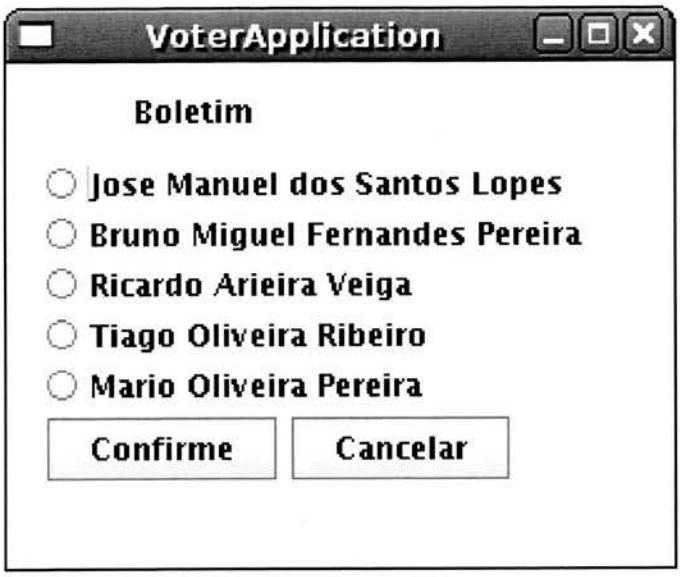


Figure 5.2: Painel gráfico da Voter Application - Boletim Voto

- Eis a confirmação da operação bem sucedida:

Figure 5.3: Painel gráfico da Voter Application - Mensagem

5.2.2 Counting Application

- Executemos a aplicação CountingApplication (disponivel em [23])

```
jar -jar CountingApplication.jar
```

- O primeiro painel tem a Eleição à qual a aplicação pretende fazer a contagem:

Figure 5.4: Painel gráfico da Counting Application: Inicio

- Depois de clicar em Contagem, a aplicação procede à contagem:

Figure 5.5: Painel gráfico da Counting Application: Contagem

- No final da contagem é apresentado um painel com os resultados da votação:

Figure 5.6: Painel gráfico da Counting Application: Resultados

* No final existe a possibilidade de submeter os resultados para o EDS:

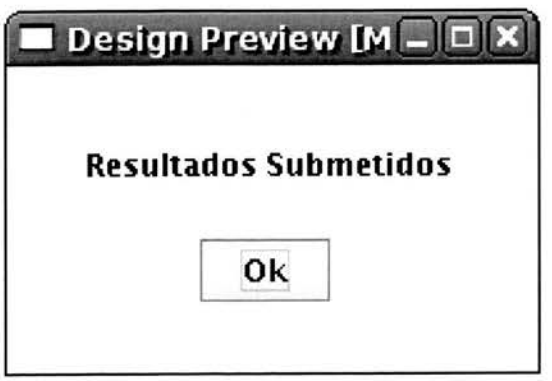


Figure 5.7: Painel gráfico da Counting Application: Submeter resultados

5.2.3 Electoral Commission Application

Esta aplicação tem vários casos de utilização (ver secção **??**), podendo alguns deles serem acedidos simultaneamente. Por isso, nesta apresentação usamos menus para acessar às funcões, e em seguida iremos mostrar apenas os paineis para cada caso de utilização.

Figure 5.8: Painel gráfico da Electoral Commission Application: Menu Eleição

Figure 5.9: Painel gráfico da Electoral Commission Application: Menu Circulo Eleitoral

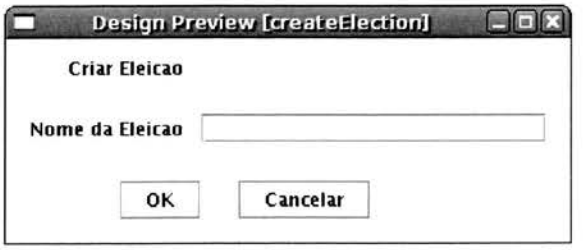


Figure 5.10: Painel gráfico da Electoral Commission Application: Criar Eleição

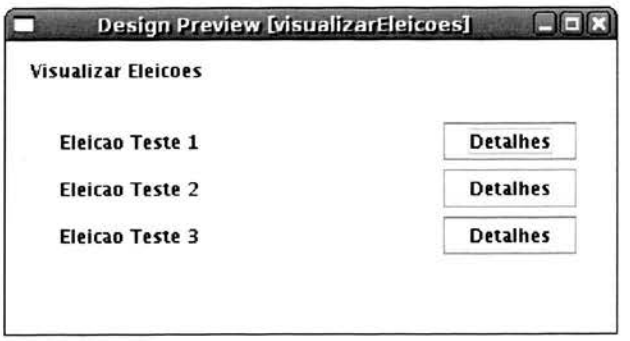


Figure 5.11: Painel gráfico da Electoral Commission Application: Visualizar Eleições

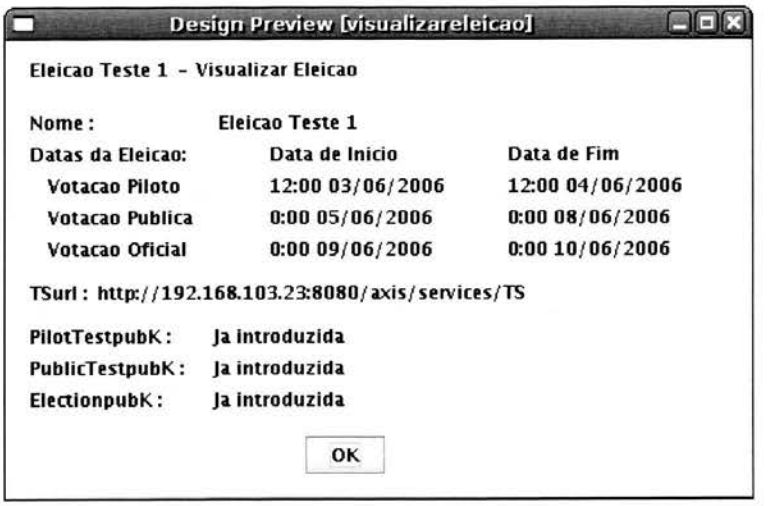


Figure 5.12: Painel gráfico da Electoral Commission Application: Visualizar Eleição

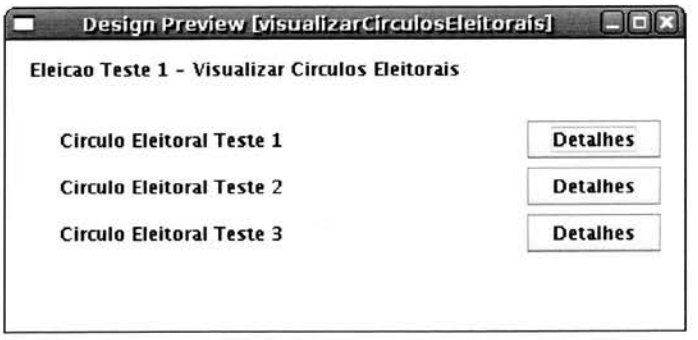


Figure 5.13: Painel gráfico da Electoral Commission Application : Visualizar Circulos Eleitorais

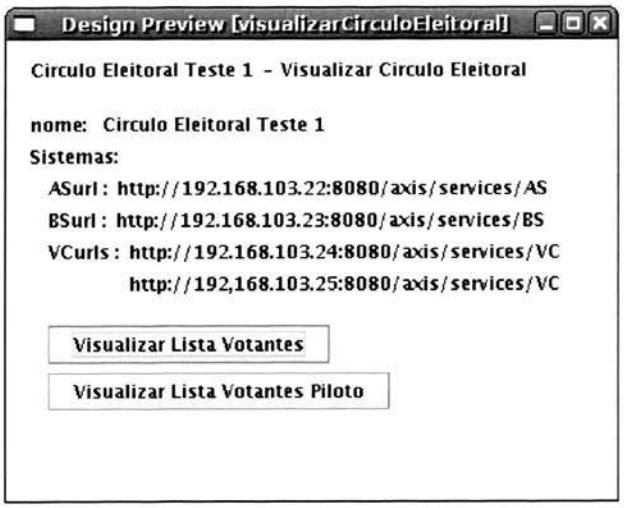


Figure 5.14: Painel gráfico da Electoral Commission Application: Visualizar Circulo Eleitoral

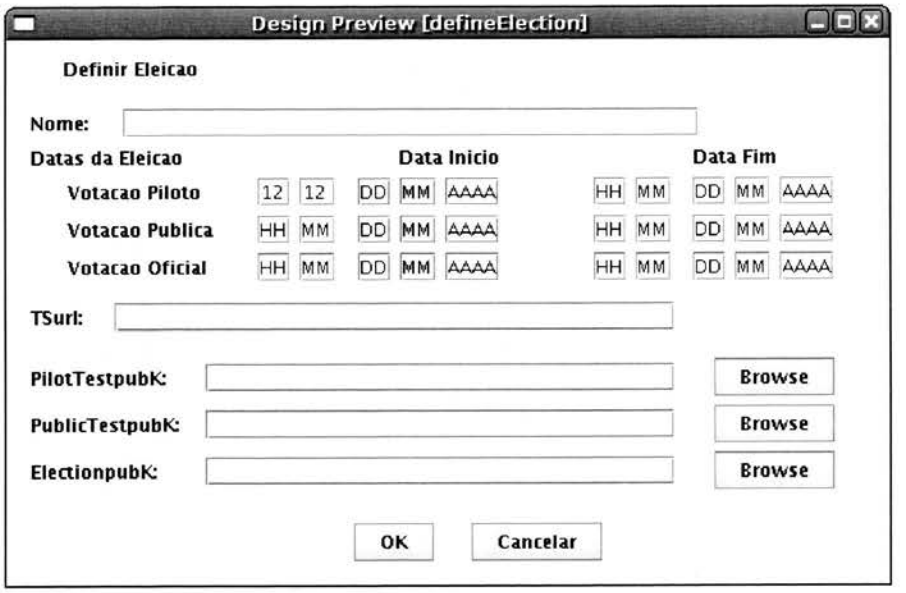


Figure 5.15: Painel gráfico da Electoral Commission Application: Definir Eleição

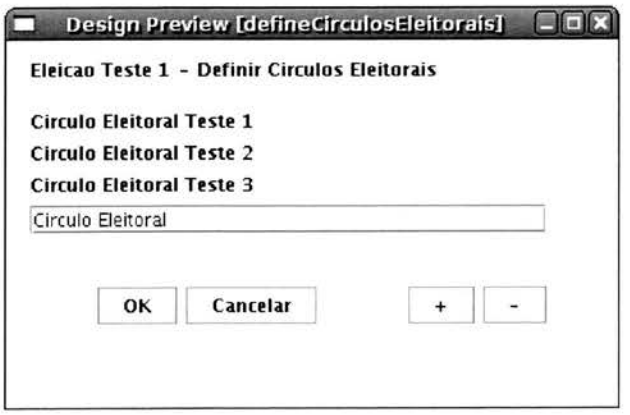


Figure 5.16: Painel gráfico da Electoral Commission Application: Definir Circulos Eleitorais

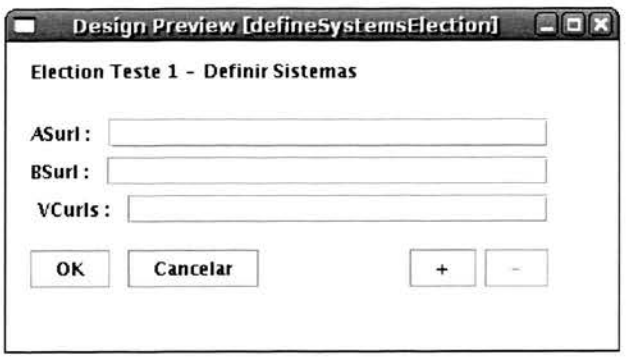


Figure 5.17: Painel gráfico da Electoral Commission Application: Definir Sistemas por eleição

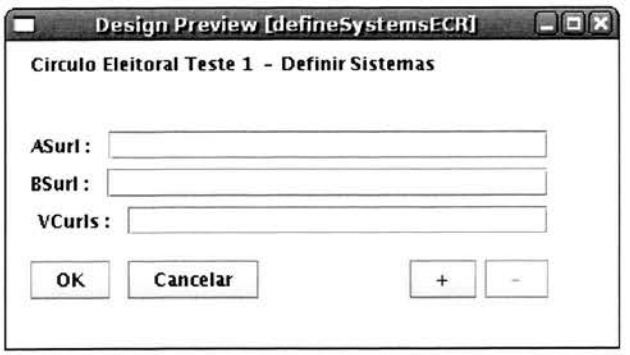


Figure 5.18: Painel gráfico da Electoral Commission Application: Definir Sistemas por circulo eleitoral

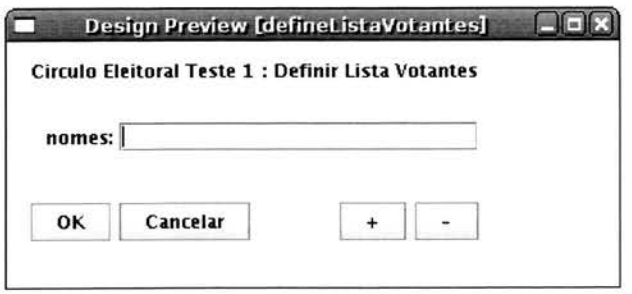


Figure 5.19: Painel gráfico da Electoral Commission Application: Definir lista de votantes oficial

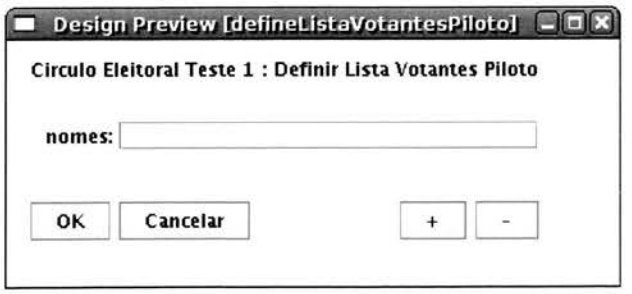


Figure 5.20: Painel gráfico da Electoral Commission Application: Definir lista de votantes piloto

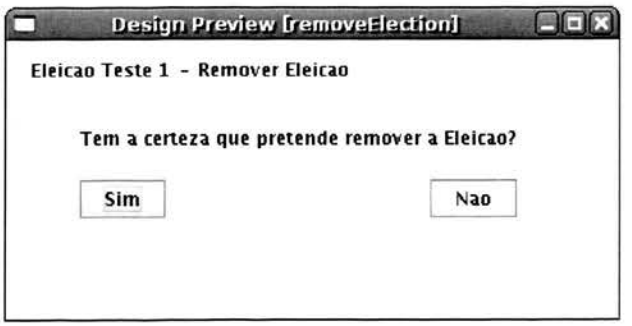


Figure 5.21: Painel gráfico da Electoral Commission Application: Remover Eleição

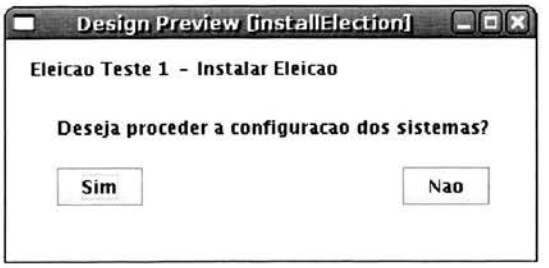


Figure 5.22: Painel gráfico da Electoral Commission Application: Instalar Eleição

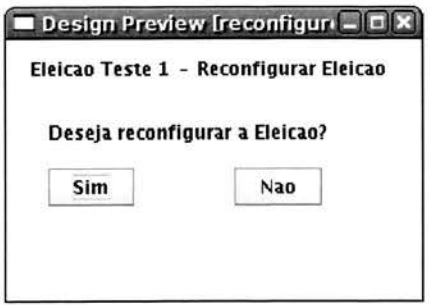


Figure 5.23: Painel gráfico da Electoral Commission Application: Reconfigurar Eleição

Chapter 6

Conclusões

Ao concluir o projecto consideramos que os objectivos inicialmente propostos foram atingidos.

A nossa proposta inicial de trabalho foi partir de uma aplicação de votação electrónica (VE) desenvolvida no âmbito de uma Tese de Mestrado,(a qual foi alvo de uma publicação no EGOV 2005 em Copenhaga, editada pela Springer na série Lecture Notes in Computer Science, Volume 3591/2005) e implementar-la em *web-services*.

Esta aplicação levantou algumas questões de operacionalização ao longo da familiarização com os requisitos que o sistema de votação electrónica deveria possuir. A questão de operacionalização inicial e a qual deu origem a esta proposta de projecto foi que para cada eleição seria necessário proceder à instalação, configuração e criação dos boletins de voto.

Após várias reuniões sobre como deveria ser a arquitectura do nosso sistema para satisfazer os objectivos propostos chegamos a conclusão que deveriamos disponiblizar os sistemas por *Web-Services* onde a instalação (*Hardware* e *Software*) e configuração da eleição só seria efectuada uma única vez.

Para além do principal objectivo proposto que era desenvolver uma aplicação en web-services da PGVE e depois das várias discussões efectuadas tivemos a necesidade de criar um sistema para criar e configurar a eleição. Este sistema foi descrito ao longo do relatório (ver secção 3.8). Para além deste sistema criamos o *Trust System* para tratar da autenticação do eleitor, sendo este sistema externo e disponiblizado pela entidade reguladora da eleição ou comissão eleitoral.

Ao longo da implementação deste trabalho tivemos que fazer vários estudos devido a grande complexidade e ao grande número de tecnologias (ver capítulo 3), usadas para conseguir desenvolver um sistema de votação electrónica capaz de satisfazer a proposta inicial bem como os *upgrades* que efectuamos.

Um requisito que não era um objectivo do projecto proposto mas que tinha bastante importância foi a utilização do XMLsignature, pois esta recomendação do W3C deu-nos garantias sobre a autenticidade dos documentos XML assinados.

Uma ferramenta que foi de grande ajuda foi a utilização da aplicação web Apache Axis a qual fornecia uma API para desenvolvimento de serviços web baseados em troca de mensagens SOAP (para mais informação ver secção 3.5). Na altura de optarmos pelo Axis1 ou Axis2, decidimos pelo Axis1, pois no momento que iniciamos o projecto o Axis 2 ainda estava numa versão testing, por isso decidimos utilizar o Axis1.

6.1 Propostas Futuras

Ao longo do desenvolvimento do trabalho fomos optando por diversas tecnologias por existir mais documentação e por ser mais familiar trabalhar com certas tecnologias em vez de outras.

Um trabalho futuro seria desenvolver este projecto utilizando a API AXIS 2 visto que neste momento já se encontra estável e no futuro esta versão terá certamente mais actualizações comparativamente a versão utilizada neste projecto (ver secção 3). Apache Axis2 suporta não somente o SOAP 1.1 e o SOAP 1.2, mas também integrou a sustentação para o popular REPLY style de serviços. O Apache Axis2 é mais eficiente, mais modular e mais orientado ao XML do que a versão mais velha. É projectado com cuidado para suportar a fácil adição dos "módulos" que estendem sua funcionalidade para características tais como a segurança e a confiabilidade. Para mais informações consultar o site sobre a API Axis2 [19].

Outra implementação que podia ser feito mas por falta de tempo não foi possível implementar-lo, seria utilizar a API WSS4J [20] ou Web Services Security for Java pois com o uso desta tecnologia podíamos garantir trocas de mensagens entre o sistemas e o utilizador de uma forma segura.

Para a ligação com as bases de dados poderíamos utilizar o hibernate 3 [21]. Esta ferramenta de mapeamento objecto/relacional para java transforma tabelas de base de dados em um grafo de objectos definidos pelo programador. Com o uso do hibernate seria desenvolvido mais rapidamente o trabalho, já que o programador não teria que escrever muito do código de acesso a base de dados e de SQL[22].

Outras funcionalidades que poderiam ser implementadas podiam ser a criação de uma aplicação para fazer a auditabilidade do sistema bem como boletins de voto dinâmicos por forma a estes serem utilizados em inquéritos e criar grupos indefinidos em número de forma a agrupar os círculos eleitorais.

Também um requisito que a eleição deveria possuir seria, a independência dos fusos horários visto que em muitos casos a mesma eleição ocorre em fusos de horários diferentes (ex: Eleições autárquicas em Portugal Continental e nos Açores).

Umas das ideias apresentadas na workshop sobre voto electrónico [2] e que achamos interessante para uma futura implementação foi a replicação do *Ballot System* de forma a que um voto só será contado se passar por n/2+1 BS (onde o valor de n é o numero de *Ballot Systems*).

Para eliminar as existências de vírus poderiamos criar um CD Boot onde as fragilidades do sistema operativo desapareceriam.

Bibliografia

[1] CNPD quer garantias de privacidade no voto electrónico http://tek.sapo.pt/4M0/627338.html

[2] Workshop sobre Voto pela Internet http://www.ieeta.pt/~avz/WorkshopEVoto06/

[3] iBallot http://www.iballot.com

[4] SafeVote http://www.safevote.com/

[5] SCYTL - Secure Electronic Voting http://www.scytl.com/

[6] VoteHere http://www.votehere.net/

[7] EveryoneCounts http://www.everyonecounts.com/

[8] Costa, Ricardo André Fernandes. Plataforma Genérica de Votação Electrónica. Porto: FEUP, 2006.

[9] Gentoo Linux http://www.gentoo.org

[10] XML http://en.wikipedia.org/wiki/Xml

[11] XML W3C Recomendation http://www.w3.org/TR/2004/REC-xml-20040204/

[12] SOAP W3C Tecnical Reports http://www.w3.org/TR/soap/

[13] SOAP wikipedia http://en.wikipedia.org/wiki/SOAP

[14] WSDL 1.1 W3C Note http://www.w3.org/TR/wsdl

[15] Apache Axis http://ws.apache.org/axis

[16] JavaBeans Activation Framework http://java.sun.com/products/javabeans/jaf/index.jsp

[17] Apache XML Security http://xml.apache.org/security/

[18] Base de Dados Postgresql http://www.postgresql.org

[19] AXIS2 http://ws.apache.org/axis2/

[20] WSS4J http://ws.apache.org/wss4j/

[21] Hibernate http://www.hibernate.org/

[22] SQL http://www.sql.org/

[23] Pagina do Projecto http://wiki:8668/space/start