



**UNIVERSIDADE TÉCNICA DE ANGOLA  
FACULDADE DE ENGENHARIAS  
DEPARTAMENTO DE ENSINO E INVESTIGAÇÃO DAS TECNOLOGIAS DE  
INFORMAÇÃO E COMUNICAÇÃO - DEITIC**

**CURSO DE ENGENHARIA INFORMÁTICA**

**(DESENVOLVIMENTO DE UM SISTEMA DE VOTOS ELECTRÓNICOS EM  
ANGOLA)**

**ROMÁRIO MEIRELES SEBASTIÃO ÂNGELO**

**LUANDA-ANGOLA**

**MARÇO/2017**

**UNIVERSIDADE TÉCNICA DE ANGOLA  
FACULDADE DE ENGENHARIAS  
DEPARTAMENTO DE ENSINO E INVESTIGAÇÃO DAS TECNOLOGIAS DE  
INFORMAÇÃO E COMUNICAÇÃO - DEITIC**

**CURSO DE ENGENHARIA INFORMÁTICA**

**(DESENVOLVIMENTO DE UM SISTEMA DE VOTOS ELECTRÓNICOS EM  
ANGOLA)**

**ROMÁRIO MEIRELES SEBASTIÃO ÂNGELO**

*Trabalho de Fim de curso  
apresentado a Universidade Técnica  
de Angola, como requisito para  
obtenção do título de licenciado em  
Engenharia Informática*  
Orientador: Eng. Martinho Kassequele.

**LUANDA-ANGOLA**

**MARÇO/2017**

**ROMÁRIO MEIRELES SEBASTIÃO ÂNGELO**

**DESENVOLVIMENTO DE UM SISTEMA DE VOTOS ELECTRÓNICOS EM  
ANGOLA**

**Aprovado:** \_\_\_\_/\_\_\_\_/\_\_\_\_

**BANCA EXAMINADORA**

Presidente do Júri.....	Assinatura.....
Primeiro Vogal.....	Assinatura.....
Segundo Vogal.....	Assinatura.....

## DEDICATÓRIA

Dedico este trabalho a toda minha família,  
pelo apoio que me foi dado, durante este  
percurso, especialmente os meus pais,  
**Sebastião Vanda Ângelo Madalena**  
**e Amélia António Sebastião,**  
que foram os pilares fundamentais para  
essa concretização.

## AGRADECIMENTOS

Primeiramente agradeço a Deus,  
por me ter proporcionado saúde, força de vontade,  
e como também por iluminar o meu caminho,  
de modo a explorar o poder da mente.  
Agradeço a Universidade Técnica de Angola,  
por tentar fazer sempre juz a frase  
“garantia de qualidade” e por me formar.  
Ao meu tutor **Engº Martinho Kassequele**,  
que foi o meu pai na universidade, que desde sempre  
partilhou conhecimento comigo, apoiou-me, acreditou em mim,  
tenho muito para agradecer a este senhor  
Aos professores que apoiaram intensamente,  
que deram o seu contributo de forma espontânea  
para a minha formação, tal como o **Msc. Amiralde Xavier**,  
sempre o tive como um mestre,  
durante muito tempo que vem a me motivar e confiar em mim,  
é uma honra poder ter este senhor por perto.  
agradeço também ao **Msc. Emanuel Mateus** tenho o como  
impulsionador de carreiras, agradeço muito  
pela sua paciência, partilha de conhecimento,  
experiência, pela sua postura inspiradora.  
Agradeço a CNE pela disponibilidade em prestar informações,  
tanto presencialmente como a distância.  
Aos meus colegas que directa ou indirectamente deram-me  
muita ajuda neste percurso, especialmente o meu caro colega  
**Francisco Cafumana** pela troca de ideias, experiência e  
pela sua humildade também.  
Finalmente, em especial à minha família, graças a eles  
estou a realizar o sonho de ser engenheiro, agradeço muito a eles  
Por serem o meu pilar, para mim eles são a base de tudo.

## EPIGRÁFE

*“a persistência  
é o caminho do êxito.”  
Charles Chaplin.*

## **RESUMO**

O acto de votar é um dos actos mais representativos de uma democracia e constitui um dos principais direitos que esse regime confere aos cidadãos.

Dada a sua relevância, o processo de votação tem sido alvo de muitos estudos e melhorias ao longo do tempo para colmatar lacunas existentes em sistemas de votação, sabemos que a forma tradicional do direito ao voto resume-se em o próprio cidadão se dirigir as assembleias de voto, identificar-se para garantir o secretismo do mesmo.

O objectivo deste trabalho é descrever uma forma alternativa, mais eficaz e cómoda de voto, que permite aumentar a participação dos cidadãos angolanos nos actos eleitorais em Angola, diminuir o abstencionismo. Esta alternativa consiste na criação de um sistema de votos electrónicos baseado em tecnologias web, utilizando princípios que regem a boa prática de desenvolvimento de sistemas, práticas essas como, inversão de controle através da injeção de dependências de modo a evitar que o nosso projecto tenha um forte acoplamento entre os demais componentes, aplicação de webdesign reponsivo com o bootstrap para termos um webiste ajustável a qualquer tela, utilização do ASPNET MVC para melhor separação das responsabilidades na camada de apresentação, a arquitectura do nosso projecto foi modelada com base na abordagem do DDD.

**Palavras Chaves:** votação electrónica, votos electrónicos

## **ABSTRACT**

The act of voting is one of the most representative acts of a democracy and is one of the main rights that this regime gives citizens.

Given its relevance, the voting process has been the subject of many studies and improvements over time in order to fill gaps in voting systems, we know that the traditional form of voting rights boils down to the citizen himself addressing the assemblies to identify themselves to guarantee the secrecy of the same.

The purpose of this paper is to describe an alternative, more effective and convenient way of voting, which allows to increase the participation of the Angolan citizens in the electoral acts in Angola, to reduce the abstention. This alternative consists in the creation of an electronic voting system based on web technologies, using principles that govern the good practice of systems development, practices such as, inversion of control through the injection of dependencies in order to avoid that our project has a strong coupling between the other components, application of webdesign reponsivo with the bootstrap to have a webiste adjustable to any screen, use of ASPNET MVC to better separation of the responsibilities in the presentation layer, the architecture of our project was modeled based on the DDD approach .



## **LISTA DE SIGLAS E ABREVIATURA**

API – Application Programming Interface

ASP – Active Server Page

CASE – Computer-Aided Software Engineering

CNE – Comissão Nacional Eleitoral

DDD – Domain Drive Design

DADI – Definition, Architecture, Design, Implementation

DDL – Data Definition Language

DML – Data Manipulation Language

DQL – Data Query Language

DCL – Data Control Language

DTL – Data Transaction Language

FDD – Feature Driven Development

GUI – Graphical User Interface

HTTP – HyperText Transfer Protocol

ISO – International Organization for Standardization

IDEA – International Institute for Democracy and Electoral Assistance

IIS – Internet Information Service

LDAP – Lightweight Directory Access Protocol

LDF – Log Data File

LINQ – Language Integrated Query

MDF – Master Data File

MVC – Model View Controller

MVW – Model View Whatever

MAXDOP – Max Degree Of Parallelism

MDD - Model Drive Design

NoSql – Not only Sql

RAID – Redundant Array of Inexpensive Drives

RF – Requisitos Funcionais

RNF – Requisitos não funcionais

REST-Representational State Transfer  
SDLC – Software Development Life Cycle.  
SI – Sistema de Informação  
SQL – Structured Query Language  
SVU – Sistema de Votação em Urna  
SVE – Sistema de Votos Electrónicos  
SVEP – Sistema de Votação Electrónica Presencial  
SGBD – Sistema de Gestão de Base de dados  
SPA – Single Page Application  
SOAP-Simple Object Access Protocol  
TDD – Test Driven Development  
UML – Unified Modelling Language  
URL – Uniform Resource Locator  
VU – Votação em Urna  
VER – Votação Electrónica Remota  
VE – Votação Electrónica  
WWW – World Wide Web  
XP – Extreme Programming  
EAI-Enterprise Application Integration

## LISTA DE FIGURAS

<b>Figura 1:</b> Linguagens do SQL.....	30
<b>Figura 2:</b> Arquitectura em camadas, utilizada para separar o domínio do resto da aplicação..	32
<b>Figura 3:</b> Diagrama de caso de uso .....	49
<b>Figura 4:</b> Diagrama de entidade-relacionamento.....	53
<b>Figura 5:</b> Diagrama lógico.....	54
<b>Figura 6:</b> Diagrama de classe.....	55
<b>Figura 7:</b> Diagrama de sequência incluir eleitor ao caderno eleitoral .....	56
<b>Figura 8:</b> Diagrama de sequência cadastrar ano eleitoral.....	56
<b>Figura 9:</b> Diagrama de sequência votar .....	57
<b>Figura 10:</b> Diagrama de actividade cadastrar ano eleitoral .....	57
<b>Figura 11:</b> Diagrama de actividade incluir eleitor ao caderno eleitoral .....	58
<b>Figura 12:</b> Diagrama de actividade votar.....	58
<b>Figura 13:</b> Diagrama de instalação .....	59
<b>Figura 14:</b> Exemplo de validação de dados no sistema.....	66
<b>Figura 15:</b> Padrão Template Method .....	69
<b>Figura 16:</b> Interoperabilidade do sistema.....	70
<b>Figura 17:</b> Tela do administrador .....	74
<b>Figura 19:</b> Tela de cadastro de eleitor .....	75
<b>Figura 20:</b> Tela de inclusão do eleitor ao caderno eleitoral.....	75
<b>Figura 21:</b> Tela de pesquisar eleitor no caderno eleitoral .....	76
<b>Figura 22:</b> Tela de visualização de boletim de voto .....	76
<b>Figura 23:</b> Tela de confirmação do voto.....	77

## LISTA DE TABELAS

<b>Tabela 1:</b> Requisitos funcionais .....	45
<b>Tabela 2:</b> Requisitos não funcionais .....	46
<b>Tabela 3:</b> Regras de negócio .....	48
<b>Tabela 4:</b> Casos de uso .....	50
<b>Tabela 5:</b> Detalhamento do caso de uso cadastrar ano eleitoral.....	51
<b>Tabela 6:</b> Detalhamento do caso de uso incluir eleitor ao caderno eleitoral.....	52
<b>Tabela 7:</b> Detalhamento do caso de uso votar .....	52
<b>Tabela 8:</b> AnoEleitoral .....	81
<b>Tabela 9:</b> Eleitor.....	82
<b>Tabela 10:</b> BI .....	82
<b>Tabela 11:</b> Caderno Eleitoral .....	83
<b>Tabela 12:</b> Estado Eleitor .....	83
<b>Tabela 13:</b> Mandato .....	84
<b>Tabela 14:</b> Comissao.....	84
<b>Tabela 15:</b> Tipo Comissao .....	85
<b>Tabela 16:</b> Província .....	85
<b>Tabela 17:</b> Município .....	86
<b>Tabela 18:</b> Bairro .....	86
<b>Tabela 19:</b> Rua.....	87
<b>Tabela 20:</b> Grupo .....	87
<b>Tabela 21:</b> Boletim.....	88
<b>Tabela 22:</b> Partido .....	88
<b>Tabela 23:</b> Voto .....	89
<b>Tabela 24:</b> CarregamentoSms .....	89
<b>Tabela 25:</b> EnvioSms .....	90
<b>Tabela 26:</b> CategoriaUsuario .....	90
<b>Tabela 27:</b> AspNetUsers.....	91
<b>Tabela 28:</b> Recenseador.....	92
<b>Tabela 29:</b> Orçamento .....	93

## ÍNDICE

<b>DEDICATÓRIA</b> .....	i
<b>AGRADECIMENTOS</b> .....	ii
<b>EPIGRÁFE</b> .....	iii
<b>RESUMO</b> .....	iv
<b>ABSTRACT</b> .....	v
<b>LISTA DE SIGLAS E ABREVIATURA</b> .....	vi
<b>LISTA DE FIGURAS</b> .....	viii
<b>LISTA DE TABELAS</b> .....	ix
<b>1. INTRODUÇÃO</b> .....	1
<b>1.1 DEFINIÇÃO DO PROBLEMA</b> .....	2
<b>1.2 JUSTIFICATIVA</b> .....	2
<b>1.3 OBJECTIVOS</b> .....	3
<b>1.3.1 OBJECTIVO GERAL</b> .....	3
<b>1.3.2 OBJECTIVOS ESPECÍFICOS</b> .....	3
<b>1.4 DELIMITAÇÃO</b> .....	3
<b>1.5 ESTRUTURA DO TRABALHO</b> .....	3
<b>2. CAPÍTULO 1-FUNDAMENTAÇÃO TEÓRICA</b> .....	5
<b>2.1 Contexto e motivação</b> .....	5
<b>2.2 Potencialidades e problemas</b> .....	5
<b>2.3 Processo eleitoral tradicional</b> .....	6
<b>2.3.1 Recenseamento</b> .....	6
<b>2.3.2 Validação da identificação do eleitor</b> .....	7
<b>2.3.3 Recolha dos votos</b> .....	7
<b>2.3.4 Contagens dos votos</b> .....	7
<b>2.4 Votação em urnas</b> .....	7
<b>2.4.1 Processo de votação em urnas</b> .....	8
<b>2.4.2 Requisitos gerais da votação em urnas</b> .....	9
<b>2.4.3 Vantagens e limitações</b> .....	10
<b>2.5 Votação electrónica</b> .....	12
<b>2.5.1 Votação electrónica presencial</b> .....	13
<b>2.5.2 Votação electrónica remota</b> .....	14
<b>2.5.3 Requisitos dos sistemas de votação electrónica</b> .....	15

2.5.3.1 Acessibilidade e Conveniência .....	16
2.5.3.2 Anonimato.....	17
2.5.3.3 Audibilidade e certificabilidade .....	18
2.5.3.4 Autenticidade .....	19
2.5.3.4 Detetabilidade e recuperabilidade .....	20
2.5.3.5 Disponibilidade .....	20
2.5.3.6 Integridade do voto .....	21
2.5.3.7 Mobilidade .....	21
2.5.3.8 Não-coercibilidade .....	22
2.5.3.9 Singularidade .....	23
2.5.3.10 Transparência .....	23
2.5.3.11 Verificabilidade .....	24
2.5.4 Confiança.....	24
2.5.4.1 Confiança nos sistemas de votação electrónica .....	25
2.5.5 Desvantagens do sistema de votação electrónica .....	26
2.5.5.1 Falhas de sistemas de votação electrónica.....	26
2.6 Sistemas de informação.....	27
2.7. Internet.....	27
2.8 Site.....	28
2.9 Base de dados .....	28
2.9.1 Sistemas de gestão de base de dados .....	29
2.9.2 SQL.....	30
2.10. Plataformas de desenvolvimento web.....	30
2.10.1 ASP.NET MVC .....	31
2.11 Domain Drive Design .....	31
2.11.1 Linguagem Ubíqua.....	31
2.11.2 Camada de anti-corrupção .....	34
2.12 Integração de tecnologias .....	34
2.12.1 ASP.NET WEB API.....	35
2.12.2 Google Maps API.....	36
2.13 Segurança.....	36
2.14 Metodologias de desenvolvimento de software.....	37
2.14.1 Visão geral da XP.....	38
2.14.1.1 Equipe coesa.....	38
2.14.1.2 Histórias de usuário .....	39

2.14.1.5 Desenvolvimento guiado por testes (TDD- Test-Driven Development).....	39
2.14.1.6 Refacturação.....	39
2.14.2 Ciclo de vida do desenvolvimento de sistema.....	40
2.15 UML.....	41
2.15.1 Ferramentas CASE.....	42
3. CAPITULO 2 - METODOLOGIA.....	42
3.1 Tipo de pesquisa.....	42
3.2 Campo de estudo.....	43
3.3 Estudo de viabilidade.....	43
3.4 Levantamento de requisitos.....	43
3.4.1 Análise de requisitos.....	44
3.4.1.1 Requisitos funcionais.....	44
3.4.1.2 Requisitos não funcionais.....	45
3.4.1.3 Regras de negócio.....	46
3.5 Análise do sistema.....	48
3.5.1 Diagramas do sistema.....	48
3.5.2 Diagrama de caso de uso.....	48
3.5.2.1 Descrição dos principais casos de uso.....	50
3.5.3 Diagrama de entidade-relacionamento.....	52
3.5.4 Diagrama lógico.....	54
3.5.5 Diagrama de classe.....	55
3.5.6 Diagrama de sequência.....	55
3.5.8 Diagrama de instalação.....	59
3.6 Ferramentas de desenvolvimento.....	60
3.7 Arquitectura lógica do sistema.....	60
3.7.1 Camada de apresentação.....	61
3.7.2 Camada de serviços.....	61
3.7.3 Camada de aplicação.....	61
3.7.4 Camada de domínio.....	61
3.7.5 Camada de infra-estrutura.....	62
3.7.6.1 Camada de testes.....	62
3.8 Desenvolvimento de base de dados.....	62
3.8.1 Criação da base de dados.....	62
3.9 Tecnologias envolvidas no projecto.....	63
3.9.1 Base de dados.....	63

3.9.1.1	Procedimentos armazenados.....	63
3.9.1.2	Sequências.....	63
3.9.1.3	Column Store Indexes.....	63
3.9.1.4	Paralelismo.....	63
3.9.1.5	CTE.....	64
3.9.1.6	Audit.....	64
3.9.1.7	Always encrypted.....	64
3.9.2	Programação.....	65
3.9.2.1	ASP.NET 4.5.2.....	65
3.9.2.2	Bootstrap.....	65
3.9.2.3	Validação.....	65
3.9.2.4	jQuery.....	66
3.9.2.5	AngularJS.....	66
3.9.2.6	Inversão de controle.....	67
3.9.2.6.1	Injeção de dependência.....	67
3.9.2.6.2	Ninject.....	67
3.9.2.7	Padrão repositório.....	67
3.9.2.8	Entity Framework 6.0.....	67
3.9.2.9	Scaffolding.....	68
3.9.2.10	Nuget.....	68
3.9.2.10	ASP.NET Identity.....	68
3.9.2.11	LINQ.....	68
3.9.2.12	ADO.NET.....	69
3.9.2.13	AutoMapper.....	69
3.9.2.14	Padrão de projecto Template Method.....	69
3.9.2.15	Json.....	70
3.9.2.16	ASPNET WEB API.....	70
3.9.3	Implementação de segurança.....	71
3.9.4	Fase de testes.....	71
3.9.4.1	Teste de unidade.....	72
3.9.4.2	Teste de integração.....	72
3.9.4.3	Teste de sistema.....	72
3.9.4.4	Teste de aceitação.....	73
4	RESULTADOS.....	74
4.1	Descrição do projecto.....	74



<b>4.2</b>	<b>Interface do sistema.....</b>	<b>74</b>
4.2.1	Tela principal do administrador.....	74
4.2.2	Tela de cadastro de eleitor.....	75
4.2.3	Tela de inclusão do eleitor ao caderno eleitoral.....	75
4.2.4	Tela de pesquisar eleitor no caderno eleitoral.....	76
4.2.6	Tela de visualização de boletim de voto.....	76
4.2.7	Tela de confirmação de voto.....	77
<b>4.3</b>	<b>Análise dos resultados .....</b>	<b>77</b>
<b>CONCLUSÃO.....</b>		<b>78</b>
<b>RECOMENDAÇÕES.....</b>		<b>79</b>
<b>REFERÊNCIAS BIBLIOGRAFICAS.....</b>		<b>80</b>
<b>APÊNDICE I – DICIONÁRIO DE DADOS.....</b>		<b>81</b>
<b>APÊNDICE II – ORÇAMENTO .....</b>		<b>93</b>
<b>APÊNDICE III – ROTEIRO DA ENTREVISTA.....</b>		<b>94</b>
<b>APÊNDICE IV – EXEMPLO DA ENTREVISTA.....</b>		<b>95</b>
<b>ANEXOS .....</b>		<b>97</b>

# 1. INTRODUÇÃO

O acto eleitoral é uma peça chave na democracia e representa um dos seus principais e mais complexos direitos. Este acto está fortemente associado à escolha de representantes para diversos cargos de diferentes áreas, como o presidente do condomínio ou delegado de turma, mas o que o torna bastante complexo e de elevada importância são as suas aplicações a votações/eleições nacionais, tipicamente para eleger órgãos governamentais, e todas as características de segurança e secretismo associadas.

A forma mais tradicional de exercício do direito ao sufrágio consiste no acto do próprio cidadão dirigir-se às assembleias de voto, onde para já inclui um certo custo, identificar-se e proceder à votação através de boletins em papel não identificados para garantir o secretismo do mesmo, esses votos depois são contados manualmente e correm o risco de ser mal contados, actualmente em Angola é feito desta forma.

E como tanto quanto o mundo tem se tornado cada vez mais digitalizado, hoje em dia em função do avanço tecnológico tem se desenvolvido várias soluções para termos uma sociedade melhor, desenvolvendo várias técnicas mais sofisticadas, essas técnicas são passadas de geração para geração, assim como o homem cria o martelo e o martelo recria o homem.

O sistema de votos electrónicos a ser apresentado é apenas mais um recurso (técnica) que nos permite escolher através de votação de forma digital o órgão a governar o país, em alguns países já implementaram o mesmo, embora que ainda existam algumas desconfianças, mas traz muitas melhorias, assim como aposta na comodidade e conforto dos cidadãos, redução de custos de processos eleitorais, permitindo também, de acordo com experiência internacional capitalizar a interação com os cidadãos.

## **1.1 DEFINIÇÃO DO PROBLEMA**

Actualmente em Angola encontra-se em uso o sistema de votação em urnas, que apresenta certas lacunas, tal como, a integridade e a verificabilidade(auditabilidade) do voto, que visa garantir que os votos não sejam alterados, nem eliminados durante e depois das eleições, e que seja possível provar que os votos foram devidamente contados. Existem intervenientes e até equipas de auditores que podem pôr em causa o cumprimento destes requisitos através da alteração dos votos em fases mais restritas das eleições, onde apenas estes estão presentes.

No sistema de votação em urnas, os votos são contados manualmente, desta forma proporcionam maior probabilidade de erro devido à intervenção humana e tornam o processo muito mais demorado, requerem transporte seguro dos votos.

No sistema de votação em urna é necessário a distribuição dos boletins de votos nos mais variados locais de votação. Há estudos que indicam que as opções colocadas nas posições finais do boletim têm menor probabilidade de serem escolhidas.

Tendo em conta os problemas apresentados eis que se levanta a seguinte questão de pesquisa para este trabalho:

Como solucionar por meio tecnológico os problemas encontrados no sistema de votação em urnas?

## **1.2 JUSTIFICATIVA**

Partindo do pressuposto que um projecto final do curso visa solucionar sempre problemas de forma a inovar, torna-se ainda mais interessante quando as soluções apresentadas são de impacto social, então por essa razão decidimos desenvolver um sistema de votos electrónicos afim de suprir algumas debilidades que os sistemas de votação em urnas apresentam conforme citados anteriormente, de modo a encontrar uma forma alternativa segura, mais eficaz e cómoda de voto, que permite aumentar a participação nos actos eleitorais, diminuir o abstencionismo, dada a dispersão geográfica dos cidadãos com incidência aos que estão na diáspora que em eleições gerais, se tem vindo a acentuar. Num ponto de vista social essa é a motivação que tive para desenvolver esse tema.

Num ponto de vista técnico também surgiram algumas razões, ou seja, motivos para desenvolver este tema, sendo um informático, esse tema proporciona alguns desafios, levando em conta os atributos necessários para concebe-lo, tal como a segurança, privacidade,

autenticidade, e outros mais, com estes pormenores o tema suscitou em nós um grande interesse e de certa forma passou a ser mais atractivo pelos recursos que poderão ser implementados, a fim de poder estender o meu conhecimento para poder desenvolver o tema.

## **1.3 OBJECTIVOS**

### **1.3.1 OBJECTIVO GERAL**

O objectivo deste trabalho consiste em criar um sistema de votos electrónicos, para solucionar os problemas encontrados no sistema de votação em urnas.

### **1.3.2 OBJECTIVOS ESPECÍFICOS**

- Levantamento e análise de requisitos;
- Desenvolver uma base de dados para persistir os dados de acordo as necessidades;
- Implementar módulo para o recenseamento eleitoral, votação, contabilização e comunicação;
- Fazer integração entre o sistema de votos electrónicos com o sistema da CNE através de uma web api;
- Enquadrar uma funcionalidade de geolocalização com o google maps, a fim de facilitar a localização do local de voto para quem não puder votar remotamente;
- Testar e mostrar as funcionalidades do sistema.

## **1.4 DELIMITAÇÃO**

Dada as circunstâncias actuais do país, a implementação deste trabalho delimita-se inicialmente em Luanda para uma fase experimental, levando em conta também os pontos a serem analisados e a tamanha complexidade dos mesmos para implementar o sistema no país como um todo.

## **1.5 ESTRUTURA DO TRABALHO**

Para melhor enquadramento e situação no relatório, este relatório foi estruturado por capítulos, conforme é apresentado abaixo:

**Capítulo 1: Fundamentação teórica** - Este é o capítulo reservado a abordagem mais teórica sobre o projecto desenvolvido, apresentando algumas tecnologias envolvidas no projecto, arquitecturas e metodologias usadas para a construção de softwares.

**Capítulo 2 – Metodologia** - Este apresenta as técnicas, os procedimentos e os métodos de pesquisa usados para a elaboração deste projecto, apresentamos os diagramas UML em função da análise que foi feita, fizemos a descrição das tecnologias envolvidas e a sua razão, e mais outros factores importantes.

**Capítulo 3 – Resultados** - Neste capítulo apresentamos o nosso projecto de forma funcional, a interface juntamente com alguma explicação do funcionamento do mesmo.

## **2. CAPITULO 1-FUNDAMENTAÇÃO TEÓRICA**

### **2.1 Contexto e motivação**

Os sistemas de votação representam o meio utilizado pelas pessoas possuidoras do direito de votar (eleitores) para escolherem livremente entre diferentes opções. Ao longo do tempo este tem sido o principal método utilizado para a escolha dos representantes governamentais, baseados normalmente na regra da maioria, apesar de existir uma grande variedade de sistemas de apuramento dos resultados. A generalidade do sistema de votação actual parte de alguns pressupostos bases como:

- Necessidade do recenseamento antes da votação;
- Identificação do eleitor perante a entidade reguladora da eleição;
- Escolha da opção de voto por parte do eleitor;
- Contagem dos votos;
- Divulgação dos resultados (podendo ou não haver recontagem dos votos).

A motivação maior desse trabalho é de descrever e encontrar uma forma alternativa segura, mais eficaz e cómodo de voto, que permite aumentar a participação nos actos eleitorais angolanos, diminuir o abstencionismo, dada a dispersão geográfica das nossas e da diáspora que, em eleições gerais, se têm vindo a acentuar.

A construção de uma sociedade baseada na informação e no conhecimento, assim como a oposta na boa governação e dada a grande aceitação das novas tecnologias de informação e comunicação, as mesmas podem ser vistas como facilitadores de práticas democráticas, nomeadamente no suporte ao exercício do voto.

### **2.2 Potencialidades e problemas**

São inúmeras as potencialidades que os sistemas de votação electrónica encerram sobre si. Após um investimento inicial, tais sistemas permitem uma enorme poupança de dinheiro, quer por parte da entidade organizadora da votação, quer por parte dos partidos políticos.

Para além dessa poupança, pode-se poupar-se tempo por parte da entidade organizadora, pois, uma eleição suportada electronicamente pode ser organizada num menor intervalo de tempo do que uma eleição convencional.

Por parte dos votantes, a poupança de tempo reflecte-se no facto de não ser necessária a sua presença numa determinada assembleia de voto, uma vez que essa presença é, por vezes, sujeita a intermináveis filas de espera.

Contudo, existe ainda, uma falta de maturidade no desenho e desenvolvimento de sistemas de suporte à votação electrónica. Esta falta de maturidade prende-se com a complexidade inerente a um sistema deste tipo, complexidade essa que é reflectida por vários requisitos que são necessários assegurar, tais como auditabilidade, certificabilidade, confiabilidade, detectabilidade, integridade do sistema, disponibilidade, verificabilidade, privacidade, entre outros.

Outro dos problemas que se afigura é de natureza cultural. Numa eleição convencional, os votantes são confrontados, numa assembleia de voto, com documentos que os identificam (Bilhete de Identidade, normalmente), boletins de voto de papel, urnas de voto, representantes da entidade organizadora da eleição e de partidos políticos. No entanto, numa votação electrónica, os votantes deparam-se ainda com uma entidade física, um computador ou um outro qualquer dispositivo electrónico, com um inexpressivo software para muitos deles.

Por outro lado, existe o problema da conveniência social. Infelizmente, muitas são as pessoas que não podem aceder às últimas tecnologias, apesar de, tal como os outros cidadãos, terem o direito a exercer o seu voto.

## **2.3 Processo eleitoral tradicional**

Um dos objectivos de qualquer SVE é proporcionar um aumento das oportunidades de voto, ou seja, criar maior número de locais onde se torna possível exercer o direito de voto e não obrigatoriedade de o eleitor se apresentar nos locais de recenseamento.

Seguidamente, descrevo brevemente o processo do voto tradicional.

### **2.3.1 Recenseamento**

Como sabemos, o recenseamento consiste do registo, no caderno eleitoral, da pessoa que tenha as condições exigidas por lei para o processo de eleição em questão. Cada cidadão deve receber um cartão de eleitor com um número único, no respectivo caderno, que serve de identificação, na altura da eleição.

### **2.3.2 Validação da identificação do eleitor**

A validação da identificação do eleitor é feita na altura em que o eleitor se dirige à mesa de voto e requisita o seu boletim de voto. O presidente da mesa pronuncia o nome em voz alta e número do eleitor e os restantes elementos da mesa registam, na sua lista de eleitores, o votante. Esta fase acontece no decorrer da eleição, ao contrário da fase anterior.

O mesmo registo serve para evitar o voto múltiplo, isto é, serve para garantir a unicidade do voto (princípio democrático de uma pessoa votar uma única vez).

### **2.3.3 Recolha dos votos**

Após o levantamento do boletim de voto, o eleitor dirige-se à cabine de voto onde regista no boletim a sua votação para depois dobrar o boletim e dirige-se, novamente à mesa de voto onde introduz o seu boletim na urna. Termina nesta fase a participação do eleitor. Os elementos da mesa acompanham esta fase.

### **2.3.4 Contagens dos votos**

Esta tarefa fica a cargo dos elementos da mesa. Após o encerramento da votação, as urnas são abertas e os votos são validados, classificados e contados, sendo depois publicado o resultado da contagem.

## **2.4 Votação em urnas**

A VU é o sistema de votação mais utilizado e aceite em todo o mundo. De forma sucinta, é uma votação pessoal e presencial (salvo exceções referidas na legislação) na qual o eleitor faz uma marca na sua opção de voto no boletim de papel e coloca-o numa urna, sendo os votos presentes na urna contados manualmente no final do tempo alocado para a votação. No entanto o processo de votação em urna engloba um conjunto de fases e requisitos a ter em conta antes, durante e depois da votação e cuja análise contribui para uma melhor compreensão da sua complexidade.

O SVU veio, aquando do seu aparecimento no século XIX (Gaspar 2007), colmatar várias lacunas existentes nos sistemas até então, em que não se utilizava recenseamento nem o voto era secreto pois era comunicado a um juiz, demonstrando graves problemas de privacidade e segurança. A introdução do voto em papel e outras melhorias ao longo do tempo levaram ao



que é actualmente a VU, com cada vez mais regras e requisitos que têm que ser rigorosamente cumpridos, tornando-a no sistema mais utilizado e aceite em todo o mundo. Alguns autores definem ainda a divisão da VU em duas variantes: um sistema de votação localizada onde os votos são contados localmente sendo os resultados comunicados a uma entidade superior e sistemas onde a contagem dos votos é feita numa central de contagem sendo os votos das várias mesas de voto enviados para essa entidade (Monteiro 2001).

Com o aperfeiçoamento de que foi alvo ao longo dos tempos, o SVU é actualmente não só o mais utilizado e aceite em todo o mundo, como naturalmente o ponto de comparação para com todos os novos sistemas de votação que surgem.

### **2.4.1 Processo de votação em urnas**

A votação em urna é composta por um conjunto de fases requerendo determinados meios físicos e intervenientes claramente definidos sem os quais não se pode realizar. As principais fases e respectivos intervenientes no sistema eleitoral angolano são as seguintes:

- **Recenseamento eleitoral** - fase realizada antes do dia da votação em que os potenciais eleitores (que ainda não o tenham feito em votações anteriores) efectuem o recenseamento que lhes permite participar no acto eleitoral e que tem como resultado a lista das pessoas que estão habilitadas a exercer o direito de voto (eleitores). Nesta fase existe a interacção entre o recenseador e todos os potenciais eleitores que pretendem ficar, onde os que já possuem o registo de eleitor apenas são actualizados.
- **Votação** - fase principal de todo o processo, que define a data principal do processo de votação em curso. Esta fase é composta pela autenticação do eleitor em que se verifica se a pessoa que se apresenta para votar está elegível para o acto (apenas estas pessoas podem votar) e se ainda não o fez (só pode votar uma vez em cada votação), acto de votar por parte do eleitor e depósito na urna do seu voto. Nesta fase intervêm o eleitor que se apresenta para votar e os elementos da mesa eleitoral para a verificação da sua identidade e elegibilidade para votar, sendo o voto depositado na urna pelo próprio eleitor.
- **Contagem** - fase que geralmente é iniciada imediatamente no final do período estipulado para a votação e que começa pela recolha de todos os votos depositados na urna, procedendo-se depois à contagem manual dos votos válidos

em cada opção de votação. Neste processo participa o escrutinador, que procede à contagem dos votos, o presidente da mesa e delegados dos vários partidos.

- **Divulgação** - fase subsequente à contagem dos votos e que consiste na divulgação dos resultados eleitorais. Normalmente, é efectuada através de comunicação dos resultados locais/regionais a entidades superiores que fazem a gestão dos números de votos a nível nacional e a sua divulgação quando terminada a votação. A esta fase segue-se um período de tempo durante o qual os intervenientes nas eleições podem apresentar reclamações sobre os resultados ou o processo eleitoral. Esta actividade é normalmente realizada pelo presidente da mesa eleitoral.

São estas as várias fases apresentadas resumidamente que representam o processo normal da votação tal como ela se executa actualmente em Angola, com algumas implicações e requisitos a cumprir que serão explorados na secção seguinte.

#### **2.4.2 Requisitos gerais da votação em urnas**

Para se estar perante um sistema de votação de qualidade e que transmita a confiança necessária aos seus eleitores é essencial que o mesmo cumpra um importante grupo de requisitos. A lista seguinte condensa o conjunto de requisitos que devem ser cumpridos num SVU.

**Autenticidade** - indica que apenas pessoas autorizadas devem poder votar. Pretende-se que este requisito seja garantido pela passagem na mesa de voto onde o eleitor se autentica perante o presidente da mesa e os delegados e que mediante a documentação apresentada verificam se aquele se encontra autorizado a votar e se ainda não votou nessa votação.

**Singularidade** - indica que cada eleitor não vota mais do que uma vez. Este requisito é garantido pela mesa de voto onde os delegados registam os eleitores que já votaram nessa votação e facilmente se detecta se um eleitor tentar votar novamente.

**Anonimato** - indica que a associação entre o voto e a identidade do eleitor deve ser impossível, garantindo assim também a liberdade na votação. Este requisito é garantido pela câmara de voto (local resguardado onde apenas se encontra o eleitor e onde este assinala a sua opção de voto) e também pela urna fechada onde é colocado o voto junto dos restantes.

**Não-coercibilidade** - indica que não deve ser dada a possibilidade aos eleitores poderem provar qual foi a sua orientação de voto. Com este requisito pretende-se que seja

garantido pela câmara de voto, onde apenas se encontra o eleitor e não existe fisicamente mais ninguém que possa testemunhar a sua orientação de voto.

**Integridade do voto** - indica que os votos não podem ser modificados ou eliminados nem durante nem após o período eleitoral. Pretende-se que este requisito seja garantido durante o processo eleitoral pela urna fechada e, aquando da contagem dos votos, pela presença de elementos representantes das várias opções em votação.

**Verificabilidade/Auditabilidade** - indica que tem que se conseguir verificar que todos os votos foram contados correctamente e que o número de votos existentes corresponde ao número de pessoas que votaram, possibilitando a recontagem. Pretende-se que este requisito seja garantido por entidades reguladoras de todo o processo eleitoral e pela mesa de voto reunida antes e depois do acto eleitoral para análise.

Com os anos de aperfeiçoamento e pela quantidade de vezes que foi posto à prova, este sistema de votação (votação em urna) é hoje tido como o mais seguro, fiável e por isso mais utilizado a nível mundial e que transmite maior confiança aos eleitores. Todavia não existem sistemas totalmente seguros, pelo que o SVU, apesar de ser o que inspirará mais confiança, não é excepção, apresentando algumas limitações.

### 2.4.3 Vantagens e limitações

A VU é um sistema que apresenta já uma maturidade e quantidade de utilizações elevadas que lhe conferem um elevado grau de confiança à vista de todos. No entanto este sistema de votação apresenta ainda certas limitações, ao nível da garantia dos requisitos e na comparação com outros sistemas de votação, que dificilmente serão alvo de correção.

Começando pelas limitações e no que diz respeito ao cumprimento dos requisitos apresentados pode-se constatar o seguinte:

- A **não-coercibilidade**, que visa garantir que ninguém consegue provar em que opção votou (requisito com o propósito principal de evitar a venda de votos ou a violação da vontade individual de opção) é outro requisito que pode estar em causa no sistema de votação em urna. Apesar de, no momento exacto da votação, ser mais complicado provar qual a opção de voto, com o avanço das tecnologias e os dispositivos móveis de captura de imagens serem cada vez mais simples, qualquer pessoa, estando sozinha na câmara de voto, pode registar imagens do seu boletim de voto com a sua opção seleccionada, podendo até transmitir em

tempo real essas informações para o exterior para outras pessoas. Isto é um problema que põe em causa o referido requisito, e que também apresenta grandes dificuldades em ser combatido pois os dispositivos de captura de imagem são cada vez mais pequenos e indetectáveis. Porém este é um problema que alcança proporções ainda maiores quando o sistema é de votação electrónica, mais especificamente a remota.

- A **Integridade do voto** e a **Verificabilidade/Auditabilidade** visam garantir, respectivamente, que o conteúdo dos votos não é alterado nem eliminado durante e depois das votações, que o sistema pode ser auditado e que é possível provar que todos os votos foram devidamente contados. Estes requisitos têm a sua garantia de cumprimento nas mãos da mesa de voto e demais intervenientes nestas operações. Apesar de ser uma situação complexa, pois a mesa de voto é constituída geralmente por elementos afectos a —lados opostos‖ da votação, a verdade é que pode existir uma associação fraudulenta entre os vários intervenientes e até equipas externas de auditores que podem pôr em causa o cumprimento destes requisitos através da alteração dos votos em fases mais restritas das eleições, onde apenas estes elementos estão presentes. Esta é uma situação que se pode sempre melhorar com a introdução de novas pessoas, rotatividade de funções e aumento do número de presentes, mas a verdade é que existe sempre esta possibilidade.

Além destas questões relacionadas com os requisitos, a votação em urna apresenta também outras desvantagens em relação a outras arquitecturas. Alguns investigadores referem as seguintes:

- Votos contados manualmente proporcionam ao processo de contagem uma maior probabilidade de erro devido à intervenção humana (Frith 2007) e tornam o processo muito mais demorado – num SVE procede-se muito mais rapidamente à contagem dos votos e reduz-se o erro de contagem devido à intervenção humana na contagem;
- Requerem o transporte seguro dos boletins de voto, votos e outros documentos - SVEs não implicam grande parte desses custos;
- Preparação prévia dos locais de voto, distribuição dos boletins de voto pelas várias localidades – SVEs, principalmente remotos, não implicam esses custos.

- Os boletins de voto em papel —sofreml de falloff (há estudos que indicam que as opções colocadas nas posições finais do boletim têm menor probabilidade de serem escolhidas) (Costa 2006) - os SVE conseguem facilmente gerar boletins de voto com posições aleatórios para os candidatos, coisa que também poderia ser feito para os boletins de votação em papel mas que implica um esforço muito maior.
- É uma votação presencial o que faz com que as pessoas tenham que se deslocar a um local específico para poderem votar - esta é uma questão impossível de resolver neste tipo de arquitectura, sendo que apenas a VER apresenta uma alternativa a esta característica.

## **2.5 Votação electrónica**

O voto eletrónico consiste na utilização de sistemas informáticos que automatizem qualquer etapa ou processo de uma eleição (o recenseamento eleitoral, a votação, a transmissão dos resultados e a publicação dos resultados oficiais). Já existem várias implementações de sistemas de contagem de votos, sistemas de urnas automáticas, entre outros, nos países como Brasil, Portugal, Holanda, Suécia, e outros mais. Pensa-se que o primeiro processo de votação eletrónica terá acontecido por volta de 1960 (Bellis 2011) e desde então, com os avanços tecnológicos foi inevitável a introdução das tecnologias de informação nos processos de votação. A introdução dos meios tecnológicos foi surgindo como tentativa de responder a algumas das limitações apresentadas pelo SVU principalmente o tempo que demora a contagem dos votos e assim a divulgação dos resultados em eleições de grande dimensão (em países como Brasil, América ou Índia com um grande número de eleitores).

Ao longo desta secção serão apresentados os dois grupos de SVEs em que se foca este estudo:

- Sistema de votação electrónica presencial (SVEP) – SVE que tal como a VU implica a presença num local de voto previamente definido e devidamente controlado pela Comissão Eleitoral;
- Sistema de votação electrónica remota (SVER) – SVE que não implica a presença num local previamente definido, podendo a votação ser feita em qualquer local visto utilizar a internet como método de transmissão dos dados e comunicação entre o sistema e o eleitor.

Para tornar estes sistemas confiáveis perante os eleitores deve ser feita uma análise dos requisitos que estes sistemas devem cumprir e como o fazem. Nesta secção será apresentada uma definição dos requisitos que os sistemas de votação eletrónica devem cumprir com a inclusão de requisitos associados à componente tecnológica destes sistemas e uma análise de como os sistemas de votação apresentados cumprem, ou não, esses mesmos requisitos.

### **2.5.1 Votação eletrónica presencial**

A VEP é um sistema de votação que utiliza meios tecnológicos na totalidade ou em partes do processo de votação. O eleitor continua a ter que se deslocar a um local de voto previamente definido, tem as mesmas fases já apresentadas na votação em urna, mas a interação nessas fases pode ser feita através de meios tecnológicos, como a autenticação através da introdução de número de eleitor, palavra-chave e código de 2 fatores gerado pelo sistema, a escolha da opção poder ser realizada num ecrã ou dispositivo eletrónico que a transmite para um servidor onde este armazenará na sua base de dados, que armazena o voto, e que terminado o período de votação apresenta rapidamente os resultados.

Estes sistemas de votação têm sido alvo de muita investigação e investimento para conseguirem cumprir os requisitos dos sistemas de votação em urna, o que conduziu à elaboração de arquitecturas de grau de complexidade elevada para satisfazer um conjunto de requisitos, apesar de ainda porem em causa alguns requisitos importantes.

Os sistemas de votação baseados em votação eletrónica presencial apresentam um conjunto relevante de benefícios:

- Maior rapidez de contagem de votos – aspecto principal, não muito relevante em países com pequeno número de eleitores mas fundamental para países com número de eleitores elevado onde o tempo de contagem manual dos votos, os recursos que necessita para efectuar essa operação e a probabilidade da ocorrência de erros é muito elevada (ACE 2011; Cranor 1996; Freitas e Brandão 2008; Gaspar 2007);
- Maiores acessibilidades – pessoas com necessidades especiais podem votar sozinhas sem perda de privacidade, adicionalmente facilita a disponibilização de diferentes linguagens e formatos do boletim no ecrã (ACE 2011; Freitas e Brandão 2008);
- Redução de custos – principalmente ao nível da redução dos custos com pessoal pois envolve menos gente no processo que o sistema de votação em urna;
- Menor quantidade de erros na contagem dos votos;

- Possibilidade de dispor aleatoriamente as várias opções de voto no boletim evitando assim o falloff (Costa 2006);

### 2.5.2 Votação electrónica remota

A votação eletrónica remota (VER) utiliza um sistema de votação semelhante aos sistemas de votação eletrónica presencial diferenciando-se pelo facto de o eleitor estar perante uma interface baseado em web browsers ou um software específico instalado em dispositivos (computadores, telemóveis) ligados à internet para fazer a comunicação dos dados (tanto a identificação do eleitor como a sua opção de voto).

O procedimento de votação envolve sempre a identificação do eleitor perante o sistema (para averiguar se pode participar no ato eleitoral e se ainda não votou nessa eleição) e em seguida, mediante o sucesso desse passo, a escolha da sua opção de voto. Pensa-se que a primeira vez que a votação eletrónica remota foi exercida foi em 1997 por correio eletrónico pelo astronauta americano David Wolf, que se encontrava na estação espacial MIR, votando para as eleições do estado do Texas (Antunes 2007).

Os sistemas de votação eletrónica remota apresentam as seguintes vantagens:

- **Mobilidade** – permitem a votação realizada a partir de qualquer local com acesso à internet, sendo esta a principal vantagem destes sistemas, visando combater o desinteresse e abstenção nos actos eleitorais (ACE 2011; Antunes 2007);
- **Maior rapidez de contagem de votos** – tal como sucede com os sistemas de votação eletrónica presencial (ACE 2011; Cranor 1996; Freitas e Brandão 2008; Gaspar 2007);
- Maior acessibilidade - pessoas com necessidades especiais podem votar sozinhas sem perda de privacidade, diferentes linguagens e formatos do boletim no ecrã (ACE 2011; Freitas e Brandão 2008);
- **Redução de custos** – principalmente ao nível da redução dos custos com pessoal pois envolve menos gente no processo que o sistema de votação em urna e mesmo em relação aos sistemas de votação eletrónica presencial devido à utilização de canais de comunicação já existentes e à não necessidade de infra-estruturas (ACE 2011; Antunes 2007; Cranor 1996; Gaspar 2007);
- Redução da quantidade de erros na contagem dos votos (Freitas e Brandão 2008; Frith 2007);

- Possibilidade de dispor aleatoriamente as várias opções de voto no boletim evitando assim o falloff (Costa 2006).

De todas as vantagens apresentadas as mais importantes são sem dúvida a mobilidade, que permite ao eleitor votar de qualquer local recorrendo à internet, e a redução de custos em infra-estruturas e recursos humanos, algo que nenhum dos sistemas em análise permite tendo sempre que se deslocar a um local de voto físico e contactar com algumas pessoas torna o processo mais demorado e dispendioso para ambas as partes. Além disso, e especificamente em relação ao sistema de votação em urna, apresenta também a importante vantagem de proceder à contagem dos votos muito mais rapidamente.

### **2.5.3 Requisitos dos sistemas de votação electrónica**

Com a introdução de meios tecnológicos em alguma das fases do processo de votação surgiu um conjunto de novos requisitos a cumprir para se manter a qualidade de uma votação. Assim, os sistemas de votação electrónica devem cumprir não só os requisitos já apresentados anteriormente relativos a um sistema de votação em urna, como um conjunto de novos requisitos.

A lista seguinte apresenta 12 requisitos (e uma breve descrição dos mesmos) que foram identificados e considerados relevantes (através da revisão de literatura) assim como alguns dos principais autores que se referem aos mesmos e dissertaram sobre a sua importância (nota-se que a ordem pela qual são apresentados não traduz o seu grau de importância ou prioridade):

#### **Acessibilidade e Conveniência;**

- Acessível a pessoas com necessidades especiais.

#### **Anonimato;**

- Sistema de autenticação e recolha de votos separados.

#### **Auditabilidade e Certificabilidade;**

- Agentes independentes e externos.

#### **Autenticidade;**

- Autenticação do eleitor perante o sistema através do sistema de autenticação de 2 factores.

#### **Detectabilidade e Recuperabilidade;**



- Capaz de detectar erros, falhas ou ataques;
- Capaz de recuperar a informação de votação até ao momento da falha.

#### **Disponibilidade;**

- Sistema disponível ininterruptamente durante o período de votação.

#### **Integridade do voto;**

- O voto tem que ser correctamente registado e não pode ser alterado em nenhum momento.

#### **Mobilidade;**

- Factor crítico que põe em causa muitos sistemas mas é um acrescento de mais-valias inegável a qualquer serviço de votação electrónica.

#### **Não-coercibilidade;**

- Proibida a utilização de qualquer outro meio tecnológico;

#### **Singularidade;**

- Cada eleitor apenas pode votar uma vez.

#### **Transparência;**

- O sistema tem que ser transparente e confiável, principalmente à vista dos intervenientes na votação.

#### **Verificabilidade;**

- Existência de suporte físico dos votos para a recontagem.

Nas próximas secções apresenta-se detalhadamente cada um destes requisitos e a forma como estes podem ou não ser cumpridos nos dois sistemas de votação electrónica considerados.

### **2.5.3.1 Acessibilidade e Conveniência**

A acessibilidade e a conveniência são requisitos técnicos de um sistema de votação que garantem, respetivamente, que o sistema é acessível a todas as pessoas (incluindo as que têm necessidades especiais) e que é um acto onde o votante pode exercer o seu direito de voto rapidamente, numa só sessão e com o mínimo de equipamento e competências necessários. Estes dois requisitos encontram-se interligados e por isso se apresentam conjuntamente. Na votação em urna a conveniência é uma realidade pois é um processo relativamente rápido, que

não exige qualquer tipo de material ao eleitor e que este consegue efetuar sem necessidade de possuir grandes competências. Quanto à acessibilidade esta não é tão bem conseguida, sendo que as pessoas com necessidades especiais são obrigadas a ser acompanhadas por terceiros para poderem exercer o seu direito de voto (quebrando assim alguns requisitos como anonimato ou não-coercibilidade). No que toca aos sistemas de votação eletrónica estes tornam-se o inverso do sistema de votação em urna no que toca a estes requisitos. Por um lado, apresentam um conjunto de acessibilidades muito superior (como a capacidade para áudio e outros formatos de boletim de voto) principalmente no sistema de votação eletrónica remoto que, ao recorrer ao computador do eleitor, poderá utilizar as acessibilidades que este normalmente utiliza mantendo assim todos os requisitos que eram quebrados pela presença de outra pessoa no processo de votação. Por outro lado, principalmente no que respeita aos sistemas de votação eletrónica remota, apresentam-se menos convenientes pois necessitam não só de algum equipamento como já requerem um conjunto de competências na área das tecnologias de informação para se pode efetuar a votação. Neste aspeto os sistemas de votação eletrónica presencial apresentam alguma vantagem perante os remotos, pois não requerem meios técnicos para efetuar a votação, apesar de requererem um nível de competências necessárias para efetuar a votação também ele superior ao da votação em urna. Assim, os sistemas de votação eletrónica apresentam desvantagens ao nível da conveniência comparativamente com os sistemas de votação em urna, principalmente por necessidade de equipamentos e alguns conhecimentos para votar. Por outro lado, as acessibilidades representam uma mais-valia que os sistemas de votação eletrónica disponibilizam em relação ao sistema de votação em urna, pois melhoram indiscutivelmente as acessibilidades para os eleitores e tendem a tornar-se ainda de mais fácil acesso a todos com o choque tecnológico que vem acontecendo ao longo dos anos, o desenvolvimento das tecnologias e a adaptação das pessoas às mesmas.

#### **2.5.3.2 Anonimato**

O anonimato é o requisito do sistema de votação que garante que é impossível associar um determinado voto à pessoa que o efectuou. É mais um dos requisitos críticos pois é com ele que se garante o voto livre sem coação. Na votação em urna este requisito é garantido pela presença do eleitor sozinho na câmara de voto aquando da escolha do voto, e pelo depósito na urna onde junta o seu voto aos restantes, impossibilitando assim a associação daquele boletim, em tudo idêntico aos restantes, ao eleitor que lá o colocou.

Relativamente à votação electrónica remota, o anonimato é dos pontos mais críticos. Além dos pontos referidos na votação electrónica presencial, existem outros três factores que devem ser levados em consideração:

- As comunicações são feitas a partir de qualquer local com acesso à internet, sendo impossível garantir que a pessoa vota livremente sem estar sobre coação de outras pessoas que podem verificar qual a sua opção de voto. Esta é uma questão incontornável na votação remota.
- O canal de comunicação é a internet, e apesar de ser possível a encriptação dos dados e a utilização de ligações seguras, as comunicações entre a máquina do eleitor e o centro de votações podem ser interceptadas. Apesar de a informação estar cifrada o interceptor pode tentar proceder à sua decifração, podendo assim ver qual a opção de voto do eleitor. A solução deste problema passa pela utilização de ligações seguras e de cifras robustas nas mensagens trocadas.
- O instrumento de votação utilizado é tipicamente um computador, que pode ser vítima de um conjunto de ataques por vírus, trojan<sup>5</sup> ou spyware<sup>6</sup>, que podem monitorizar a máquina utilizada para a votação e assim ter acesso à opção de voto do eleitor. Para combater este aspecto apenas podem ser dadas algumas recomendações de protecção dos sistemas dos eleitores como sugere a California Internet Task Force com medidas como reinstalar o sistema operativo<sup>7</sup> e o software de voto (se existir software específico) antes de realizar a votação e utilizar equipamentos de segurança extra como firewall<sup>8</sup> e antivírus (Monteiro, et al. 2001).

### **2.5.3.3 Audibilidade e certificabilidade**

A audibilidade e a certificabilidade são requisitos técnicos de um sistema de votação que garantem que o sistema possa ser avaliado, testado e certificado, e se possa provar que todo o processo decorre dentro do previsto e especificado pelo sistema. Estes requisitos são muito importantes para a criação de uma imagem segura e fiável dos sistemas de votação. Para tal, existe um conjunto de recomendações que devem ser levadas em linha de conta e verificadas pelas entidades auditoras:

- Deve ser realizada por agentes externos;
- Os resultados destas actividades devem ser disponibilizados publicamente para poderem ser analisados pelos vários eleitores, aumentando assim não só o grau

de confiança (caso o sistema apresente a qualidade necessária) dos eleitores no sistema como a capacidade de detecção de falhas na própria auditoria, podendo ser posta à prova a integridade do sistema através dos resultados de actividades disponibilizados;

- Deve existir uma monitorização que se reflecta no registo de todos os eventos que ocorrem em todo o sistema, que para além do evento registe a hora a que este se realizou, possua mecanismos de protecção contra perdas de energia e de gravação em formatos que não permitam a sua alteração;
- Todos os sistemas de auditoria e certificabilidade devem observar as normas de boas práticas;
- Sempre que for detectado um erro devem ser tomadas as medidas de correção do mesmo e produzir-se um relatório que deve ser tornado público no final da votação;
- De forma aleatória, no dia das eleições, equipas de testes devem testar uma máquina para verificar se tudo corre como previsto;
- O sistema deve encontrar-se em local físico seguro, implementado em módulos separados e de preferência com os vários componentes separados também fisicamente;
- A auditoria e certificabilidade não podem colocar em causa os requisitos apresentados anteriormente nomeadamente a privacidade dos eleitores;

Com estas recomendações relacionadas com a auditoria e certificabilidade contribui-se não só para um sistema mais seguro, eficiente e eficaz, mas também para o melhoramento da imagem dos sistemas de votação electrónica e, consequentemente, para o fortalecimento da confiança dos intervenientes nesses sistemas.

#### **2.5.3.4 Autenticidade**

A autenticidade é o requisito de um sistema de votação que garante que apenas pessoas com direito a votar o fazem.

Relativamente à votação eletrónica presencial, o que se tem vindo a assistir é em tudo semelhante ao que acontece na votação em urna, tendo algumas arquiteturas recentemente desenvolvidas ignorado esta fase mantendo-a como anteriormente com a autenticação perante a mesa de voto. Com a introdução das tecnologias na votação é expectável que a autenticação possa ser feita também recorrendo às mesmas tecnologias, através da combinação de duas ou

mais formas de autenticação, pois atualmente é o método mais seguro de autenticação. Este processo eliminaria, assim, a necessidade de elementos (pelo menos de tantos elementos) na mesa de voto para a verificação da identidade do eleitor, reduzindo assim os custos.

#### **2.5.3.4 Detetabilidade e recuperabilidade**

A detetabilidade e a recuperabilidade são requisitos técnicos de um sistema de votação que garantem, respetivamente, que o sistema é capaz de detetar tentativas de intrusão de agentes externos, ataques e falhas do sistema, e que tem a capacidade de recuperar dessas situações sem perder a informação das votações já realizadas pondo em prática medidas de resolução da anomalia verificada. Estes dois requisitos não eram muito relevantes para a votação em urna porque o sistema é vigiado e monitorizado por várias pessoas que podem certificar-se sobre a existência ou não de falhas e ataques, tomar medidas para os combater, pelo que dificilmente a integridade do sistema está em causa. Quando se fala de sistemas de votação eletrónica, principalmente sistemas de votação eletrónica remota a questão é diferente, pois a deteção de erros, falhas ou ataques é mais complexa (Pereira 2006). Assim, os sistemas devem ser capazes de detetar esse tipo de ocorrências, registando no histórico de atividades/ocorrências a hora e tipo de ocorrência, e caso estejam preparados para responder à ocorrência pôr em prática medidas de correção dos erros, caso contrário deverão emitir alertas para a necessidade de atuação. A recuperabilidade entra também neste processo e terá que ser mantida sempre de modo a que até à ocorrência da situação anómala não se perca nenhuma informação. Este requisito pode ser cumprido localmente de forma fácil pela cópia periódica dos dados para outra máquina, no caso de se tratar de uma votação remota a complexidade de garantir que as informações são mantidas exige uma quantidade e complexidade de recursos mais elevada. Caso seja implementada a impressão dos boletins de voto em papel em adição à votação eletrónica e o sistema não consiga salvaguardar cópias dos dados existentes, estes votos em papel passarão a ser o único registo existente. É mais um requisito que requer grande quantidade de meios técnicos e muito difícil de garantir principalmente na votação eletrónica remota onde o sistema será mais facilmente alvo de ataques.

#### **2.5.3.5 Disponibilidade**

A disponibilidade é um requisito técnico de um sistema de votação que garante que o sistema consegue estar sempre disponível durante o período eleitoral. Este é um requisito que na votação em urna é mantido pela disponibilidade do local de voto e durante o horário previsto dos meios necessários a que qualquer cidadão consiga realizar uma votação. Nos sistemas de votação eletrónica presencial, apesar de se levantarem algumas questões técnicas relacionadas

com a quantidade de vezes que o sistema vai ser utilizado num curto espaço de tempo e os recursos que necessita para armazenar uma tão grande quantidade de informação, este requisito pode ser garantido pelo sistema. O grande problema de disponibilidade coloca-se quando a arquitetura em análise é uma arquitetura de votação eletrónica remota. Aí será necessária uma grande estrutura de servidores e respetivos componentes para suportar a comunicação de tantos eleitores que pretendem efetuar a votação sem nunca negar a disponibilidade da aplicação, mesmo momentaneamente.

#### **2.5.3.6 Integridade do voto**

A integridade do voto é um dos requisitos de um sistema de votação que garante que os votos não são modificados ou destruídos durante ou após o processo eleitoral. Este requisito é garantido na votação em urna pela presença de uma urna fechada, e durante o processo de contagem pela presença de várias entidades independentes. É um requisito que pode ser cumprido desde que não haja erro humano.

Relativamente à votação eletrónica presencial, a integridade do voto tem uma abordagem totalmente distinta da votação em urna, pois durante a votação os votos são armazenados numa —urna eletrónica, e o processo de contagem não tem intervenção humana, reduzindo o risco de associação entre as pessoas que fazem e assistem à contagem dos votos. Assim, manter a integridade dos votos numa votação eletrónica presencial baseia-se em ter sistemas confiáveis e transparentes que não alterem os dados de votação sob pretexto algum. É um requisito essencial e que pode ser cumprido neste sistema de votação.

Relativamente à votação eletrónica remota, a integridade do voto depois deste se encontrar no sistema levanta as mesmas questões que em sistemas de votação eletrónica presencial. Contudo, existe um acréscimo de dificuldade em garantir a integridade dos votos durante as trocas de mensagens entre a máquina do eleitor e o servidor de votação, que assentam na utilização da internet como canal de comunicação. A questão é semelhante à do anonimato, em que a interceptação das mensagens trocadas, alteração do seu conteúdo e reenvio para o legítimo receptor, bem como a existência de software destinado a alterar ou apagar o voto realizado nessa máquina, põem em risco a integridade dos dados. Esta questão pode ser resolvida por elementos técnicos, principalmente pela existência de ligações seguras entre a máquina do eleitor e do sistema de voto.

#### **2.5.3.7 Mobilidade**

A mobilidade é um requisito técnico de um sistema de votação que garante que o sistema não apresenta restrições logísticas no que concerne ao local em que o eleitor pode exercer o seu direito de voto. Este requisito consiste em dar a total liberdade ao eleitor de escolher o local em que pretende realizar a votação. É um requisito que não se verifica inteiramente na votação em urna, pois apesar de ser uma votação em que é instalado um posto de votação em cada freguesia o eleitor tem sempre que se deslocar a esse local (adicionando a isso o problema de ter que se deslocar especificamente a uma freguesia).

Com o aparecimento dos sistemas de votação electrónica, principalmente os sistemas de votação electrónica remota, que utilizam a internet como canal de comunicação entre o centro de votações e o eleitor, a mobilidade passou a ser uma realidade, podendo o eleitor votar a partir de qualquer local onde possua o acesso à rede. Todavia, este requisito representa também muitos riscos para os requisitos apresentados anteriormente como o anonimato e a integridade do voto e para um outro requisito a discutir seguidamente, a não-coercibilidade.

#### **2.5.3.8 Não-coercibilidade**

A não-coercibilidade é o requisito de um sistema de votação que garante que o eleitor não consegue provar qual a sua opção de voto. É um requisito garantido na votação em urna pela presença do eleitor sozinho na câmara de votação apesar do já identificado problema que surge com a banalização dos meios de captura de imagem.

Relativamente à votação eletrónica presencial, a não-coercibilidade apresenta os mesmos problemas identificados no capítulo anterior para a votação em urna, ou seja, a evolução e simplicidade dos dispositivos móveis de captura de imagens dão ao eleitor a capacidade de captar a sua opção de voto e provar em que opção votou.

Relativamente à votação eletrónica remota, a não-coercibilidade torna-se ainda mais crítica, pois a votação pode ser feita à vista de outras pessoas e assim provar facilmente a opção de voto.

Esta é uma questão incontornável em todos os sistemas, e se é verdade que no sistema de votação eletrónica remota existe uma maior facilidade de quebra deste requisito, não deixa de ser verdade que hoje em dia, aplicando-se também aos outros sistemas de votação, quem o pretender fazer consegue-o com relativa facilidade. Trata-se de um requisito crítico, pois o seu não cumprimento pode levar a uma questão muito complexa que é a comercialização de votos. A sua implementação é atualmente impossível já que seria necessário conseguir evitar que as pessoas levassem para a câmara de votação dispositivos tecnológicos de recolha de imagens

que é algo complexo e que implicaria outras questões de logística e burocracia do processo, tornando-o assim passível de ser quebrado em qualquer tipo de sistema de votação dos apresentados.

### **2.5.3.9 Singularidade**

A singularidade é o requisito do sistema de votação que garante que cada voto conta apenas uma vez e cada eleitor só consegue exercer uma vez o direito de voto. Este requisito é satisfatoriamente cumprido na votação em urna (partindo do princípio que as listas de eleitores estão correctamente definidas), e nos sistemas de votação electrónica acontece a mesma situação.

Relativamente à votação electrónica presencial, a singularidade consegue ser mantida desde que existam listas de eleitores correctamente definidas, através da verificação nas listas se o eleitor se encontra elegível para votar e se já votou ou não. Estes dois pontos de verificação essenciais são facilmente implementáveis no software garantindo assim a singularidade no voto.

Relativamente à votação electrónica remota, a singularidade apresenta os mesmos controlos e as mesmas soluções da votação electrónica presencial, aumentando um pouco a complexidade, face ao problema de concorrência em que uma pessoa tenta votar em mais de uma máquina ao mesmo tempo, mas que pode ser facilmente solucionado pela integridade na base de dados, desde que se aceite apenas o primeiro voto.

### **2.5.3.10 Transparência**

A transparência é um requisito técnico de um sistema de votação que garante que o sistema é claro em todo o seu funcionamento e consegue transmitir rigor, precisão e segurança aos eleitores. Este requisito não é difícil de demonstrar e manter no processo de votação em urna pois todas as fases são já conhecidas e verificáveis pelos eleitores. No que toca à votação electrónica, este requisito assume um papel essencial na medida em que a introdução das tecnologias e a desmaterialização torna o processo menos claro e verificável por parte do eleitor, principalmente no que toca à votação remota onde a percepção do utilizador sobre todo o processo é muito reduzida, assumindo uma natural posição de reserva perante o sistema. Para contribuir para a transparência e ao mesmo tempo transmitir uma imagem de confiança do sistema são efectuadas as seguintes recomendações:



- As especificações do software bem como documentação relativa ao hardware devem ser disponibilizadas livremente, possibilitando assim a análise e verificação de como funciona todo o processo de votação. Uma outra recomendação prende-se com a não alteração do sistema nos seis meses anteriores à data das eleições;
- Em todas as fases da criação e implementação do software devem ser levadas em consideração as boas práticas de desenvolvimento;
- A comunidade científica e especialistas da área da tecnologia devem ser consultados durante todo o processo de desenvolvimento e implementação de arquitecturas;
- Deve ser impresso um voto em papel (nas condições já referidas anteriormente) para tentar materializar um pouco o processo;
- Deve apresentar uma arquitectura simples que seja facilmente compreensível pela população em geral.

O cumprimento deste requisito é muito importante, mas a sua satisfação é difícil, apesar de ter um conjunto de recomendações que podem ser levadas em linha de conta. Apesar de todas estas preocupações, anomalias (casuais, falhas de hardware) vão sempre existir e sem se estar na presença de um sistema aberto e transparente é impossível detectar o que foi um erro inocente, uma anomalia ou uma tentativa de fraude. Com o cumprimento deste requisito a confiança e transparência neste tipo de sistemas por parte dos eleitores tenderá a aumentar.

#### **2.5.3.11 Verificabilidade**

A verificabilidade é o requisito de um sistema de votação que garante que o eleitor consegue verificar que o seu voto é realmente contado e que no final da votação se consiga verificar que o resultado é o correto, através da recontagem dos votos.

Com estas medidas referidas pretende-se reforçar a segurança do sistema de votação eletrónica, mas um dos fatores muito importante para a utilização do sistema é a confiança, portanto este (confiança) é o próximo ponto a ser retratado.

#### **2.5.4 Confiança**

A confiança é o aspecto basilar deste trabalho de investigação. Como já foi referido, sem confiança nenhum sistema de votação conseguirá vingar e qualquer desconfiança que recaia sobre os sistemas de votação descibiliza e pode pôr em causa as eleições e respectivos resultados.

Nesta secção concentra-se a atenção sobre o que representa o acto de confiar, de que depende, como esta é percebida e aplicada nos sistemas de votação electrónica, e como será avaliada no âmbito deste estudo.

#### **2.5.4.1 Confiança nos sistemas de votação electrónica**

Nos sistemas de votação em geral, a confiança representa a certeza, por parte de todos os intervenientes eleitorais, que todo o processo decorre dentro dos pressupostos desejados, mais concretamente no que toca aos requisitos que os sistemas de votação devem cumprir, atestando assim a qualidade do mesmo e garantindo o cumprimento de condições de segurança. Por muitos e complexos elementos técnicos que sejam introduzidos no processo de votação, ainda que assegurem o cumprimento dos requisitos de segurança e até possuam um conjunto de vantagens relevantes em relação ao sistema de votação existente, se não se conseguir transmitir essa mensagem, provando que este novo sistema de votação cumpre de facto os requisitos necessários, dados os erros e falhas que já aconteceram com a utilização de sistemas de votação electrónica, estes vão continuar a ser contestados e levantar desconfianças que corroem qualquer sistema de votação e consequente resultado eleitoral. Tão importante como cumprir os requisitos é transmitir, através de provas irrefutáveis como estes são cumpridos. Além disso, a aceitação dos sistemas de votação electrónica deve ser transversal à sociedade, a todos os eleitores e não apenas àqueles que estão predispostos a utilizá-la, pois caso se mostre que existe um grande número de eleitores cépticos em relação a esse método de votação, a confiança na democracia pode estar comprometida.

Além destes dados sobre as variações da confiança em diferentes faixas etárias da população foram sendo também identificados alguns factores que se consideram como relevantes para fazerem variar a confiança em sistemas de votação electrónica:

- **Informação disponibilizada** – a quantidade e qualidade da informação disponibilizada e as sessões de esclarecimento sobre o SVE podem influenciar a confiança dos eleitores no sistema;
- **Historial de utilizações** – a quantidade de utilizações do SVE sem apresentar erros relevantes pode influenciar a confiança dos eleitores no sistema;

- **Testes realizados** – a quantidade e qualidade dos testes realizados ao sistema podem influenciar a confiança dos eleitores no sistema;
- **Auditorias e Certificações** – a existência de auditorias e certificações desde o processo de desenvolvimento até aos actos eleitorais pode influenciar a confiança dos eleitores no sistema;
- **Equipa de desenvolvimento** – o reconhecimento e reputação da equipa de desenvolvimento do sistema perante os eleitores pode influenciar a confiança dos eleitores no sistema;
- **Capacidade de cumprir requisitos** – a capacidade que um sistema tem de cumprir requisitos, principalmente requisitos que mexam com os direitos de anonimato do eleitor durante o acto de votação pode influenciar a confiança dos eleitores no sistema;

### **2.5.5 Desvantagens do sistema de votação electrónica**

Os SVEs têm vindo a ganhar espaço e cada vez mais experiências são realizadas com novos softwares e máquinas que pretendem automatizar o processo de votação, no entanto, todo este tempo de experiência e testes ainda não foi capaz de satisfazer a necessidade de um sistema de qualidade e confiável, principalmente por um aspecto muito relevante: uma alteração/ataque no sistema de votação em urna afecta um número de voto muito limitado, enquanto que se considera sistemas de votação electrónica uma simples alteração pode mudar o conteúdo de milhares de votos. Além das referidas limitações o contexto em que os SVEs são utilizados torna-os alvos ainda mais apetecíveis por parte de potenciais atacantes, o que faz prevalecer desconfiança, principalmente quando existe o espectro da possibilidade de existir um “negócio” tão rentável como seria a compra de votos.

#### **2.5.5.1 Falhas de sistemas de votação electrónica**

Como apresentado os sistemas de votação electrónica apresentam ainda um conjunto de limitações que podem comprometer os resultados e consequentemente todo o processo eleitoral. Têm vindo a ser identificados um conjunto de potenciais problemas que podem afectar um sistema de votação electrónica:

##### **Configurações e dados das eleições:**

- Alterações das configurações do sistema levando a boletins de voto confusos (Amari, et al. 2007; Kohno, et al. 2004; Schneier 2004<sup>a</sup>);
- Alterações das definições de voto (Kohno, et al. 2004);

- Alteração do resultado da eleição e ligação do voto com o respectivo eleitor (Kohno, et al. 2004); o Ataques no início das eleições (Kohno, et al. 2004);
- Alterações a configuração através do código fonte (Kohno, et al. 2004);
- Processo do código – alterações que podem levar à leitura errada da escolha do eleitor por parte do computador (Kohno, et al. 2004; Saltman 1993; Schneier 2004);

Estas falhas podem acontecer devido a erro ocasional ou serem premeditadas e terem origem em tentativas de fraude, residindo aí mais uma questão de muito complexa resolução, perante que tipo de ataque estamos, se é de origem intencional ou não (Schneier 2004b).

## **2.6 Sistemas de informação**

Um Sistema de Informação (SI) é um conjunto de elemento inter-relacionados cujo objetivo é coletar, armazenar, tratar e fornecer informações de tal modo a apoiar as funções ou processos de uma organização no controlo e tomada de decisões ou para disponibilizar informações para fins educacionais ou simplesmente informativos. (Gouveia, 2009)

Um SI é composto de um subsistema social e de um subsistema automatizado. O primeiro inclui as pessoas, processos, informações e documentos. O segundo consiste dos meios automatizados (máquinas, computadores, redes de comunicação) que interligam os elementos do subsistema social. (Gouveia, 2009)

As pessoas, juntamente como os processos que executam e com as informações e documentos que manipulam, também fazem parte do SI. O SI é algo maior que um software, pois além de incluir o hardware e o software, também inclui os processos (e seus agentes) que são executados fora das máquinas. (Gouveia, 2009)

Os SI possuem as seguintes vantagens: Otimização do fluxo de informação permitindo maior agilidade e organização; Redução de custos operacionais e administrativos e ganho de produtividade; Maior integridade e veracidade da informação; Maior estabilidade; Maior segurança de acesso à informação. (Gouveia, 2009)

## **2.7. Internet**

O termo internet é muito utilizado para descrever uma rede em que tudo se pode e tudo se consegue. Essa popularização está relacionada à sua larga utilização por utilizadores com ou sem experiência na área de informática, ou seja, qualquer pessoa com um computador

conectado a um modem, com uma identificação, pode navegar pela rede. A internet trouxe a todas as áreas a possibilidade de compartilhar conhecimento e muito entretenimento.

## 2.8 Site

Um **site** ou **sítio** é um conjunto de páginas Web, isto é, de hipertextos acessíveis geralmente pelo protocolo HTTP na Internet. O conjunto de todos os sites públicos existentes compõe a World Wide Web. As páginas num site são organizadas a partir de um Uniform Resource Locator (URL- Localizador de Recursos Universal) básico, onde fica a página principal, e geralmente residem no mesmo diretório de um servidor. As páginas são organizadas dentro do site numa hierarquia observável no URL, embora as hiperligações entre elas controlem o modo como o leitor se apercebe da estrutura global, modo esse que pode ter pouco a ver com a estrutura hierárquica dos arquivos do site

Sites são escritos em, ou dinamicamente convertidos para HTML e o acesso é feito usando um software cliente chamado Web browser ou navegador. Sites consistem de páginas HTML estáticas ou páginas criadas dinamicamente usando tecnologias como JSP, PHP ou ASP.NET. Um site também requer um software conhecido como servidor web, como o Apache, Xampp, IIS entre outros. Frequentemente sites possuem também conteúdo armazenado em base de dados.

## 2.9 Base de dados

Segundo Campos (2007), as bases de dados têm como finalidade principal armazenar informações e permitir que os utilizadores busquem e atualizem estas informações de forma eficiente e rápida quando necessário. E apresenta as seguintes vantagens para as bases de dados:

- **Organização dos dados:** a estrutura dos dados é organizada de formas a facilitar a sua manipulação
- **Volume de dados:** permitem o armazenamento de grandes volumes de dados
- **Velocidade/Eficiência:** os dados são obtidos e atualizados rapidamente no BD
- **Concorrência:** diferentes aplicações/usuários podem compartilhar os mesmos dados
- **Redundância controlada:** Pode reduzir ou eliminar a redundância dos dados, pois os dados vão estar armazenados em apenas um local.
- **Aplicabilidade:** Diminuição no tempo de desenvolvimento da aplicação.

Com o passar do tempo surgiram vários modelos tal como: o modelo em rede, hierárquico, relacional que é uma teoria matemática desenvolvida pelo famoso Edgar Frank Codd, temos também o modelo orientado a objeto que surgiu por ultimo, mas ainda assim o modelo relacional é o mais utilizado.

### 2.9.1 Sistemas de gestão de base de dados

Devido às vantagens em manter um único conjunto lógico e organizado de dados, completamente autónomos das aplicações que os processam, torna-se necessário desenvolver um mecanismo que faça a gestão desse conjunto, garantindo que todas as solicitações do nível aplicacional à base de dados passem por si. Este mecanismo é o sistema de base de dados (SGBD) (Pereira, 1998).

Por definição, o SGBD é um conjunto de softwares, destinado a gerir todo o armazenamento e manipulação dos dados do sistema, fazendo a interface entre o nível aplicacional e a base de dados propriamente dita (Pereira, 1998).

A principal função dos SGBDs é permitir que vários utilizadores e aplicações possam aceder e/ou manipular concorrentemente dados armazenados na BD, ficando o SGBD responsável por garantir que a coerência dos dados da BD é preservada. Mais concretamente, as operações pedidas pelos diferentes utilizadores/aplicação são enviadas no contexto de transações e os SGBDs asseguram que a execução de cada transação satisfaz quatro propriedades normalmente designadas por ACID (atomicidade, coerência, isolamento e persistência – (durability)), (Correia e Sousa, 2010).

- **Atomicidade:** garante que a execução de uma transação só pode ter um de dois possíveis desfechos – Transação terminada com sucesso ou a transação é abortada e nenhuma é executada.
- **Coerência:** assegura que uma transação seja executada de forma coerente deixa a base de dados num estado coerente, quer seja terminada com sucesso ou abortada.
- **Isolamento:** garante que apesar de eventuais execuções concorrentes de várias transações, cada transação não vê os dados alterados por outras transações ainda não terminadas com sucesso.
- **Persistência:** garante que as ações efetuadas por uma transação com sucesso não podem ser desfeitas.

Uma transação é uma unidade lógica de trabalho. Ou todo o trabalho é concluído como uma unidade inteira, ou nenhum trabalho é concluído.

Em projetos de pequena dimensão, recorre-se normalmente a SGBD de pequeno porte, tais como o Access e o Visual FoxPro. Em estruturas organizacionais mais complexas e que requerem uma política de segurança mais apertada e uma utilização mais intensiva, recorre-se aos sistemas Oracle, SqlServer, FireBird, Mysql e ASE (mais conhecido por Sybase), que possuem uma maior capacidade de armazenamento e uma maior velocidade de acesso à informação. Estes últimos SGBD são indicados para o funcionamento em rede e em ambientes multiutilizadores.

### 2.9.2 SQL

A sigla SQL abrevia o termo Structured Query Language, que em português significa linguagem de consulta estruturada, que é o nome dado a linguagem responsável pela interação com as bases de dados relacionais.

O SQL foi desenvolvido originalmente nos anos 60, nos laboratórios da IBM e se tornou num grande padrão nas bases de dados, isto decorre da sua simplicidade e facilidade de uso, ela se diferencia das outras linguagens de consulta a base de dados, no sentido de que uma consulta SQL especifica a forma do resultado e não o caminho para chegar a ele, ela é uma linguagem declarativa em oposição as linguagens procedurais. Não obstante o SQL que é o mais utilizado, existe também o NoSql que se reflete em base de dados não-relacionais. Os recursos disponibilizados por essa linguagem (SQL) podem ser agrupados em 5 funcionalidades que juntas formam o SQL, tal como são apresentadas abaixo:



**Figura 1:**Linguagens do SQL

**Fonte:** [www.softblue.com.br](http://www.softblue.com.br), Fevereiro, 2017.

### 2.10. Plataformas de desenvolvimento web

Plataformas de desenvolvimento Web são aquelas cujo seu foco principal de desenvolvimento é a internet, são usadas normalmente para desenvolvimento de sites e aplicações Web. Existe no mercado uma grande diversidade destas plataformas, a título de exemplo podemos citar as seguintes: JAVA, PHP e ASP.NET sendo que esta última merecerá uma abordagem mais profunda pois é a plataforma de eleição para o desenvolvimento do nosso projecto.

### 2.10.1 ASP.NET MVC

É um framework da Microsoft que possibilita o desenvolvimento de aplicações web, fazendo uso do padrão arquitetural MVC (Model-View-Controller, ou Modelo-Visão-Controlador, em português). Embora o ASP.NET MVC faça uso deste padrão, ele não define uma arquitetura de desenvolvimento por si só.

O padrão MVC busca dividir a aplicação em responsabilidades relativas à definição de sua sigla. A parte do Modelo trata as regras de negócio, o domínio do problema, já a Visão busca levar ao usuário final informações acerca do modelo, ou solicitar dados para registros. Desta maneira, o ASP NET MVC busca estar próximo a este padrão. Ele traz, em sua estrutura de projeto, pastas que representam cada camada do MVC.

### 2.11 Domain Drive Design

Domain Drive Design significa projecto orientado ao domínio, é uma abordagem de desenvolvimento de software, que quando aplicada resulta numa de terminada estrutura do ponto de vista arquitectural do nosso sistema, ele veio do título do livro de Eric Evans, que na qual o mesmo livro é baseado na experiência do autor ao longo de mais de 20 anos de desenvolvimento de software, utilizando técnicas de orientação a objetos.

#### 2.11.1 Linguagem Ubíquia

Linguagem comum, com termos bem definidos, que fazem parte do domínio do negócio e que são usados por todas as pessoas que fazem parte do processo de desenvolvimento de software. Nessa linguagem estão termos que fazem parte das conversas diárias entre especialistas de negócio e times de desenvolvimento. Todos devem usar os mesmos termos tanto na linguagem falada quanto no código. Isso significa que, se durante uma conversa com um cliente do sistema de cobrança, por exemplo, ele disser: “Temos que emitir a **fatura** para o **cliente** antes da **data limite**”, vamos ter no nosso código alguma coisa do tipo:

- Uma classe para a entidade Cliente;
- Uma classe para a entidade Fatura;
- Algum serviço que tenha um método emitir;
- Algum atributo com o nome de data limite.

Essa linguagem ubíqua deve ser compreendida por todos e não pode haver ambiguidades. Toda vez que alguém perceber que um determinado conceito do domínio possui várias palavras

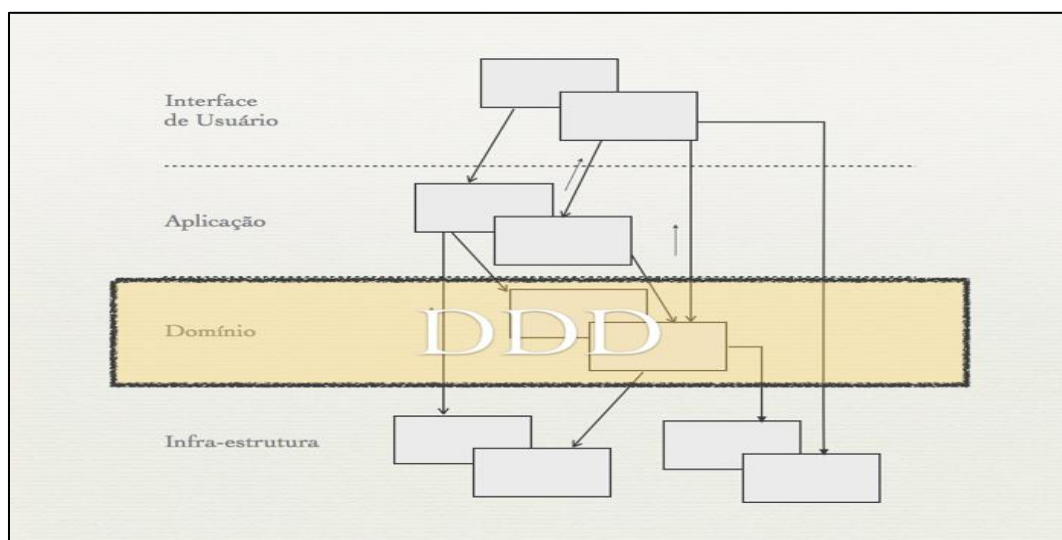


que o represente, essa pessoa deve tentar readequar tanto a linguagem falada e escrita, quanto o código.

Em um time que cria software temos de um lado os especialistas de negócio e de outro os desenvolvedores e arquitetos. Num processo ágil defendido pelo MDD a criação do modelo abstrato deve ser feita em grupo, com todas as pessoas juntas.

Uma vez que decidimos criar um modelo usando MDD, precisamos, inicialmente, isolar o modelo de domínio das demais partes que compõem o sistema. Essa separação pode ser feita utilizando-se uma arquitetura em camadas, que dividirá nossa aplicação em quatro partes:

- **Interface de Usuário** – parte responsável pela exibição de informações do sistema ao usuário e também por interpretar comandos do usuário;
- **Aplicação** – essa camada não possui lógica de negócio. Ela é apenas uma camada fina, responsável por conectar a Interface de Usuário às camadas inferiores;
- **Domínio** – representa os conceitos, regras e lógicas de negócio. Todo o foco de DDD está nessa camada;
- **Infra-estrutura** – fornece recursos técnicos que darão suporte às camadas superiores. São normalmente as partes de um sistema responsáveis por persistência de dados, conexões com base de dados, envio de mensagens por redes, gravação e leitura de discos, e outros mais.



**Figura 2:**Arquitectura em camadas, utilizada para separar o domínio do resto da aplicação

**Fonte:** [www.agileandart.com](http://www.agileandart.com), Abril, 2017.

Depois de dividirmos o sistema em camadas, nos preocuparemos apenas com a camada de domínio. Para modelar essa parte, utilizamos alguns Padrões propostos em DDD. Esses padrões

são chamados de blocos de construção e serão utilizados para representar nosso modelo abstrato. Esses blocos podem ser:

- Entidades – classes de objetos que necessitam de uma identidade. Normalmente são elementos do domínio que possuem ciclo de vida dentro de nossa aplicação: um Cliente, por exemplo, se cadastra no sistema, faz compras, se torna inativo, é excluído, etc.;
- Objetos de Valores – objetos que só carregam valores, mas que não possuem distinção de identidade. Bons exemplos de objetos de valores seriam: strings, números ou cores. Por exemplo: se o lápis de cor da criança acabar e você der um novo lápis a ela, da mesma cor, só que de outra caixa, ela não vai se importar. Para a criança, o lápis vermelho de uma caixa é igual ao lápis vermelho de outra caixa.
- Agregados – compostos de Entidades ou Objetos de Valores que são encapsulados numa única classe. O Agregado serve para manter a integridade do modelo. Elegemos uma classe para servir de raiz do Agregado. Quando algum cliente quiser manipular dados de uma das classes que compõem o Agregado, essa manipulação só poderá ser feita através da raiz;
- Fábricas – classes responsáveis pelo processo de criação dos Agregados ou dos Objetos de Valores. Algumas vezes, Agregados são relativamente complexos e não queremos manter a lógica de criação desses Agregados nas classes que o compõem. Extraímos então as regras de criação para uma classe externa: a fábrica;
- Serviços – classes que contém lógica de negócio, mas que não pertence a nenhuma Entidade ou Objetos de Valores. É importante ressaltar que Serviços não guardam estado, ou seja, toda chamada a um mesmo serviço, dada uma mesma pré-condição, deve retornar sempre o mesmo resultado;
- Repositórios – classes responsáveis por administrar o ciclo de vida dos outros objetos, normalmente Entidades, Objetos de Valor e Agregados. Os repositórios são classes que centralizam operações de criação, alteração e remoção de objetos. Em linguagens como Java e .NET, repositórios são comumente implementados usando-se frameworks como Hibernate ou Nhibernate. Já em RubyOnRails, o ActiveRecord faz o papel de repositório;
- Módulos – abstrações que têm por objetivos agrupar classes por um determinado conceito do domínio. A maioria das linguagens de programação oferecem suporte a módulos (pacotes em Java, namespaces em .NET ou módulos em Ruby). Um anti-

padrão comum é a criação de módulos que agrupam as classes segundo conceitos de infra-estrutura. Um exemplo seria, ao se trabalhar com Struts, em Java, criar um pacote que conterá todas as Actions do sistema. Ao usar DDD devemos agrupar classes se esse agrupamento faz sentido do ponto de vista do domínio, ou seja, do negócio. Se tivermos, por exemplo, várias classes que compõem informações de Paciente num sistema médico, podemos criar um módulo chamado paciente e colocar classes como Ficha, PrescricaoMedica, RegistroDeConsulta e HistoricoDeCirurgias num mesmo pacote.

### **2.11.2 Camada de anti-corrupção**

Quando temos um sistema legado, com código muito bagunçado e uma interface complexa, e estamos escrevendo um sistema novo com o código razoavelmente bem feito, criamos uma camada entre esses dois sistemas. O nosso sistema novo e bem feito falará com essa camada, que possui uma interface bem feita. E a camada anti-corrupção é responsável por traduzir e adaptar as chamadas para o sistema legado, usando uma fachada interna.

### **2.12 Integração de tecnologias**

O acesso a informações relevantes é de valor inestimável para os gestores das organizações e está baseado em sistemas de informação. Para gerar estas informações é necessário combinar dados armazenados em vários sistemas que precisam integrar-se para que informações novas e relevantes sejam geradas. As técnicas utilizadas para integração pelos desenvolvedores e organizações produtoras de sistemas de informação sustentam-se num paradigma consolidado pela engenharia de software denominada Enterprise Application Integration (EAI).

A integração dos dados é uma exigência constante em algumas organizações visto que nem todos os processos de negócio são atendidos pelo ERP. No cenário empresarial, de constante necessidade de integração externa com parceiros, clientes e fornecedores ou ainda aqueles oriundos de fusões e aquisições entre corporações, vê-se sistemas de diferentes fornecedores e tecnologias concebidos em momentos diferentes que acabam por gerar um ecossistema de dados descentralizado e heterogêneo (BERGAMASCHI, 2004, p. 13).

### **2.12.1 ASP.NET WEB API**

ASP.Net Web API é um framework que facilita a construção de serviços REST HTTP que alcançam uma grande variedade de clientes incluindo Mobile, Browsers e aplicações locais. É a plataforma ideal para construção de serviços REST baseados em .Net.

Serviços de internet são populares já faz um bom tempo, WebServices foram desenvolvidos e consumidos durante longos anos sendo uma tecnologia livre de plataforma, ou seja, aplicações .Net e Java se comunicam por WebServices sem dependência de tecnologias. Nos anos seguintes a Microsoft lançou o Remoting e por fim o famoso WCF que engloba (HTTP, TCP, MQ).

Apesar do Remoting e WCF serem tecnologias Microsoft o que WebServices, Remoting e WCF tem em comum? Todos eles são baseados em SOAP (Simple Object Access Protocol). O SOAP é baseado em XML e busca padronizar o conteúdo que é trafegado entre as pontas. O problema do SOAP é que algumas plataformas não conseguiram acompanhar sua evolução e a adoção se tornou complicada devido sua implementação, por este motivo começaram a surgir soluções alternativas mais simples.

Uma solução alternativa ao SOAP e altamente adotada é o REST (Representational State Transfer), baseado totalmente em HTTP e seus recursos. Para o melhor entendimento sobre ASP.Net Web API e REST sugiro conhecer um pouco sobre o protocolo HTTP.

Grandes empresas como Google, Facebook, LinkedIn, Netflix entre diversas outras já disponibilizam APIs Web para serem consumidas, todas elas são baseadas em HTTP com REST.

O ASP.Net Web API utiliza HTTP com REST e diferente do SOAP não depende de XML para trafegar as informações, o formato padrão para isso é JSON (Java Script Object Notation).

JSON é um formato amplamente utilizado em qualquer plataforma (não apenas .Net), é um subconjunto da notação de objeto de JavaScript, mas seu uso não requer JavaScript exclusivamente. Isso proporciona um potencial muito grande para os serviços HTTP, pois é um formato intercambiável, leve e livre de plataformas.

É possível retornar XML com Web API ao invés de JSON ou outros formatos como CSV. Também é possível criar um formato próprio para retornar os dados, tudo fica a critério do desenvolvedor.

### **2.12.2 Google Maps API**

Google Maps é um serviço de pesquisa e visualização de mapas e imagens de satélite da Terra gratuito na web fornecido e desenvolvido pela empresa estadunidense Google.

Atualmente, o serviço disponibiliza mapas e rotas para qualquer ponto nos Estados Unidos, Canadá, na União Europeia, Austrália e Brasil, entre outros. Disponibiliza também imagens de satélite do mundo todo, com possibilidade de um zoom nas grandes cidades, como Nova Iorque, Paris, São Paulo, Rio de Janeiro, Brasília, entre outras. Nos Estados Unidos o Google fez uma parceria com órgãos públicos, que incluirão as linhas de trem americanas e seus cruzamentos com rodovias ao Google Maps.

Juntamente com o lançamento da versão brasileira do Google Maps, a empresa introduziu o Local Business Center, ferramenta que permite com que qualquer empresa faça seu cadastro e seja então encontrada no Google Maps por qualquer usuário. No cadastro as empresas podem preencher seus dados cadastrais, horário de atendimento, formas de pagamento, logotipo e fotos, sendo necessária confirmação do cadastro através de uma ligação telefônica, SMS ou carta.

## **2.13 Segurança**

Segundo Correia & Sousa (2010), o objectivo da segurança consiste em garantir um conjunto de três atributos: confidencialidade, integridade e disponibilidade.

- A confidencialidade é a ausência de divulgação não autorizada de informação.
- A integridade diz respeito à ausência de alterações não autorizadas ao sistema ou à informação em causa.
- A disponibilidade está relacionada com a prontidão do serviço fornecido pelo sistema ou da informação em causa.

Frequentemente, a estas três é ainda adicionada o atributo de autenticidade, ou seja da medida em que a informação ou serviços fornecidos são genuínos.

Segundo Correia & Sousa (2010) um conceito fundamental no domínio da segurança de software é o de vulnerabilidade. Uma vulnerabilidade é um defeito do sistema (de software ou outro) relevante para efeitos da sua segurança, ou seja, um defeito que pode ser explorado por um atacante com objectivo de subverter a política de segurança.

Os ataques por injeção são os mais comuns na Web, segundo Correia e Sousa (2010) Eles ocorrem quando dados não confiáveis são enviados para um interpretador (ex: SGBD, Serviço LADP, Sistema Operativo) como parte de um comando ou de uma query. Estes dados não confiáveis podem enganar o interpretador e permitir a execução de comandos indesejáveis ou possibilitar a leitura, criação, alteração ou remoção não autorização de dados. Segundo pesquisas o problema de injeção SQL é o mais importante e perigoso hoje em dia, por isso mereceu mais atenção de nossa parte.

## **2.14 Metodologias de desenvolvimento de software**

Segundo Pressman (2006) um software é um produto que os profissionais de software constroem e, depois mantêm ao longo do tempo. O processo de desenvolvimento de software consiste genericamente num conjunto de fases, tarefas e atividades, realizadas por intervenientes que desempenham várias funções, de modo a elaborarem diversos artefactos e que conjuntamente contribuem para a produção de um sistema de software. O conceito de metodologia implica adicionalmente a definição de aspetos que torna concretizável a noção de processo, designadamente a utilização de técnicas, notações e ferramentas (Silva e Videira, 2008).

Em 1968 diversos engenheiros propuseram metodologias para o desenvolvimento de software, os quais continham um conjunto de atividades como delimitações exata, ou seja, as metodologias não eram flexíveis, já que os desenvolvedores tinham que seguir fielmente a função de cada atividade, diversas metodologias podem ser encontradas, como por exemplo a metodologia em cascata que é a mais antiga criada em 1970, onde os desenvolvedores seguem vários passos em ordem específica, a metodologia espiral que surgiu em 1988 como forma de integrar as diversas metodologias existentes na época, eliminando os problemas e explorando os pontos positivos, essa metodologia assume que o processo de desenvolvimento ocorre em ciclos, cada ciclo contém fases de avaliação e planeamento, já na metodologia iterativa e incremental, criada em resposta das fraquezas da metodologia em cascata, o desenvolvimento é realizado por meio de incrementações e iterações sucessivas do âmbito do sistema.

Além destas metodologias temos também a RUP, ICONIX, DADI, metodologias ágeis como a Scrum, Crystal, FDD, ADP, XP, etc. Contudo, a ampla maioria das equipes ágeis bem-sucedidas tem recorrido a todos esses processos para ajustar sua forma de agilidade. Essas adaptações parecem se aglutinar em torno de uma combinação de Scrum e XP, na qual as práticas de Scrum são utilizadas para gerenciar várias equipes que utilizam XP.

A escolha da metodologia para o desenvolvimento do nosso projeto recaiu sobre a metodologia ágil XP, pois para além de ser uma metodologia ágil as suas práticas favorecem as exigências do nosso projecto, práticas essas como, equipe coesa, onde os clientes e desenvolvedores trabalham juntos de modo a colaborarem para resolver o problema, histórias de usuário que nos elucidam os requisitos, testes de aceitação que permite verificar se o sistema está se comportando conforme os clientes especificaram, desenvolvimento guiados por testes ou seja TDD que é umas das práticas necessárias para desenvolver software de qualidade, metáfora que na verdade é um sistema de nome que nos permite estabelecer uma ligação perfeita com os clientes a nível do vocabulário utilizado para o desenvolvimento do software, onde os termos que fazem parte do domínio sejam entendidos até pelos programadores. Além destas práticas a XP nos oferece também valores que são de suma importância para o desenvolvimento do nosso projecto e a boa colaboração entre os programadores e os clientes que são: comunicação, coragem, feedback, respeito e simplicidade.

A metodologia XP chega a ser também compatível com a abordagem utilizada para arquitectar o nosso projecto, nesse caso o domain drive design, onde utiliza-se a linguagem ubíquo que equiparável a metáfora presente na XP.

## **2.14.1 Visão geral da XP**

### **2.14.1.1 Equipe coesa**

Gerentes e desenvolvedores trabalham juntos para que todos conheçam os problemas uns dos outros para resolvê-los. O cliente de uma equipe XP é a pessoa ou grupo que define e prioriza recursos. Às vezes, o cliente é um grupo de analistas de negócio, especialistas em garantia de qualidade e/ou especialistas em marketing trabalhando na mesma empresa que os desenvolvedores. Às vezes, o cliente é um representante do usuário comissionado pelo grupo de usuários. Outras vezes, é quem de facto está pagando. Mas, e um projecto de XP, o cliente, seja lá como for definido, é um membro da equipe e está disponível para ela.

#### **2.14.1.2 Histórias de usuário**

Para planejar o projecto, devemos saber algo sobre os requisitos, mas não precisamos saber muito. Para propósitos de planeamento, precisamos saber sobre um requisito apenas o suficiente para estimá-lo. Talvez se acha que para estimar um requisito precisa conhecer todos os seus detalhes. Mas isso não é totalmente verdade. É necessário saber que existem detalhes e os tipos de detalhes em termos gerais, mas não precisa conhecer os pormenores.

Os detalhes específicos de um requisito provavelmente mudam com o tempo, especialmente quando o cliente começa a ver o sistema a ser montado. Nada foca melhor os requisitos do que ver o sistema ganhar vida. Portanto, capturar os detalhes específicos sobre um requisito muito antes de ele ser implementado resultará em esforço em vão e em enfoque prematuro.

Uma vez iniciada uma iteração, a empresa concorda em não mudar a definição nem a prioridade das histórias nessa iteração. Durante esse tempo, os desenvolvedores estão livres para dividir as histórias em tarefas e para desenvolver as tarefas na ordem que fizer mais sentido técnico e comercial.

#### **2.14.1.5 Desenvolvimento guiado por testes (TDD- Test-Driven Development)**

Todo código de produção é escrito para fazer teste de unidade. Primeiro escrevemos um teste de unidade que falha, pois, a funcionalidade que ele está testando não existe. Depois, escrevemos o código que faz o teste passar.

Esta iteração entre escrever casos de testes e código é muito rápida, cerca de um minuto, mais ou menos. Os casos de teste e o código evoluem juntos, com os casos de teste liderando o código por uma fracção muito pequena.

Quando é escrito um código para fazer os casos de teste passar, por definição esse código é passível de teste. Além disto, há uma forte motivação para desacoplar os módulos, a fim de que eles possam ser testados de modo independente. Assim, o projecto de um código escrito desta maneira tende a ser muito menos acoplado. Os princípios do projecto orientado a objectos desempenham um papel poderoso em ajuda-lo nesse desacoplamento.

#### **2.14.1.6 Refactoração**

O código tende a se deteriorar. A medida que adicionamos recursos e lidamos com erros, a estrutura do código degrada. Se não for controlada, essa degradação levará a uma bagunça complicada e impossível de organizar.



As equipas de XP revertem essa degradação por meio da refactorização frequente. Refactorização é a prática de fazer uma série de pequenas transformações que melhoram a estrutura do sistema, sem afectar seu comportamento. Uma transformação sozinha é insignificante, mas, juntas elas se combinam em transformações significativas por projecto e arquitectura do sistema.

A programação extrema é um conjunto de práticas simples e concretas que se combinam em um processo de desenvolvimento ágil. A XP é um bom método de uso geral para desenvolver software. Muitas equipas de projecto poderão adoptá-la como ela é. Outras poderão adaptá-la, adicionando ou modificando práticas.

### 2.14.2 Ciclo de vida do desenvolvimento de sistema

O Ciclo de Vida do Desenvolvimento de Sistemas (SDLC – Systems Development Life Cycle), conhecido também com o “ciclo de vida do software” refere-se aos estágios de concepção, projeto, criação e implementação de um SI. Um desdobramento possível para SDLC é mostrado a seguir:

- **Levantamento dos requisitos:** a actividade de levantamento de requisitos corresponde a etapa de compreensão do problema aplicada ao desenvolvimento. O principal objectivo do levantamento de requisitos é que o usuário e desenvolvedores tenham a mesma visão do problema a ser resolvido. (Bezerra, 2006)
- **Análise dos requisitos:** formalmente, o termo análise corresponde a “dividir” um sistema em seus componentes e estudar como tais componentes interagem com o objectivo de entender como esse sistema funciona. No contexto dos sistemas de software, esta é a etapa na qual os analistas realizam um estudo detalhado dos requisitos levantados na actividade anterior. (Bezerra, 2006)
- **Projecto:** nesta fase determina-se “como” o sistema funcionará para atender aos requisitos, de acordo com os recursos tecnológicos existentes (esta fase considera os aspectos físicos e dependentes de implementação) (Bezerra, 2006).
- **Implementação:** nesta fase, o sistema é codificado, ou seja, ocorre a tradução da descrição computacional da fase de projecto em código executável através do uso de uma ou mais linguagens de programação (Bezerra, 2006).

- **Testes:** nesta fase são desenvolvidos testes para verificação do sistema construído, levando-se em conta a especificação feita na fase de projecto. O principal produto dessa fase é o relatório de testes, contendo informações sobre erros detectados no software (Bezerra, 2006)
- **Implementação:** nesta fase o sistema é empacotado, distribuído e instalado no ambiente do usuário, os manuais do sistema são escritos, os arquivos carregados, os dados são importados para o sistema e os usuários são treinados para utilizar o sistema correspondente (Bezerra, 2006).

## 2.15 UML

A UML é uma notação gráfica para desenhar diagramas de conceito de software. Pode ser usada para desenhar diagramas de um domínio de problema, de um projecto de software ou de uma implementação de software já concluída. Fowler descreve esses três níveis como conceitual, especificação e implementação.

Os diagramas em nível de especificação e de implementação têm uma forte ligação com o código-fonte. Aliás, o objectivo de um diagrama em nível de especificação é ser transformado em código-fonte. Do mesmo modo, o objectivo de um diagrama em nível de implementação é descrever um código-fonte já existente. Assim, os diagramas nesses níveis devem seguir certas regras e certa semântica. Tais diagramas têm pouca ambiguidade e bastante formalidade.

Por outro lado, os diagramas no nível conceitual não são fortemente relacionados com o código-fonte. Em vez disto, se relacionam com a linguagem humana. Eles são uma forma abreviada, utilizada para descrever conceitos e abstrações que existem no domínio do problema humano. Como não seguem regras semânticas rígidas, seu significado pode ser ambíguo e estar sujeito à interpretação.

A UML tem três tipos de diagramas principais. Os diagramas estáticos descrevem a estrutura lógica imutável de elementos de software, representando classes, objectos e estruturas de dados e os relacionamentos existente entre eles. Os diagramas dinâmicos mostram como as entidades de software mudam durante a execução, representando o fluxo de execução ou a maneira como as entidades mudam de estado. Os diagramas físicos mostram a estrutura física imutável das entidades de software, representando entidades físicas, como arquivos de código-fonte, bibliotecas, arquivos binários, arquivos de dados e seus semelhantes, e os relacionamentos existentes entre elas.

### **2.15.1 Ferramentas CASE**

Mcconnell (2004), sublinha que as ferramentas case são ferramentas de engenharia de software auxiliada por computadores, que consistem principalmente em ferramentas gráficas que permitem expressar um projecto de Software em notações gráficas.

Estas ferramentas modernas de projecto de Software, possuem recursos gráficos que um pacote de Software gráfico normal não pode oferecer. Usando estas ferramentas CASE é possível reduzir o tempo de desenvolvimento, pois elas auxiliam actividades de engenharia de Software, desde análise de requisitos e modelagem até programação e testes.

## **3. CAPITULO 2 - METODOLOGIA**

### **3.1 Tipo de pesquisa**

A investigação em geral caracteriza-se por utilizar os conceitos, as teorias, a linguagem, as técnicas e os instrumentos com finalidade de dar resposta aos problemas e interrogações que se levantam nos diversos âmbitos do trabalho (Reis, 2010).

Neste projecto aplicou-se o tipo de pesquisa qualitativa, visto que a pesquisa qualitativa procura descrever e analisar a cultura e comportamento dos seres humanos e seus grupos a partir do ponto de vista daqueles que estão sendo estudados, a pesquisa qualitativa inclui estudo de caso de comunidades e instituições, debates e questionários dos grupos focais, onde os sentimentos e percepções são considerados importantes (Orodho e Kombo, 2002).

A nossa pesquisa foi delineada por entrevistas e questionários aos membros da CNE que se dispuseram em responder, como também fizemos algumas questões aos internautas, especificamente nas redes sociais, tivemos como base a pesquisa teórica, consultando alguns documentos previamente existentes que caracterizam o actual sistema de votação, muitos deles disponibilizados pela CNE, concomitantemente fizemos estudo de trabalhos relacionados ao nosso tema.

### **3.2 Campo de estudo**

No que visa o campo de estudo, este projecto foi delimitado na CNE. A Comissão Nacional Eleitoral é um órgão independente que organiza, executa, coordena e conduz os processos eleitorais.

A estrutura orgânica e funcional da Comissão Nacional Eleitoral bem como as competências específicas dos seus órgãos é fixada por esta e aprovadas por lei.

A CNE é uma entidade administrativa não integrada na administração directa e indirecta do Estado, que goza de independência orgânica e funcional.

É uma entidade orçamental própria, dotada de autonomia administrativa, financeira e patrimonial.

A Comissão Nacional Eleitoral exerce colegialmente as suas competências e atribuições e não pode decidir em termos contrários às disposições referentes ao processo eleitoral, constantes da lei eleitoral em vigor.

A CNE tem a sua sede na Rua Amílcar Cabral nº 102/104, em Luanda.

### **3.3 Estudo de viabilidade**

Levando em conta os problemas encontrados e dada a sua relevância, achamos de suma importância a resolução destes problemas com o sistema de votos electrónicos, apesar de existirem desafios para o uso do sistema para a maior parte da população, mas podemos resolver isso com a formação do pessoal, mas o uso massivo das tecnologias tem se tornado cada vez mais abrangente na nossa sociedade que dentro de algum tempo as pessoas ficarão mais adaptadas com o sistema de votos electrónicos.

É viável desenvolver este projecto, visto que temos facilidade na obtenção dos dados, as ferramentas a serem utilizadas para o desenvolvimento e colecta dos dados são gratuitas, os custos envolvidos para aquisição de materiais de pesquisa como a internet, podem ser facilmente cobridos.

### **3.4 Levantamento de requisitos**

Nesta etapa procedemos a recolha de informações relativas ao projecto. Estas informações foram adquiridas por meio de entrevistas, questionários, pesquisa teórica sobre os materiais existentes, disponibilizados pela CNE, com estes meios conseguimos então fazer o levantamento dos requisitos.

### 3.4.1 Análise de requisitos

A análise de requisitos é a primeira actividade técnica no desenvolvimento do software, e pode ser entendida como responsável por definir os serviços que um sistema deve realizar.

De uma forma simples também podemos dizer que requisitos são necessidades dos futuros utilizadores do sistema a ser desenvolvido. Os requisitos agrupam-se em dois grandes grupos que são: requisitos funcionais e requisitos não funcionais.

#### 3.4.1.1 Requisitos funcionais

Um requisito é tratado como funcional quando descreve um serviço ou função que o sistema deve realizar.

Para o sistema extraiu-se os seguintes requisitos funcionais:

ID	Descrição
RF1	O sistema deverá permitir ao administrador cadastrar, consultar, alterar e inactivar usuários.
RF2	O sistema deverá permitir ao administrador cadastrar, consultar, alterar e inactivar comissão.
RF3	O sistema deverá permitir cadastrar, consultar, alterar e inactivar eleitor.
RF4	O sistema deverá permitir ao administrador cadastrar, consultar, alterar e inactivar partidos.
RF5	O sistema deverá permitir o administrador cadastrar ano eleitoral, afim de dar início ao processo de eleição.
RF6	O sistema deverá permitir incluir o eleitor ao caderno eleitoral.
RF7	O sistema deverá permitir ao eleitor visualizar o boletim de voto.
RF8	O sistema deverá permitir ao eleitor votar.
RF9	O sistema deverá permitir ao administrador fazer a contagem dos votos.
RF10	O sistema deverá permitir ao administrador publicar os resultados dos votos.

RF11	O sistema deverá permitir ao administrador emitir relatório de abstencionismo.
RF12	O sistema deverá permitir ao administrador e o recenseador enviar notificação aos eleitores.
RF13	O sistema deve permitir ao administrador gerar os boletins de voto.
RF14	O sistema deverá permitir ao recenseador e o administrador consultarem o caderno eleitoral.
RF15	O sistema deverá permitir ao administrador cadastrar, consultar, alterar e inactivar grupos.
RF16	O sistema deverá permitir ao administrador enviar mensagens por sms ao eleitor.
RF17	O sistema deverá permitir ao eleitor consultar os resultados.
RF18	O sistema deverá permitir consultar as mensagens disponíveis.
RF19	O sistema deverá permitir efectuar carregamento de mensagens.
RF20	O sistema deverá permitir gerir recenseadores.
RF21	O sistema deverá permitir ao eleitor pesquisar o local de voto

**Tabela 1:**Requisitos funcionais

### 3.4.1.2 Requisitos não funcionais

Requisitos não funcionais são requisitos relacionados ao uso da aplicação em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade e tecnologia envolvidas.

Para o sistema extraiu-se os seguintes requisitos não funcionais:

<b>Categoria</b>	<b>Descrição</b>
Segurança	<p>O sistema deverá possuir um mecanismo de segurança por login.</p> <p>O sistema deverá possuir um algoritmo de criptografia SHA 512.</p> <p>O sistema deverá possuir níveis de acesso (autorização).</p> <p>O sistema deverá possuir autenticação de dois factores.</p>

Plataforma	O sistema deverá funcionar via web, acedido através de um browser com suporte a tecnologia HTML5 e Javascript.
Interface	O sistema deverá possuir uma interface fácil de manipular.
Tecnologias	O sistema deve utilizar as seguintes tecnologias: ASP.NET MVC, Angular JS, Entity Framework, ADO.NET, Ninject(Ioc), AutoMapper, JQuery, Bootstrap, ReSharper, ASP.NET Web API, JSON e Google Maps API.
Integridade	O eleitor só poderá votar uma vez por eleição. O voto do eleitor deve ser singular e irrevogável.
Disponibilidade	O sistema deverá possuir backup, diário, semanal, mensal e anual. O sistema deverá possuir um mecanismo de redundância (alta disponibilidade).

**Tabela 2:** Requisitos não funcionais

### 2.4.1.3 Regras de negócio

As regras de negócio definem como uma empresa funciona, normalmente são políticas, validações, restrições ou condições específicas. Tais regras podem determinar um comportamento do sistema. No quadro abaixo estão descritas as regras de negócio que foram captadas.

Código	Descrição
RN-01	O processo eleitoral deve começar com o cadastro do ano eleitoral.
RN-02	Para efectuar o cadastro de um ano eleitoral, deve-se dar por terminado o ano eleitoral anterior. Onde o intervalo será de acordo o mandato.
RN-03	Para efectuar o cadastro de comissão, os campos obrigatórios devem ser preenchidos! Campos obrigatórios: rua e descrição.
RN-04	Para efectuar o cadastro de grupo, os campos obrigatórios devem ser preenchidos! Campo obrigatório: descrição.

RN-05	Para efectuar o cadastro de funcionário, os campos obrigatórios devem ser preenchidos! Campos obrigatório.
RN-06	As entidades, província, município, bairro, rua devem ser preenchidas na implementação do sistema.
RN-07	Para efectuar o cadastro de tipo de eleição, os campos obrigatórios devem ser preenchidos! Campo obrigatório: descrição.
RN-08	Para efectuar o cadastro de partido, os campos obrigatórios devem ser preenchidos! Campos obrigatórios: descrição, proposta, data de existência.
RN-09	Para efectuar o cadastro de eleitor, os campos obrigatórios devem ser preenchidos! Campos obrigatórios: nregistro,nome,sexo,data de nascimento, local de nascimento, foto. Só perá ser cadastrado como eleitor cidadãos que tenham idade superior a 18 anos, com todos critérios cumpridos o eleitor deve ser cadastrado no caderno eleitoral.
RN-10	Para efectuar o cadastro no caderno eleitoral, os campos obrigatórios devem ser preenchidos! Campos obrigatórios: rua, referência, estado do eleitor. Sendo que um eleitor só poderá ser incluso ao caderno eleitoral uma vez por ano eleitoral.
RN-11	Para efectuar o exercer o sufrágio, os campos obrigatórios devem ser preenchidos! Campo obrigatório: partido. O eleitor só poderá votar uma vez por ano eleitoral.
RN-12	Depois do eleitor exercer o sufrágio, o seu estado deve alterar de modo a identificar que ele já exerceu o sufrágio, de modo a manter a singularidade e a integridade na base de dados.
RN-14	Para inactivar qualquer registo, deve-se fazer uma pesquisa antes de inactiva-lo.
RN-15	O sistema deve possuir um log de todas operações feitas.



RN-16	O administrador deve gerar os boletins de voto, para que os eleitores possam votar.
RN-17	O sistema deve garantir somente pessoas autorizadas acederm o sistema e o seu conteúdo.

**Tabela 3:**Regras de negócio

### **3.5 Análise do sistema**

A qualidade do produto de software aumenta cada vez mais, na medida em que se aplica à análise neste desenvolvimento, pois quando os atributos de uso do sistema computacional são bem claros, o resultado no acerto e na eficácia do produto final é eminente, provendo um bom grau de satisfação do usuário final.

Na etapa da análise o padrão de diagramas, Unified Modeling Language (UML) possibilita mostrar de maneira clara e consistente a modelagem do sistema, através de seus principais diagramas, como o diagrama de casos de uso, o diagrama de classes e o diagrama de seqüência.

#### **3.5.1 Diagramas do sistema**

##### **3.5.2 Diagrama de caso de uso**

Um caso de uso é a especificação de interações entre sistema e os agentes externos que utilizam este sistema. Neste tipo de diagramas temos os actores que representa um agente externo no nosso sistema, como por exemplo o “eleitor”, consome o caso de uso “visualizar o boletim de voto” que possui um relacionamento por include que obriga-o a fazer primeiro o login, concomitantemente possui o extend com o caso de uso votar, que indica algo opcional.

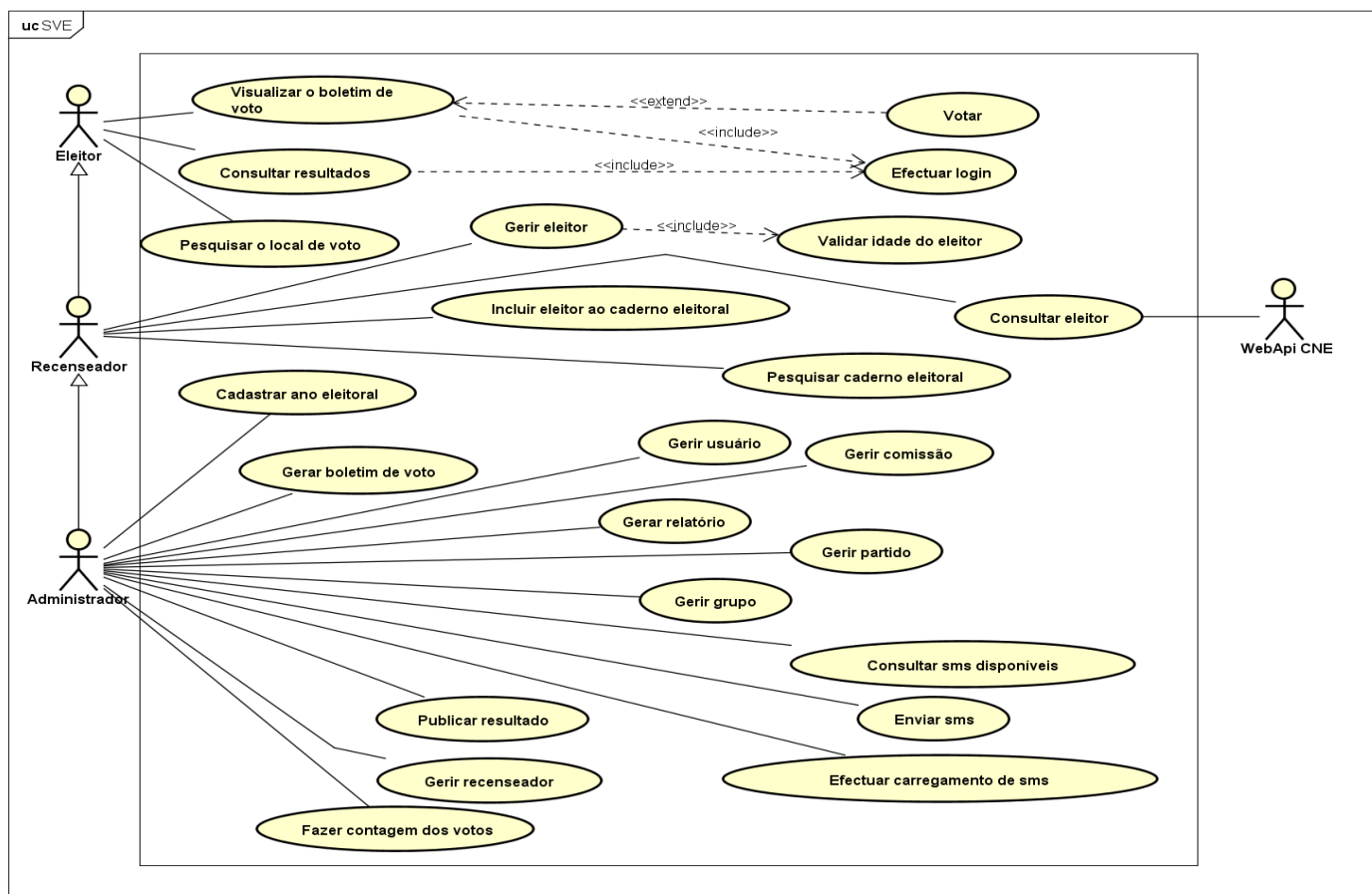


Figura 3:Diagrama de caso de uso

Código	Nome	Módulo
UC01	Efectuar login	Segurança
UC02	Gerir comissão	Recenseamento
UC03	Gerir usuário	Recenseamento
UC04	Gerir grupo	Recenseamento
UC05	Gerir partido	Recenseamento
UC06	Cadastrar ano eleitoral	Votação
UC07	Gerir eleitor	Recenseamento

UC08	Visualizar boletim de voto	Votação
UC09	Votar	Votação
UC10	Consultar resultados	Contabilização
UC11	Incluir eleitor ao caderno eleitoral	Recenseamento
UC12	Pesquisar caderno eleitoral	Recenseamento
UC13	Fazer contagem dos votos	Contabilização
UC14	Publicar resultado	Contabilização
UC15	Emitir relatório	Todos
UC16	Efectuar carregamento de sms	Comunicação
UC17	Enviar sms	Comunicação
UC18	Consultar sms disponíveis	Comunicação
UC19	Gerar boletins de voto	Votação
UC20	Validar idade do eleitor	Recenseamento
UC21	Gerir recenseador	Recenseamento
UC22	Pesquisar o local de voto	Votação

**Tabela 4:**Casos de uso

### 3.5.2.1 Descrição dos principais casos de uso

Esta tabela descreve o caso de uso permite cadastrar um novo ano eleitoral, para dar início a processo eleitoral.

Cadastrar ano eleitoral	
Descrição:	Este caso de uso permite cadastrar um ano eleitoral.
Pré-condição:	1. Entrar no sistema.

	2. Ter permissão para realizar o caso de uso
Pós-condição	O sistema cadastra o novo ano eleitoral.
Actor:	Administrador
<b>Fluxo principal</b>	<ol style="list-style-type: none"> <li>1. O usuário clica no item cadastrar ano eleitoral.</li> <li>2. O sistema apresenta a página de cadastro do ano eleitoral.</li> <li>3. O usuário insere a observação de forma opcional e clica no botão “cadastrar ano eleitoral”.</li> <li>4. O sistema confirma o cadastro.</li> <li>5. O sistema finaliza o caso de uso.</li> </ol>
<b>Fluxo alternativo</b>	O sistema informa ao usuário o erro ocorrido, apresenta a mesma página e o usuário faz a correção.

**Tabela 5:**Detalhamento do caso de uso cadastrar ano eleitoral

Esta tabela descreve o caso de uso que permite incluir o eleitor ao caderno eleitoral depois do seu recenseamento, visto que o caderno eleitoral representa a lista dos cidadãos aptos para exercerem o sufrágio.

<b>Incluir eleitor ao caderno eleitoral</b>	
Descrição:	Este caso de uso permite incluir o eleitor ao caderno eleitoral, de modo a habilitá-lo para exercer o sufrágio.
Pré-condição:	<ol style="list-style-type: none"> <li>1. Entrar no sistema.</li> <li>2. Ter permissão para realizar o caso de uso</li> </ol>
Pós-condição	O sistema inclui o eleitor ao caderno eleitoral.
Actores:	Recenseador e Administrador
<b>Fluxo principal</b>	<ol style="list-style-type: none"> <li>1. O usuário clica no item caderno eleitoral e selecciona a opção de incluir eleitor ao caderno eleitoral.</li> <li>2. O sistema apresenta a página de inclusão do eleitor ao caderno eleitoral.</li> <li>3. O usuário insere o nº do cartão do eleitor, ou o nº do BI, ou o e-mail ou o telefone e clica no botão “pesquisar”.</li> <li>4. O sistema faz a validação e retorna os dados do eleitor.</li> </ol>

	<ol style="list-style-type: none"> <li>5. O usuário insere os dados referentes ao local de voto do eleitor.</li> <li>6. O usuário clica no botão “incluir eleitor”.</li> <li>7. O sistema valida os dados e confirma a inclusão do eleitor ao caderno eleitoral.</li> <li>8. O sistema finaliza o caso de uso.</li> </ol>
<b>Fluxo alternativo</b>	O sistema informa ao usuário o erro ocorrido, apresenta a mesma página para o usuário fazer a correção.

**Tabela 6:**Detalhamento do caso de uso incluir eleitor ao caderno eleitoral

Esta tabela descreve o caso de uso que permite ao eleitor exercer o sufrágio.

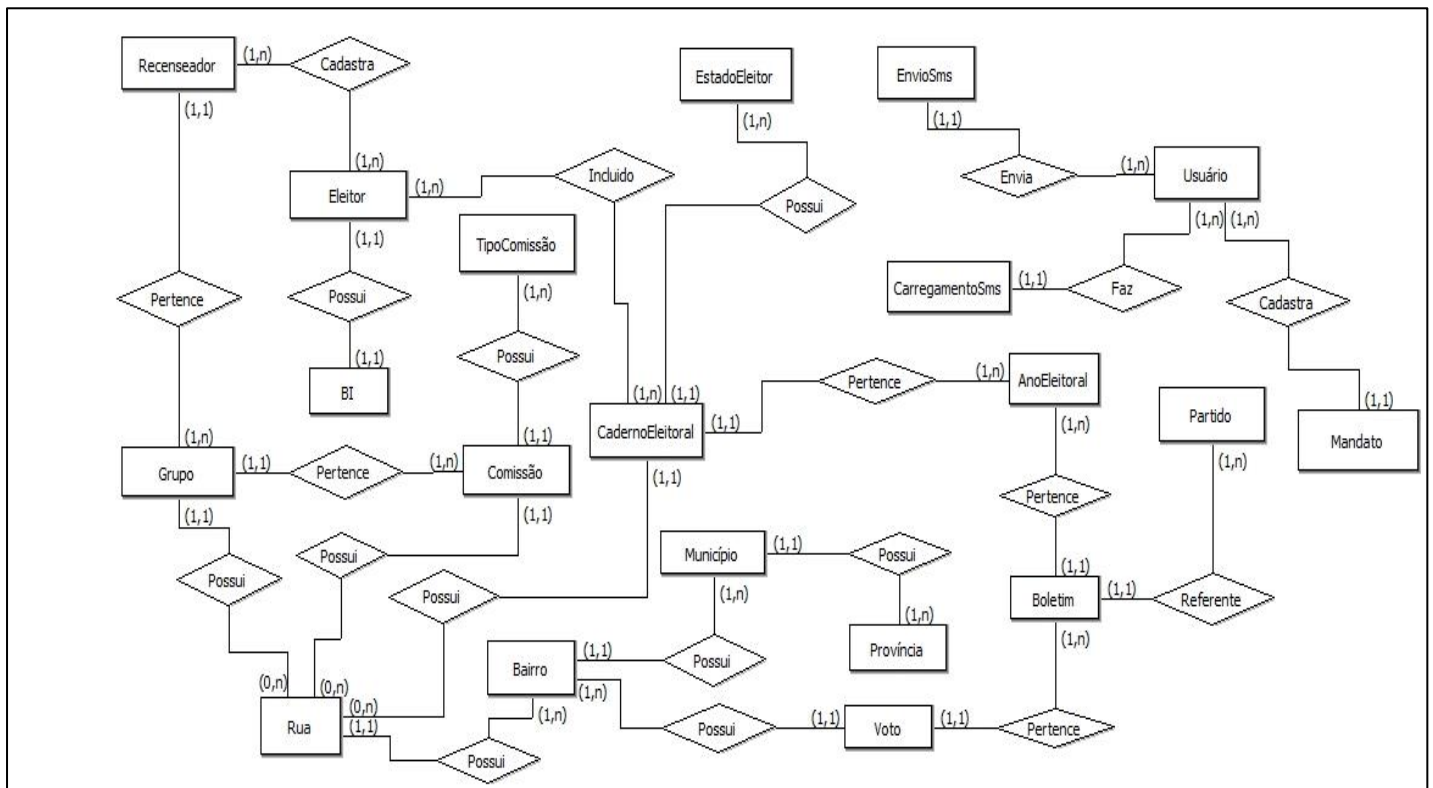
<b>Votar</b>	
Descrição:	Este caso de uso permite exercer o sufrágio.
Pré-condição:	<ol style="list-style-type: none"> <li>1. Entrar no sistema.</li> <li>2. Ter permissão para realizar o caso de uso</li> </ol>
Pós-condição	O sistema regista o voto do eleitor na base de dados.
Actor:	Eleitor
<b>Fluxo principal</b>	<ol style="list-style-type: none"> <li>1. O usuário executa o caso de uso “visualizar boletim de voto”.</li> <li>2. O sistema apresenta as opções de votos no boletim, neste caso os partidos.</li> <li>3. O usuário escolhe a sua opção de voto, clicando em votar.</li> <li>4. O sistema solicita a confirmação do usuário.</li> <li>5. O usuário confirma a sua opção, clicando em confirmar.</li> <li>6. O sistema regista o voto do eleitor na base de dados e termina o caso de uso.</li> </ol>
<b>Fluxo alternativo</b>	O sistema informa ao usuário o erro ocorrido e envia notificação ao administrador.

**Tabela 7:**Detalhamento do caso de uso votar

### 3.5.3 Diagrama de entidade-relacionamento

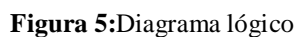
Denominamos entidade estes elementos. Atribuímos a cada entidade definidos atributos pertinentes ao sistema. Desta forma, podemos representar conceitualmente como entidades

aqueles elementos no qual gostaríamos de armazenar dados – que por sua vez. Através do relacionamento representaremos o tipo de relação existente entre as entidades.



**Figura 4:**Diagrama de entidade-relacionamento

Compreende uma descrição das estruturas que serão armazenadas no banco e que resulta em uma representação gráfica dos dados de uma maneira lógica, inclusive nomeando os elementos e ações que exercem uns sobre os outros.



### 3.5.5 Diagrama de classe

Os diagramas de classe nos permitem denotar o conteúdo estático e os relacionamentos de classes. Em um diagrama de classe também podemos mostrar se uma classe herda de outra ou se contém uma referência para outra. Em resumo, podemos retratar todas as dependências de código-fonte entre as classes.

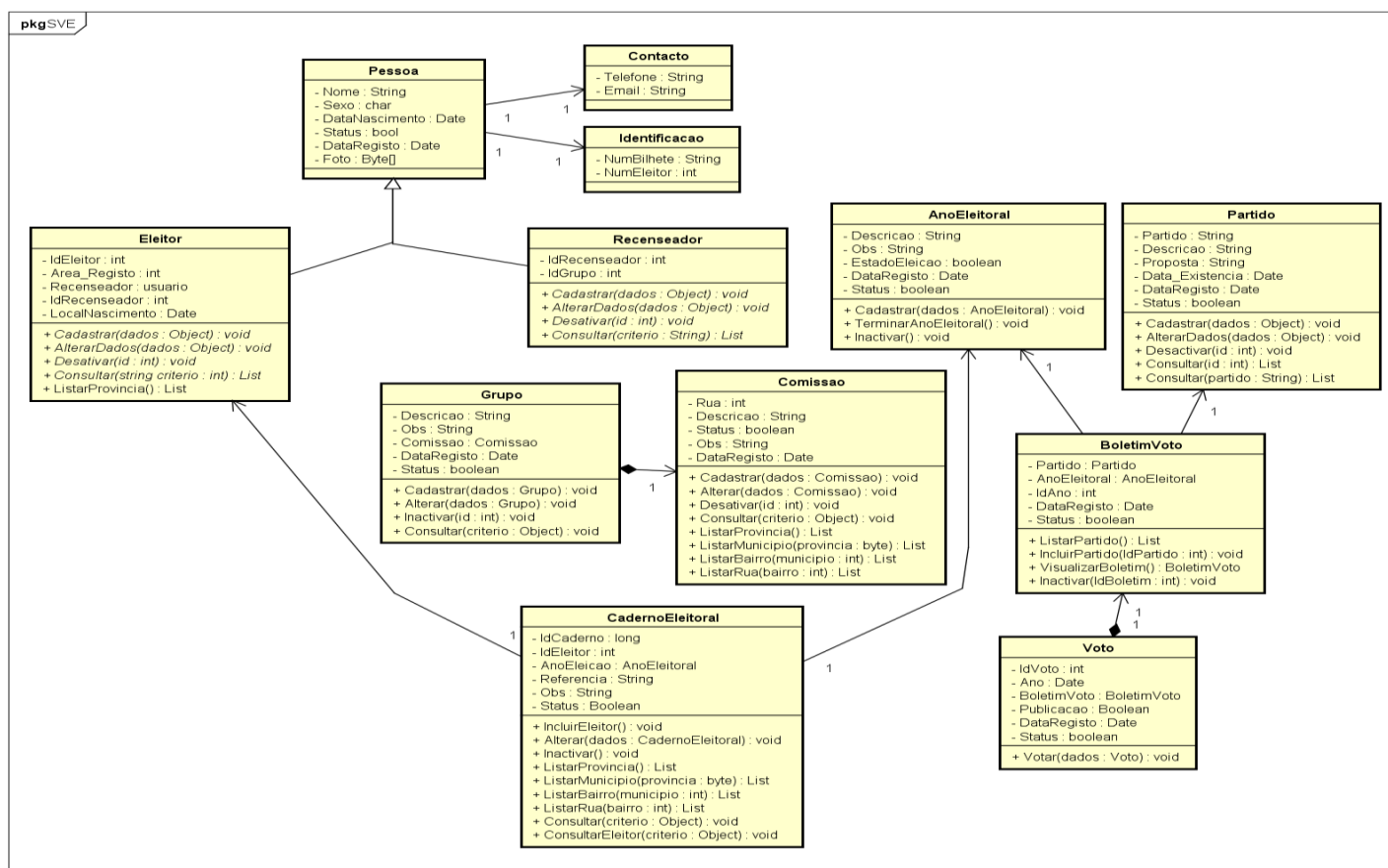


Figura 6:Diagrama de classe

### 3.5.6 Diagrama de sequência

Os diagramas de sequência permitem representar o comportamento dinâmico do sistema, nomeadamente as interações entre objectos. Os diagramas de sequência são uma das formas preferenciais de especificar os casos de uso. O diagrama consiste essencialmente na representação de sequência de chamada de métodos (troca de mensagens) entre os objectos das classes que suportam a aplicação.



Desta forma temos abaixo ilustrado alguns digramas de sequência que representam processos relevantes no nosso sistema.

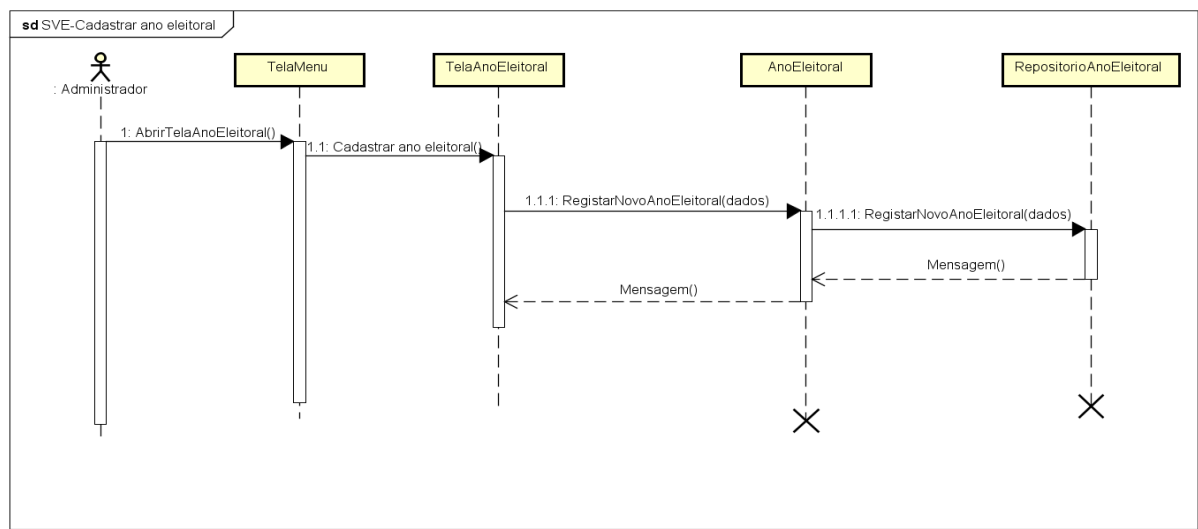


Figura 8:Diagrama de sequência cadastrar ano eleitoral

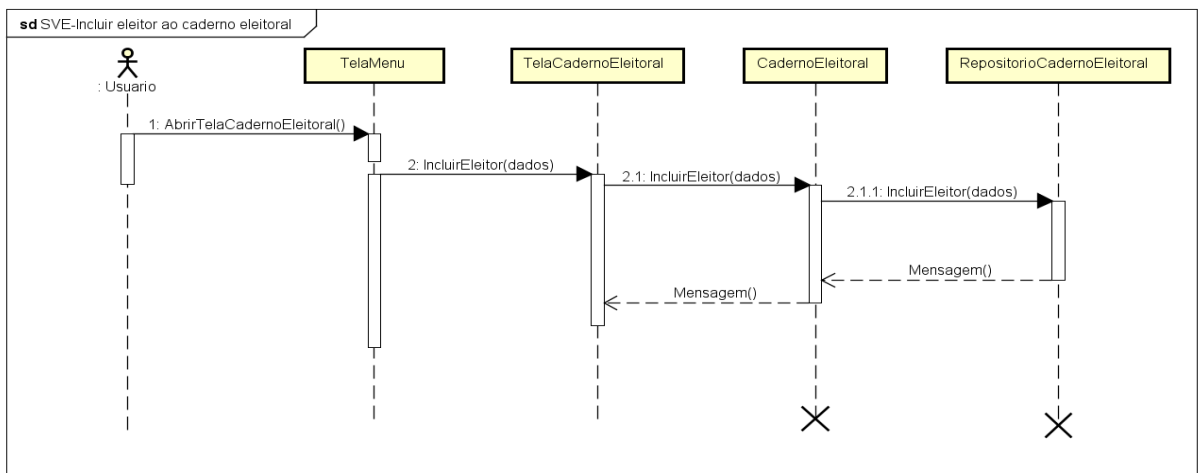
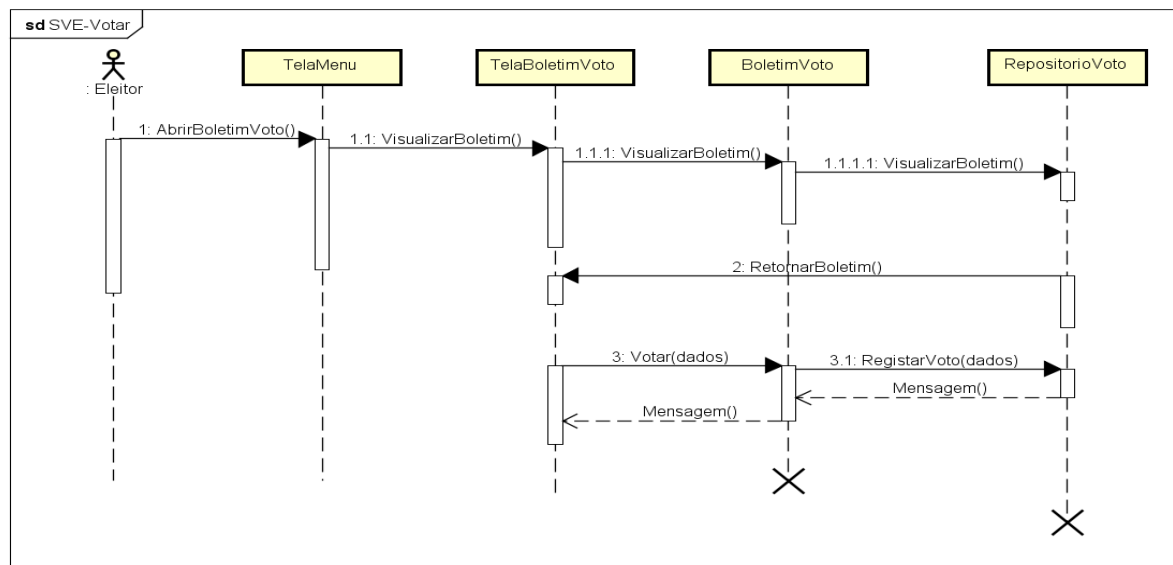


Figura 7:Diagrama de sequência incluir eleitor ao caderno eleitoral

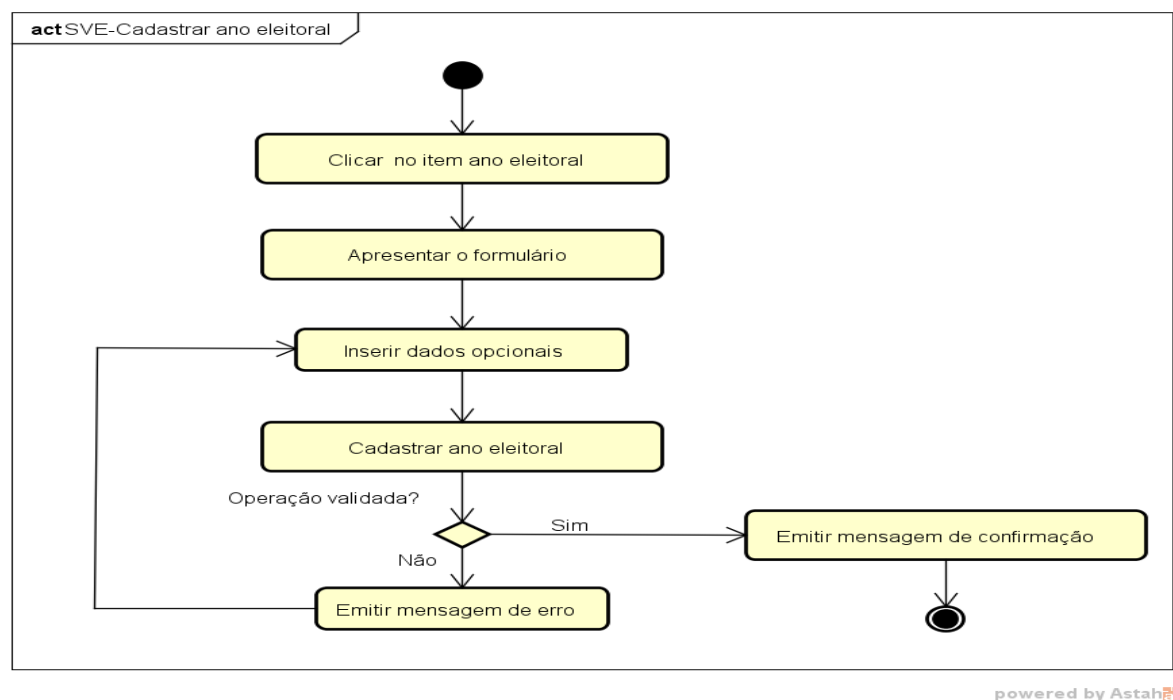


**Figura 9:**Diagrama de sequência votar

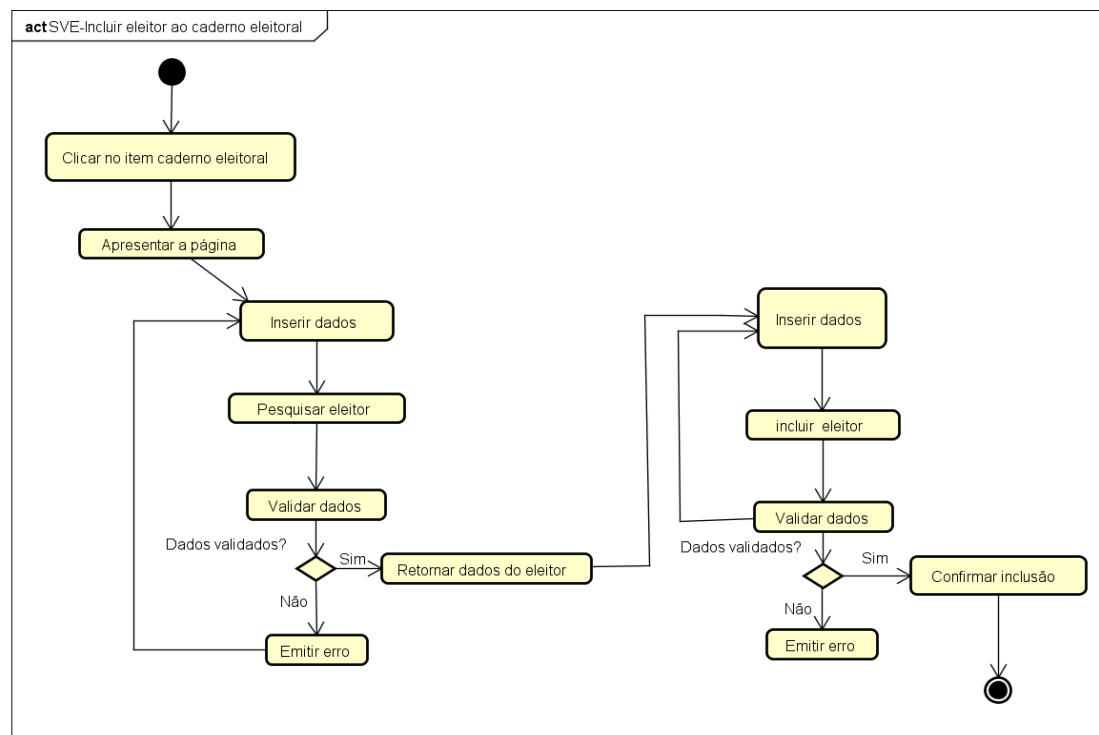
### 3.5.7 Diagrama de actividade

O diagrama de actividades permite descrever um cenário de realização de um caso de uso através do encadeamento de actividades que são realizadas pelos diversos objectos. Este diagrama é habitualmente utilizado para descrever processos organização.

Abaixo temos alguns diagramas representados:

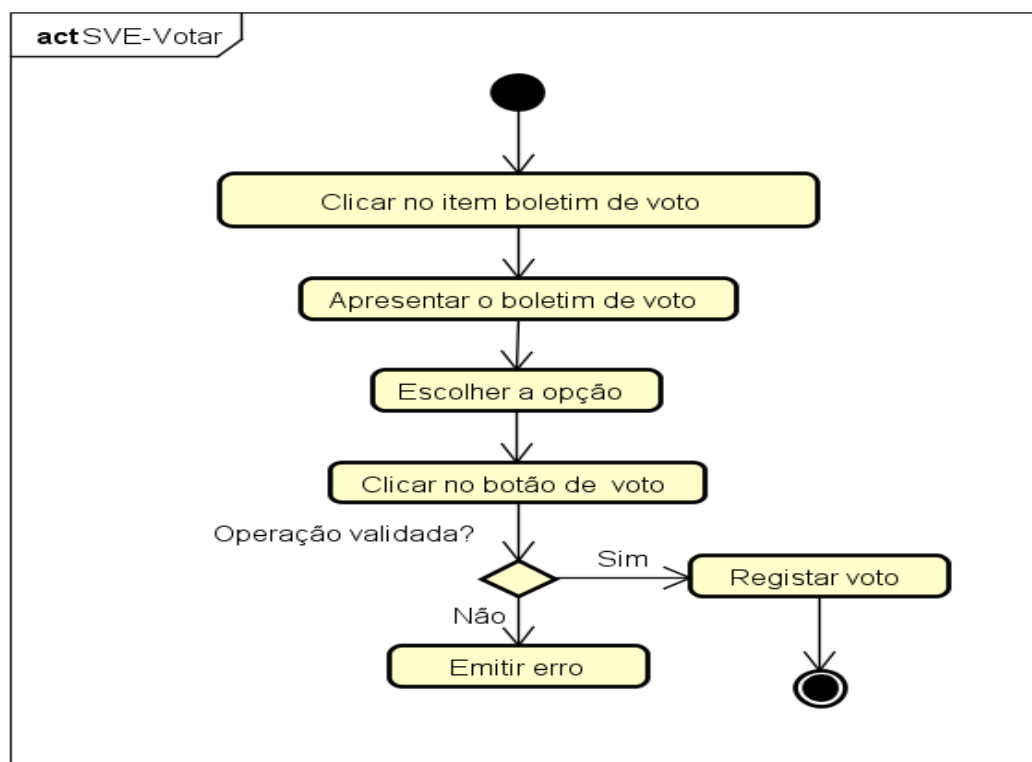


**Figura 10:**Diagrama de actividade cadastrar ano eleitoral



powered by Astah

**Figura 11:**Diagrama de actividade incluir eleitor ao caderno eleitoral



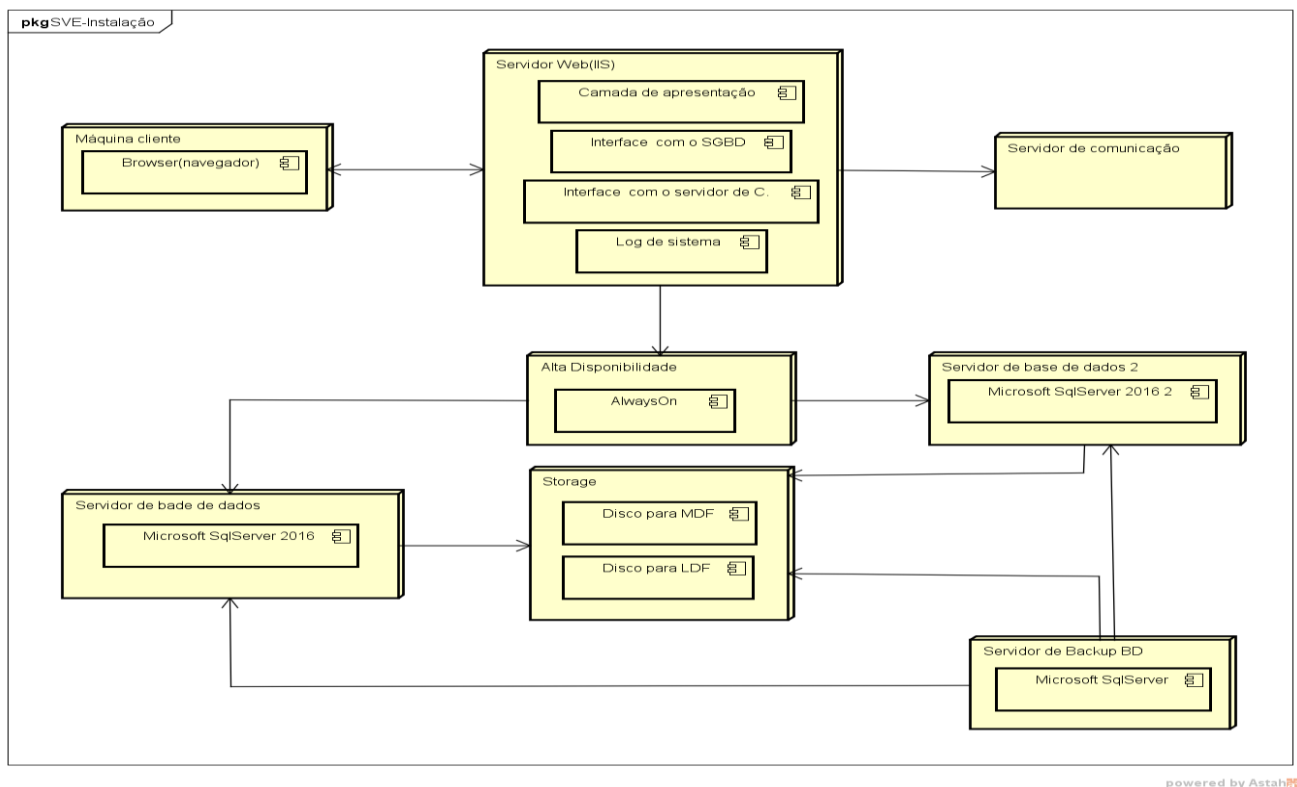
powered by Astah

**Figura 12:**Diagrama de actividade votar

### 3.5.8 Diagrama de instalação

Diagrama de instalação ou de implantação descreve os componentes de hardware e software e sua interação com outros elementos de suporte ao processamento.

O diagrama abaixo representa o layout de instalação do nosso sistema.



**Figura 13:**Diagrama de instalação

Os componentes listados no diagrama relacionam-se da seguinte forma:

Do ponto mais baixo temos uma estrutura de alta disponibilidade com o AlwaysOn, pertencente ao SqlServer, de modo a manter no ar o servidor de base dados caso um deles tenha algum problema, por isso temos um esquema que demonstra a redundância de servidor de base de dados. Não obstante a alta disponibilidade, temos também uma implementação simplista do RAID ou seja conjunto redundante de disco econômicos, separando então os arquivos MDF para um disco diferente de onde se encontra o arquivo LDF que é na verdade o arquivo de log, portanto se o disco dos arquivos MDF danificar, podemos aceder o disco do LDF para recuperar a base de dados, se ambos estivessem no mesmo disco, em caso de danificar, perderíamos toda informação, mas ainda assim os dois discos estão sujeitos a danificarem por algum motivo, razão pela qual utilizamos um servidor para o backup da base de dados. Separamos os arquivos

de instalação do SqlServer dos arquivos de dados, por isso temos um storage, visto que essa é uma das melhores práticas para sistemas robustos.

Temos um servidor de comunicação, que vai servir para enviar sms ou e-mail, de modo a não sobrecarregar o pipeline do nosso servidor web que roda o IIS, o nosso servidor IIS tem então toda estrutura lógica do nosso projecto, do ponto de vista de programação, dando mais enfoque na camada de apresentação que permite a interação com o usuário. O nosso servidor web tem uma ligação com o nosso servidor de base de dados, isso para persistir e consultar os dados, quando surge a necessidade de enviar uma mensagem ele faz uma requisição ao servidor de comunicação para processar o envio da mensagem, por isso também tem uma ligação com o servidor de comunicação.

Para melhorar a monitoração do nosso sistema, implementamos um log de sistema que nesse caso é um trace que no decorrer do funcionamento do servidor podemos consultar online.

A máquina do cliente precisa rodar um browser com suporte a html 5 e que executa códigos javascript.

### **3.6 Ferramentas de desenvolvimento**

- Astah community
- Microsoft Sql Server 2016
- Microsoft Visual Studio 2015
- Adobe PhotoShop
- Google Chrome
- brModelo
- Microsoft Office 2016
- MeiGen 1.0
- SqlServer Profiler

### **3.7 Arquitectura lógica do sistema**

O sistema foi desenvolvido numa arquitectura de N camadas, aplicando os conceitos de DDD(Domain Drive Design) que nos levou a implementar as seguintes camadas:

- Apresentação.
- Serviços.
- Aplicação.
- Domínio.

- Infra-estrutura.
- Testes

### **3.7.1 Camada de apresentação**

A arquitectura lógica apresentada possui separação de responsabilidades, onde a camada de usuário é responsável pela comunicação (interface) com os utilizadores da aplicação. Qualquer mudança nesta camada não influencia as outras.

### **3.7.2 Camada de serviços**

A camada de serviços é responsável por disponibilizar serviços, normalmente baseados no REST, a fim de serem consumidos por uma outra aplicação, esta camada foi incluída no nosso projecto a fim de abrir a possibilidade de fazer uma integração com uma aplicação para dispositivos móveis, essa integração pode ser feita através de uma WebApi ou mesmo um WCF.

### **3.7.3 Camada de aplicação**

A camada de aplicação é uma camada fina, que coordena a actividade da aplicação. Ela não controla o estado do objecto de negócio, mas pode controlar o progresso do estado de uma aplicação. Basicamente a camada de aplicação é responsável por disponibilizar as operações do domínio para a camada de apresentação ou mesmo para a camada de serviços.

### **3.7.4 Camada de domínio**

Esta camada contém informações do domínio. Ela é o coração do negócio do software. O estado do objecto de negócio são mantidos nesta camada. A persistência dos objectos de negócio, e possivelmente, seu estado, é delegado a camada de infra-estrutura. A camada de negócio(domínio) comporta toda regra de negócio do nosso sistema, e não só, ela possui também as entidades, que neste caso são as classes que mapeiam as nossas tabelas da base de dados, temos também as nossas interfaces de repositório que são implementadas no repositório da camada de infra-estrutura, e os objectos de valores que descrevem objectos imutáveis e que não têm uma identidade no mundo real.

A camada de domínio possui os serviços, que são chamados de serviços de domínios, estes serviços representam a lógica do nosso negócio, são eles que na arquitectura bolovo(3 camadas) (2007) são chamados de business logic layer (camada de lógica de negócio). Mas com abordagem DDD as coisas acabamos por ter uma diferença na nomenclatura.

### **3.7.5 Camada de infra-estrutura**

Esta camada actua como biblioteca e apoio as outras camadas, implementa persistência dos dados para os objectos de negócio, neste caso os repositórios, como também contém bibliotecas de apoio, como por exemplo a biblioteca para envio de sms, todas estas bibliotecas devem fazer parte da camada de infra-estrutura, e não do domínio, desta forma tornamos o nosso domínio agnóstico, conforme manda o DDD.

#### **3.7.6.1 Camada de testes**

Esta camada actua como facilitadora da implementação do TDD, nesta camada criamos os nossos mock objects para permitir-nos gerar os métodos nos testes e depois implementá-los e por fim testar os mesmos, ou seja o seu comportamento, desta forma temos um ganho na qualidade, sendo que testar o método antes de implementar nas outras camadas é uma certeza de que o método funciona correctamente. Neste projecto adaptamos o TDD a realidade do nosso projecto, na qual não precisamos testar todos os métodos.

### **3.8 Desenvolvimento de base de dados**

O desenvolvimento e implementação de uma base de dados é uma parte essencial do processo de engenharia de software. As bases de dados são concebidas (estrutura conceitual) e construídas (modelo físico) tendo em conta a escalabilidade. As bases de dados relacionais são especificadas e implementadas respeitando as regras de normalização no sentido de minimizar a redundância dos dados.

Os dados do negócio e do domínio da aplicação são suportados por base de dados, razão pela qual decidimos criar a base de dados para o sistema.

#### **3.8.1 Criação da base de dados**

Para criação da base de dados do projecto utilizamos o Microsoft SqlServer Developer 2016, que é uma solução de persistência de dados da Microsoft. Escolhemos o SqlServer porque é uma das soluções que atende as características do nosso projecto, levando em conta que o nosso projecto requer muita exigência a nível de segurança, então o Sql Server na versão de 2016 implementou melhorias de segurança, como o always encrypted, melhorias em soluções de alta disponibilidade que anteriormente era feito com o log shipping e agora temos o always on, melhorias em replicação e outras tantas melhorias.

### **3.9 Tecnologias envolvidas no projecto**

#### **3.9.1 Base de dados**

##### **3.9.1.1 Procedimentos armazenados**

Procedimentos armazenados (stored procedure) é um conjunto de comandos armazenados no servidor, utilizamos procedimentos armazenados nas operações mais complexas do nosso sistema, de modo a reduzir o tráfego na rede e ganhar mais performance, as outras operações mais simples deixamos para a ferramenta ORM utilizada no nosso projecto, sendo que o ADO.NET tradicional é mais rápido que a ferramenta ORM.

Apesar de beneficiarmo-nos do T-SQL, fizemos com que as nossas queries seguissem o padrão ANSI o máximo possível, de modo a facilitar a sua implementação num outro SGBD.

##### **3.9.1.2 Sequências**

Sequências (sequence) são utilizados como substitutos ao identity, para campos de auto numeração. As sequências são mais flexíveis que o identity, sendo que o identity mesmo em caso de erro a contagem não para, já com as sequências isto não acontece, não há perda de valor desnecessariamente, com a sequência podemos definir onde começa a nossa numeração e onde termina, sendo que é possível fazermos a manutenção na sequência.

Utilizamos a sequência na nossas tabelas de eleitor, caderno eleitoral e voto , visto que a possibilidade de termos um número alto de registos é positiva, então para não perdermos valores desnecessariamente, utilizamos a sequência em vez do identity, mas em outras tabelas com um número de registos reduzidos, utilizamos o identity.

##### **3.9.1.3 Column Store Indexes**

O Column Store Index é a novo tipo de índice implementado no Sql Server 2016, esse novo tipo de índice faz leitura dos dados por coluna, diferente do antigo Row Store Index que faz leitura por linha, desta forma a leitura dos dados com o Column Store Index acaba por ser mais rápida, sendo que cada página de leitura equivale a 4K. Essas diferenças acabam por ser mais significativas quando o temos muitos registos numa determinada tabela.

Aplicamos esse novo tipo de índice para as tabelas como eleitor, caderno eleitoral e voto, sendo que estas tabelas são as que possuirão mais registos no decorrer do processo eleitoral.

##### **3.9.1.4 Paralelismo**

Quando temos o SqlServer instalado em um servidor que tem mais de uma CPU, as queries executadas no SqlServer podem beneficiar-se do paralelismo. Paralelismo acontece



quando o SqlServer decide dividir o trabalho em mais de uma CPU. Quando executamos uma consulta muito pesada, o SqlServer pode optar por dividir essa consulta em mais de uma CPU com o objectivo de agilizar o processamento da consulta.

Hoje em dia com o surgimento dos processadores de nova geração, tal como o i5 que possui 4 núcleos, enquanto uma consulta pode estar a ser executada em um núcleo, podemos ocupar o outro núcleo com um outro processo, dando a ilusão que os processos estão a executar ao mesmo tempo. No nosso projecto implementamos o paralelismo através do MAXDOP, que na nova versão do SqlServer ele já pode ser implementado por base de dados, nem todas bases de dados de uma instância necessitam do paralelismo, até porque pode ser crítico.

### **3.9.1.5 CTE**

Em uma base de dados relacional CTE é uma alternativa a subqueries, views e funções definidas pelo usuário. CTE é um conjunto de resultados temporários, derivados de uma consulta simples.

Como a CTE oferece vantagens como legibilidade, facilidade de manutenção de consultas complexas, por essa razão implementamos no nosso projecto em algumas consultas, sendo que ela divide a consulta em blocos de operação lógicos simples e separados.

### **3.9.1.6 Audit**

O audit ou auditoria que pode ser de todas bases de dados do SQL Server ou de uma base de dados individual envolve o rastreamento e o registro de eventos que ocorrem no mecanismo de base de dados. A auditoria do SQL Server permite criar auditorias de servidor, que podem conter especificações de auditoria de servidor para eventos de nível de servidor e especificações de auditoria de base de dados para eventos de nível de base de dados. Os eventos auditados podem ser gravados nos logs de eventos ou em arquivos de auditoria.

Com este recursos, o DBA do sistema pode saber quais utilizadores do servidor de base de dados executaram uma determinada query, isto irá garantir a possibilidade de fazer auditoria ao nível da base de dados.

### **3.9.1.7 Always encrypted**

É uma poderosa ferramenta que permite definir a configuração de criptografia desejada para colunas da base de dados. O assistente pode criptografar uma coluna, remover criptografia ou criptografá-la novamente (por exemplo, usando uma nova chave de criptografia de coluna ou um tipo de criptografia diferente de O tipo atual, configurado para a coluna). Várias colunas podem ser configuradas em uma única execução do assistente.

Esta ferramenta está presente no SqlServer desde a versão 2014, de certa forma ajudou-nos a manter a segurança, por mais que algum usuário falso tenha acesso indevido a base de dados ele não vai poder ler os dados ou seja perceber os dados de uma coluna criptografada, ao menos que tenha a chave.

### **3.9.2 Programação**

#### **3.9.2.1 ASP.NET 4.5.2**

O sistema foi construído no visual studio 2015 Enterprise, sobre a plataforma ASP.NET 4.5.2 utilizando a linguagem de programação C#.

Em ASP.NET existem dois modelos para programação: MVC e WebForms.

Para o nosso projecto utilizamos o ASP.NET MVC, que foi lançado em 2009, esta plataforma foi construída sobre o core do ASP.NET, permite à aplicação do padrão MVC na construção de aplicações Web.

O ASP.NET recebeu algumas actualizações, neste caso o ASP.NET Core 2.0 que é o ASP.NET open source, razões como essa levou-nos também a escolher o ASP.NET MVC, sem mencionar as vantagens que leva sobre o WebForms.

#### **3.9.2.2 Bootstrap**

Para o design do nosso sistema que abrange a camada de apresentação, utilizamos o famoso framework bootstrap, que vem se tornando num padrão web front-end.

O bootstrap é um framework front-end gratuito, utilizado para criação de layouts de forma mais produtiva, sendo que este framework já possui componentes prontos para serem reutilizados. Este framework possui actualmente o html 5, css3 e javascript.

#### **3.9.2.3 Validação**

Um aspecto muito importante a nível da camada de apresentação é a validação de dados. Nunca podemos ficar na expectativa que o utilizador insira valores coerentes com os da Base de Dados, porque, por exemplo, pode digitar dados numéricos no nome de uma pessoa para criar um novo registo, sem aperceber-se do acontecido.

Existem duas formas de validar os dados introduzidos pelo utilizador. Validações feitas no lado do cliente (clientside), ou seja, pelo browser, através da linguagem JavaScript ou do lado do servidor (serverside).

No nosso projecto adoptamos as duas formas, mas as validações serverside foram feitas apenas nas operações mais relevantes, que colocam totalmente em causa a integridade do sistema, como por exemplo, o cadastro de eleitor com maior de 18 anos.

### 3.9.2.4 jQuery

o jQuery é uma biblioteca de software especificada em JavaScript. Esta biblioteca contém um conjunto de módulos para implementação de funcionalidades de gestão de eventos, leitura interactiva dos conteúdos, diferentes tipos de animações, assim como para a implementação de efeitos com a tecnologia AJAX. Portanto para a validação no lado cliente utilizamos o jQuery.

The screenshot displays the 'SVE - SISTEMA DE VOTOS ELECTRÔNICOS' web application. On the left is a sidebar menu with options like 'Home', 'Gerir comissão', 'Gerir Usuário', 'Gerir grupo', 'Gerir partido', 'Gerir eleitor', 'Caderno eleitoral', 'Gerir SMS', and 'Feições eletrônicas'. The main content area shows a user profile for 'Romário Ângelo' (romario-etical@hotmail.com). The profile includes a 'Substituir' button and a 'Remover' button. Below this are form fields for 'Nome' (with a red error message 'Este campo é obrigatório.'), 'Sexo' (a dropdown menu with 'Selecciona o Sexo' and a red error message 'Este campo é obrigatório.'), 'Data de nascimento' (a date picker with 'dd/mm/aaaa' and a red error message 'Este campo é obrigatório.'), and 'Local de nascimento' (a dropdown menu with 'Luanda'). The footer indicates '© 2017 SVE - Sistema de Votos Electrônicos' and 'Version: 1.0.0'.

**Figura 14:**Exemplo de validação de dados no sistema

### 3.9.2.5 AngularJS

Esta biblioteca, essencialmente, acrescenta novos elementos e atributos à linguagem HTML. AngularJS é também muito aplicado na especificação de aplicações web contidas numa única página de interface HTML.

Por esta descrição, utilizamos esta biblioteca para ajudar a manipular o DOM, mas com esta biblioteca fomos mais além de modo a tirar maior proveito dela, sendo que ela possui uma arquitectura MVW (Model View Whatever), com essa estrutura criamos o famoso dropdownlist em cascata, utilizando chamadas assíncronas.

### **3.9.2.6 Inversão de controle**

Inversão de controle é o nome dado ao padrão de desenvolvimento de programas de computadores onde a sequência (controle) de chamadas dos métodos é invertida em relação à programação tradicional, ou seja, ela não é determinada diretamente pelo programador. Este controle é delegado a uma infraestrutura de software muitas vezes chamada de container ou a qualquer outro componente que possa tomar controle sobre a execução. Esta é uma característica muito comum a alguns frameworks.

#### **3.9.2.6.1 Injeção de dependência**

A injeção de dependência é uma das práticas de inversão de controle.

Módulos de alto nível não devem depender de módulos de baixo nível. Ambos devem depender de abstrações.

As abstrações devem depender de detalhes. Os detalhes devem depender das abstrações.

Com a injeção de dependência evitamos o forte acoplamento no nosso sistema, onde os módulos não têm muita dependência um do outro, limitando desta forma depender de muitas instâncias, mas sim de abstrações a serem implementadas por interfaces.

#### **3.9.2.6.2 Ninject**

O Ninject é container que utilizamos no nosso projecto. O Ninject é uma biblioteca de software aberto que providencia uma framework de injeção de dependências (Injeção de Dependência) leve, fácil de integrar e de utilizar.

### **3.9.2.7 Padrão repositório**

Repositório é uma camada da aplicação responsável por manter e persistir os objectos do modelo. Utilizamos repositório genérico no nosso projecto, de modo a termos um CRUD que servirá para todas as entidades, mas apenas para entidades que não envolvam regras em específico, ou seja para entidades que não precisam de um tratamento específico no seu CRUD, mas não bastou apenas utilizarmos este padrão, para termos este CRUD genérico também fizemos o uso de uma ferramenta ORM, conforme listado abaixo no próximo ponto.

### **3.9.2.8 Entity Framework 6.0**

O Entity Framework é uma solução criada pela Microsoft que utiliza o mapeamento das tabelas da base de dados são representadas por classe e o registo de cada uma são representados por instâncias da classe correspondente.

Com essa técnica, o programador não precisa se preocupar com os comandos SQL – apenas utiliza-se uma interface de programação simples. Desta forma com a ferramenta ORM não ficamos dependentes apenas do SqlServer que é o SGBD do nosso projecto, mas caso surgir necessidade de migrar para o Oracle por exemplo, não teríamos muito trabalho, só iríamos configurar o Entity Framework para o Oracle.

### **3.9.2.9 Scaffolding**

ASP.NET scaffolding é um gerador de código que foi introduzindo aquando do lançamento da versão 3.0 do ASP.NET MVC, permite referenciar uma determinada coleção no contexto criado pela Entity Framework e obter automaticamente um controlador e respectivas vistas com as operações de CRUD. Utilizamos o scaffolding para gerar cruds básicos ou seja aqueles que não têm complexidade de lógica de negócio.

### **3.9.2.10 Nuget**

Nuget é um gerenciador de pacotes .NET gratuito, desenvolvido pela Microsoft, que acabou por ser transformado num projecto Open Source, utilizamos o Nuget para fazer referência a algumas bibliotecas, visto que o Nuget não referencia apenas bibliotecas que se encontram no GAC, mas também pode transferir bibliotecas que se encontram em repositórios globais que tenham assinatura de autenticidade. Desta forma livramos o nosso projecto de bibliotecas que possam danifica-lo.

### **3.9.2.10 ASP.NET Identity**

Recorremos a está API para tratar de assuntos como o login, perfis e mais outros assuntos relacionados ao utilizador, é de salientar que com está API também é possível fazer o login a partir das redes sociais. Anteriormente era utilizado o MemberShip, mas não tinha compatibilidade com todas plataformas de alto nível, já o ASP.NET Identity é compatível com todas plataforma de alto nível, a nova API pode ser consumida a partir de aplicações ASP.NET WebForms, ASP.NET MVC, ASP.NET Web Pages, ASP.NET Web API e Signal R, é independente do mecanismo de persistência, facilita a escrita de testes unitários e muito mais.

### **3.9.2.11 LINQ**

A partir da versão 2008, o Visual Studio passou a integrar uma linguagem própria que permite consultar, actualizar e manusear uma base de dados. O LINQ encontra-se definida na namespace System.Data.Linq.

No nosso projecto utilizamos o Linq para efectuar consulta de dados, isto em alguns métodos em específico, portanto beneficiamos do Linq to objects.

### 3.9.2.12 ADO.NET

O ADO.NET é um conjunto de classes que expõem serviços de acesso a dados para desenvolvedores do .NET Framework. O ADO.NET fornece um conjunto rico de componentes para criar aplicativos distribuídos e de compartilhamento de dados. Faz parte do .NET Framework, fornecendo acesso a dados de aplicativo relacionais, XML, Json, e outros mais.

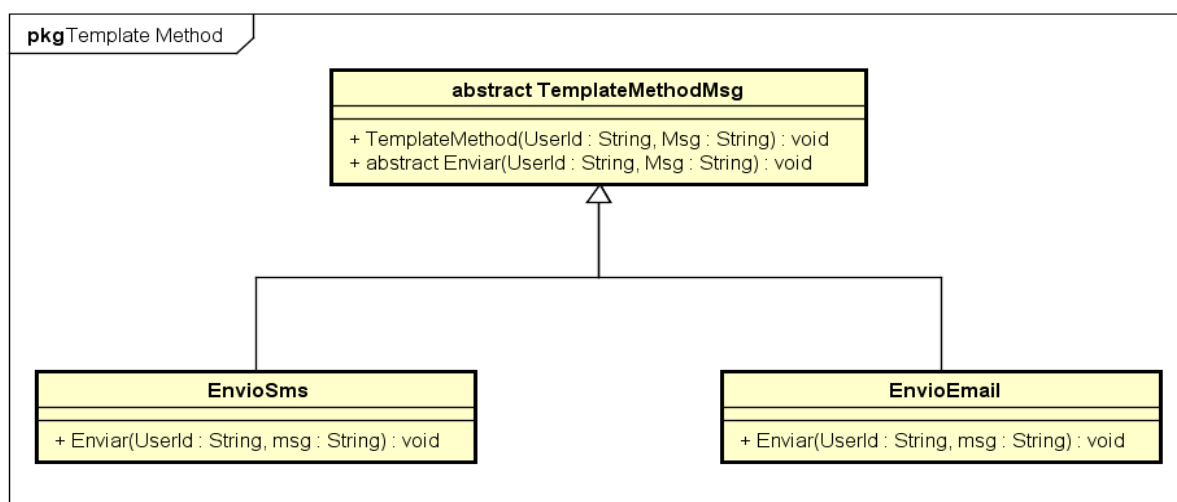
Levando em conta que em algumas partes do nosso sistema, tivemos que lidar com procedimentos armazenados, para chamar estes procedimentos utilizamos o ADO.NET, visto que ele é mais rápido que a nossa ferramenta ORM.

### 3.9.2.13 AutoMapper

Utilizamos esta API para fazer com que as nossas entidades de domínio mapeiam as nossas DTO que interage directamente com a camada de apresentação, visto que a camada de apresentação não pode ter acesso directo ao domínio.

### 3.9.2.14 Padrão de projecto Template Method

Uma coisa interessante que utilizamos aqui é o padrão de projecto **Template Method** que auxilia na definição de um algoritmo com partes do mesmo definidos por métodos abstratos. As subclasses devem se responsabilizar por estas partes abstratas, deste algoritmo, que serão implementadas, possivelmente de várias formas, ou seja, cada subclasse irá implementar à sua necessidade e oferecer um comportamento concreto construindo todo o algoritmo, abaixo temos o diagrama que descreve este cenário no nosso sistema.



powered by Astah

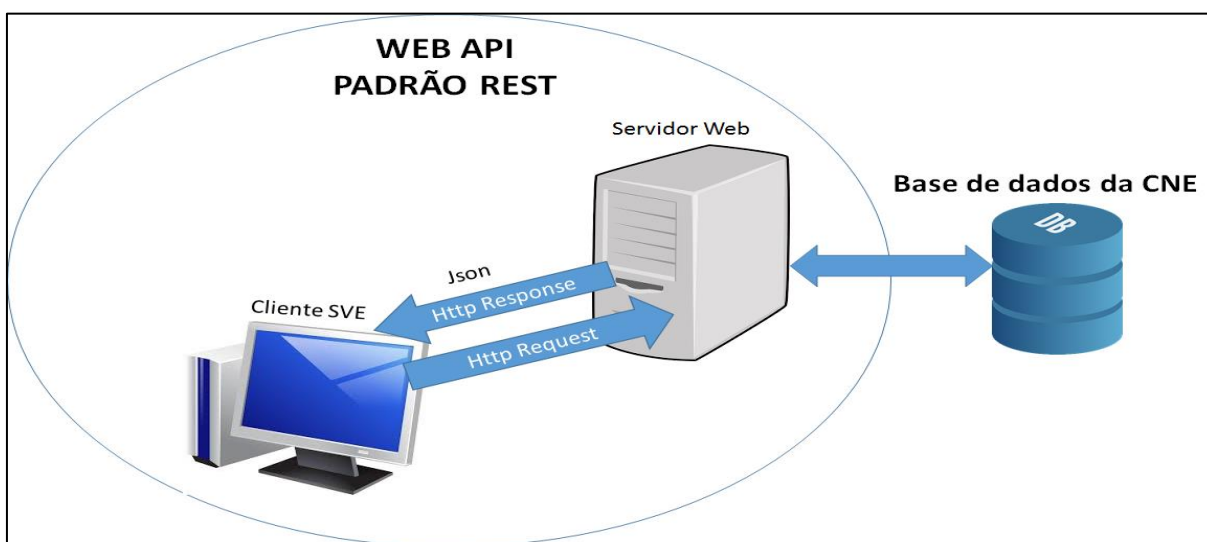
**Figura 15:** Padrão Template Method

### 3.9.2.15 Json

Json é uma formatação leve de troca de dados. Para seres humanos é fácil de ler e escrever. Para as máquinas é fácil de interpretar e gerar. No nosso projecto utilizamos alguns métodos que efectuem consultas e retornam Json para preencher os nossos dropdownlist.

### 3.9.2.16 ASPNET WEB API

Atendendo ao facto de já existir uma base de dados na CNE, tivemos então de fazer uma integração do nosso sistema com essa mesma base dados, através de uma WEB API que vêm como substituto ao famoso Web Service que é baseado em SOAP, trafegando apenas XML, já a Web API é baseado em REST e acaba por ser mais flexível, trafega Json e outros formatos e atende maior variedade de dispositivos, temos abaixo a figura que descreve essa integração:



**Figura 16:**Interoperabilidade do sistema

Levando em conta que a CNE já possui uma base de dados com os eleitores, fizemos a integração por intermédio de uma WEB API, na qual a base de dados da CNE continuará isolada do nosso sistema, apenas construímos os serviços que dependem da base de dados da CNE que o nosso sistema deverá consumir, para poder consultar os eleitores cadastrados para exercerem o sufrágio, essa WEB API é hospedada no servidor web da CNE, visto que eles possuem servidor web, o mesmo servidor vai disponibilizar os serviços baseados em REST, essa interoperabilidade entre os sistemas será feita via Json onde teremos a requisição e resposta entre o cliente e o servidor da CNE, o cliente é nosso sistema em questão.

Desta forma acabamos por ter a base de dados isolada do nosso sistema e as operações de acesso a mesma são encapsuladas na WEB API, com isto temos ganhos em termos de segurança, visto que a base de dados não precisa ser totalmente exposta.

Atendendo o facto dos serviços em REST atenderem maior variedade de dispositivos, isso nos abre a possibilidade de poder integrar a mesma WEB API numa aplicação mobile.

### **3.9.3 Implementação de segurança**

A nível de segurança na nossa aplicação, meios de prevenção aos ataques mais frequentes na web, que são: injeção de SQL, XSS, referência directa de objectos e CSRF.

Para prevenir da injeção de SQL, optamos por não utilizar código SQL dentro da nossa aplicação, e na maior parte trabalhamos com a nossa ORM que não recebe também código SQL, desta forma reduzimos a possibilidade deste tipo de ataque surtir efeito.

No caso do XSS o atacante tem a possibilidade de executar script no browser, podendo ser html. Para nos prevenirmos desta vulnerabilidade, nas nossas classes que actuam como DTOs não colocamos em nenhuma propriedade o atributo “AllowHtml”, este atributo permite injectar código html nos controles input, mas para reforçar ainda mais nos nossos controladores, deixamos o atributo “ValidateInput” activo.

A vulnerabilidade de referência directa de objectos, isso acontece quando temos muita exposição na nossa url, o usuário pode fazer uma requisição importante na url, caso não haver tratamento devido, a requisição é processada, na verdade foi tratada desde o começo do nosso projecto, a partir da arquitectura utilizada, a separação em camadas em si, já tira a possibilidade de fazer referência directa a um objecto, por isso criamos uma camada de domínio no nosso projecto e nela possui todas regras de negócio, portanto todas requisições importantes do usuário passam sempre pelo nosso domínio, e lá são verificados se cumprem com as normas para poder processar a requisição.

A vulnerabilidade de CSRF, permite o atacante conduzir a vítima a abrir uma url maliciosa, na verdade por trás da url (autentica) que a vítima abrir existe uma url numa tag para imagem, por exemplo: `<img src=http://www.banco.pt/transfere?quant=999;dest=013492 width='0' height='0'`, quando a vítima clicar na imagem o site vai abrir este link e processar a transferência, como no nosso caso poderia colocar os votos em causa. Para colmatar esta vulnerabilidade utilizamos “@Html.AntiForgeryToken” nos métodos que estão expostos na url e que desempenham um papel muito importante no nosso domínio.

### **3.9.4 Fase de testes**

A norma ISO/IEC 9126 relaciona um conjunto de características que devem ser verificadas em um software para que ele seja considerado um software de qualidade, para isso foram então realizados alguns testes no nosso projecto para se verificar a qualidade do mesmo.



#### **3.9.4.1 Teste de unidade**

Conhecido como Teste Unitário. É a fase do processo de teste em que se testam as menores unidades de software desenvolvidas (pequenas partes ou unidades do sistema). Os alvos são os métodos dos objectos ou módulo/função ou mesmo pequenos trechos de código. O objectivo é encontrar falhas de funcionamento em uma pequena parte do sistema funcionando independentemente do todo. Os testes de unidade são feitos pelo próprio programador geralmente usando o método caixa branca ou funcional.

Teste de unidade no sistema: levando em conta que utilizamos o TDD, verificamos o funcionamento de alguns métodos antes de implementá-lo, através de mock objects (objectos simulados), para ter a certeza de que funciona conforme o esperado.

#### **3.9.4.2 Teste de integração**

Na fase de teste de integração o objectivo é encontrar falhas provenientes da integração interna dos componentes de um sistema verificando se as unidades testadas individualmente executam de forma correcta quando integradas.

Nesta fase do projecto alguns elementos que forma desenvolvidos de forma isolada, ao integrar com os restantes elementos, testamos de modo a garantir a conformidade destes elementos.

#### **3.9.4.3 Teste de sistema**

O teste de sistema tem como objectivo executar o sistema sob ponto de vista de seu usuário final, varrendo as funcionalidades em busca de falhas. Os testes são executados em condições similares - de ambiente, interfaces sistémicas e massas de dados - àquelas que um usuário utilizará no seu dia-a-dia de manipulação do sistema. De acordo com a política de uma organização podem ser utilizadas condições reais de ambiente, interfaces sistémicas e massas de dados. No teste de sistema verifica-se o comportamento requisitado e a validação das funcionalidades especificadas pelo cliente.

Neste item, foram realizados alguns testes, para tal tivemos que colocar o sistema a disposição de alguns usuários. Tivemos um resultado positivo, surgindo poucos erros por corrigir.

#### 3.9.4.4 Teste de aceitação

Os testes de unidade verificam se os pequenos elementos do sistema como deveriam, mas não verificam se o sistema como um todo funciona correctamente. Os testes de unidade são testes de caixa branca, que verificam os mecanismos individuais do sistema. Os testes de aceitação são testes de caixa preta, que verificam se os requisitos do cliente estão sendo cumpridos, que normalmente é feito por pessoas que não conhecem os mecanismos internos do sistema, que nomeadamente podem ser o cliente ou mesmo um analista de negócio.

Então, para verificar a conformidade do sistema, verificou-se os seguintes elementos do sistema:

**Requisitos:** verificou-se que o sistema cumpre todos os requisitos preconizados.

**Processamento:** verificou-se que o sistema tem um tempo de resposta satisfatório.

**Validações e segurança:** as validações do sistema foram bem aplicadas, portanto verificou-se a conformidade. No que toca a segurança foram alcançadas todas as metas estabelecidas a nível de segurança, sendo assim o sistema não estará em causa no que toca a segurança, mas ainda assim o utilizador deve ser um colaborador para manter a segurança do sistema, levando em conta que nenhum sistema é 100% seguro.

## 4 RESULTADOS

### 4.1 Descrição do projecto

Como já mencionamos, este projecto consiste em desenvolver um sistema de votos electrónicos, possui 4 módulos que são: recenseamento, votação, contabilização e comunicação. Portanto o projecto permitirá que os eleitores exercem o sufrágio sem ter que sair da sua residência, a partir de um computador, com um browser, e que tenha acesso a internet.

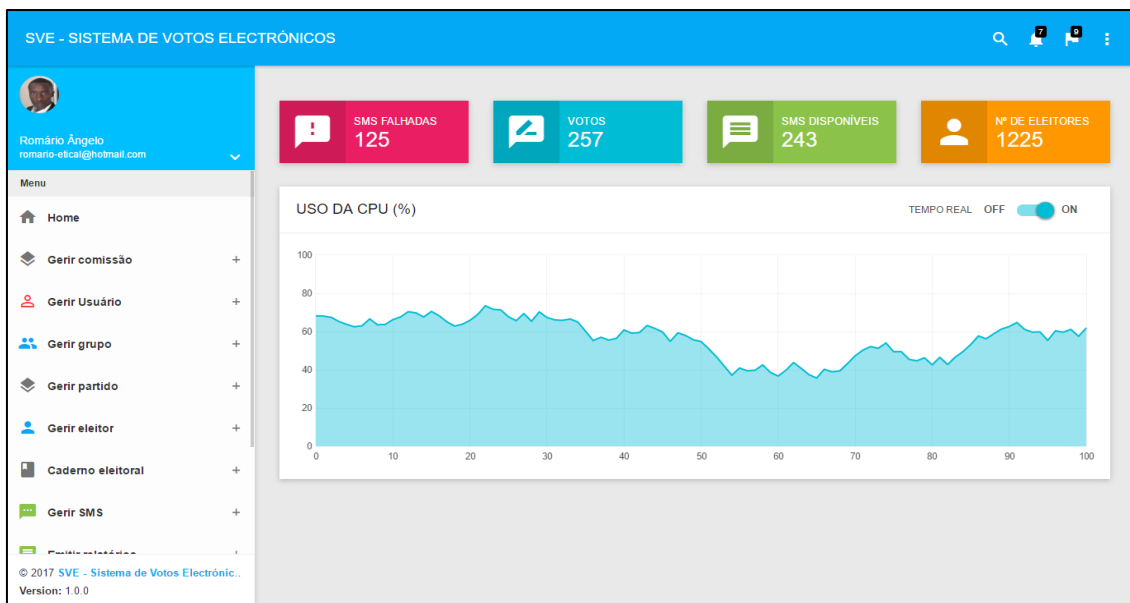
Procuramos seguir todos requisitos que são exigidos pelo sistema de votação. O eleitor poderá receber os resultados por sms, de modo a dinamizar a interação com o eleitor.

No nosso projecto procuramos integrar elementos programáticos de alto nível, no que toca ao designer procuramos seguir os padrões de usabilidade, de modo a tornar a experiência do usuário mais fácil.

O principal foco do sistema é permitir a votação, sem menosprezar as outras funcionalidades. Categorizamos os usuários por perfis, de modo a fazer a separação de privilégio por omissão.

### 4.2 Interface do sistema

#### 4.2.1 Tela principal do administrador



**Figura 17:**Tela do administrador

## 4.2.2 Tela de cadastro de eleitor

The screenshot displays the 'SVE - SISTEMA DE VOTOS ELECTRÓNICOS' interface. On the left is a sidebar menu with options like 'Home', 'Gerir comissão', 'Gerir usuário', 'Gerir grupo', 'Gerir partido', 'Gerir eleitor', 'Caderno eleitoral', 'Gerir SMS', and 'Fóruns eletrônicos'. The main area shows a three-step process: '1. Dados pessoais', '2. Dados do bilhete', and '3. Contacto-Fim'. The first step is active, showing a form for 'Romário Ângelo' (romario-elcal@hotmail.com). The form includes a profile picture with 'Substituir' and 'Remover' buttons, a 'Nome' field with 'Romário Ângelo', a 'Sexo' dropdown set to 'Masculino', a 'Data de nascimento' field with '01/01/1994', and a 'Local de nascimento' dropdown set to 'Luanda'. At the bottom right are 'Previous' and 'Next' buttons.

Figura 18:Tela de cadastro de eleitor

## 4.2.3 Tela de inclusão do eleitor ao caderno eleitoral

The screenshot displays the 'SVE - SISTEMA DE VOTOS ELECTRÓNICOS' interface for the 'Caderno eleitoral' section. The sidebar menu is identical to the previous screen. The main area is titled 'Caderno eleitoral' with the subtitle 'Incluir eleitor ao caderno eleitoral'. It features a search bar with the placeholder 'Ex: Nº Eleitor, BI, Email, Telefone' and a 'Pesquisar' button. Below this, a card displays the details of 'Kuenda Mayeye': 'Nome: Kuenda Mayeye', 'BI: |', 'Sexo: Masculino | Idade: 23', 'Email: kuenda@hotmail.com | Telefone: 934760768', 'Número do cartão: 643643659', and 'Estado: Activo'. To the right of the card is a profile picture. Below the card are four dropdown menus for 'Provincia' (Luanda), 'Município' (Cazenga), 'Bairro' (Madeira), and 'Rua' (Boxeiro). At the bottom is a 'Referência' field with the label 'Referência do local de voto' and an 'Incluir eleitor' button.

Figura 19:Tela de inclusão do eleitor ao caderno eleitoral

## 4.2.4 Tela de pesquisar eleitor no caderno eleitoral

SVE - SISTEMA DE VOTOS ELETRÔNICOS

Romário Ângelo  
romario-etical@hotmail.com

Menu

- Home
- Gerir comissão
- Gerir usuário
- Gerir grupo
- Gerir partido
- Gerir eleitor
- Caderno eleitoral
- Gerir SMS
- Fórum eletrônico

© 2017 SVE - Sistema de Votos Eletrônicos  
Version: 1.0.0

Caderno Eleitoral

Ano Eleitoral: 4 Estado: Votou Pesquisar

Copy CSV Excel PDF Print

Procurar: Man

Nome	Estado	Ano Eleitoral
Francisco Cafumana	Votou	4
Manuel Maria	Votou	4
Rosa Manuel	Ainda não votou	4

Mostrando 1 para 3 de 3 registros (filtered from 18 total entries)

Anterior 1 Proximo

Figura 20: Tela de pesquisar eleitor no caderno eleitoral

## 4.2.6 Tela de visualização de boletim de voto

SVE-Sistema de Votos Eletrônicos

HOME PARTIDOS VOTO RESULTADOS ELEITOR

Estás aqui / HOME / 404

MPLA

MPLA

UNITA

INFODEV

Votar

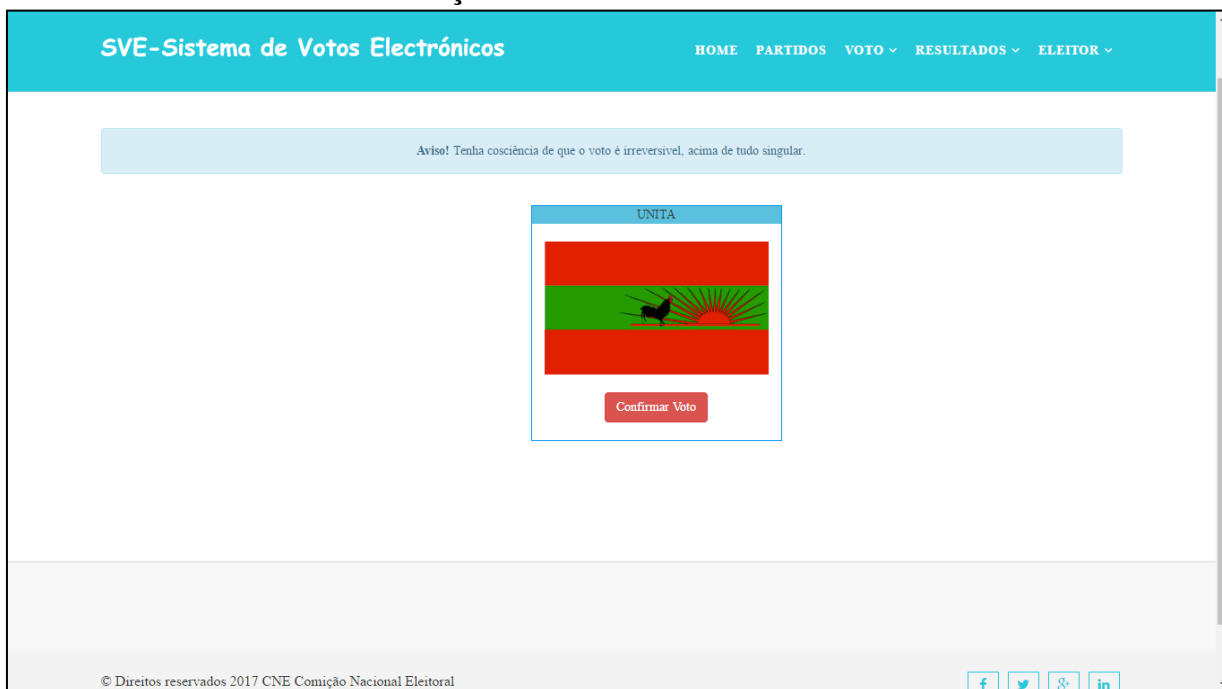
Votar

Votar

Votar

Figura 21: Tela de visualização de boletim de voto

#### 4.2.7 Tela de confirmação de voto



**Figura 22:**Tela de confirmação do voto

### 4.3 Análise dos resultados

Tendo em conta os resultados obtidos em função da nossa pesquisa, podemos dizer que o nosso trabalho atende os requisitos levantados, que na qual apostamos na usabilidade do usuário no que toca as interfaces gráficas, que são responsivas, isto é, se adaptam a qualquer telas, o manuseio do sistema é fácil, especificamente para o eleitor.

Apesar do sistema cumprir com os requisitos levantados, ainda assim nos deparamos com alguns pontos críticos, como o cumprimento do requisito de não-coercibilidade, que na qual torna – se difícil de garantir que o eleitor não partilha com mais alguém a sua opção de voto, sendo que ele pode votar em casa, portanto as possibilidades de violar este requisito são enormes, outro ponto crítico consiste na confiança do eleitor, o SVE ainda precisa ganhar confiança por parte dos eleitores, sendo que os SVU apresentam mais confiança e segurança aos eleitores.

## **CONCLUSÃO**

Visto que o nosso projecto consta no desenvolvimento de um sistema de votos electrónicos, em virtude dos factos mencionados, concluímos que as expectativas foram todas cumpridas, desta feita o nosso projecto atende todos os requisitos que foram levantados.

O projecto foi desenvolvido com as boas práticas de produção de software, não atendemos os requisitos de forma simplista apenas, mas procurarmos dar qualidade no nosso software, de modo a facilitar futuras manutenções e quaisquer outros eventos técnicos. É um projecto que promoveu também para nós alguns desafios, mas que de certa forma supre algumas lacunas encontradas no sistema de votação em urnas, conforme mencionamos anteriormente.

Apesar de cumprirmos com os requisitos levantados, ainda assim o nosso sistema conta com dificuldades encaradas como desafios, tal como a confiança dos eleitores e dos outros elementos envolvidos, a dificuldade de manipulação dos meios tecnológicos para muitos cidadão. Contudo essas dificuldades constatadas demandam um certo tempo para serem ultrapassadas, sendo que o SVE é apenas mais uma técnica, assim como o homem cria o martelo e o martelo recria o homem, o mesmo pode suceder com o SVE.

## **RECOMENDAÇÕES**

Sendo este um projecto que exige muito a nível de segurança, recomenda-se que as máquinas servidoras como também as máquinas do cliente devem ter sempre o antivírus actualizados.

Recomendamos também que futuramente desenvolva-se uma aplicação mobile, e uma WebApi para fornecer serviços à aplicação mobile.

Visto que o trabalho foi delimitado na implementação do SVE numa fase experimental apenas em Luanda, é recomendável que posteriormente é feita a continuidade dos estudos para poder implementar o SVE no país como um todo, isto é, implentar em todas as províncias.



## REFERÊNCIAS BIBLIOGRAFICAS

- Panga, Filipe. Vencer os desafios de redigir uma proposta e tese: Lições da metodologia de investigação científica. I. Local: CLC Printers (PVT) LTD.
- Correia, Miguel; Sousa, Paulo. Segurança no software. Local: FCA, 2010.
- Martin, Robert; Martin, Micah. Princípios, padrões e práticas ágeis em C#. I. Local: bookman, 2011.
- Loureiro, Henrique. C# 5.0 com Visual Studio 2012. I. Local: FCA, 2013.
- Nogueira, João. Confiança em votação electrónica. 2011.220. Engenharia e gestão de sistema de informação-Universidade do Minho, Lisboa. 2011
- Bezerra, Eduardo – Princípios de Análise e Projecto de Sistemas com UML, Local: editora Campos. 2006
- Silva, Alberto – Videira, Carlos – UML Metodologias e Ferramentas Cases 2ª Edição, Local: Centro Atlântico, 2008
- Abreu, Luís. ASP.NET 4.5.1. I. Local: FCA, 2014.
- Mendes, Douglas. Redes de Computadores. I. Local: FCA, 2014.
- Palhoto, Tiago. Desenvolvimento Ágil de Software. I. Local: FCA, 2016.
- Vasconcelos, José. Python Algoritmia e Programação Web. I. Local: FCA, 2015.
- Xavier, Amiraldes. Site de notícia online e fórum de informática. 2011.220. Engenharia informática-Universidade Técnica de Angola, Luanda. 2012
- Almeida, Marcos. Integração de sistemas de informação. 2010.220. Gestão da Informação e do Conhecimento-Universidade Federal do Paraná, Curitiba. 2010
- Cukier, Daniel – DDD-Introdução ao Domain Drive Design, 2010. Disponível em <<http://www.agileandart.com/2010/07/16/ddd-introducao-a-domain-driven-design/>>. Acesso em 17 de Abril de 2017
- Pires, Eduardo – Asp.Net Web API-Meu primeiro serviço REST, 2013. Disponível em <<http://www.eduardopires.net.br/2013/07/asp-net-web-api-meu-primeiro-servico-rest/>>. Acesso em 19 de Abril de 2017

## APÊNDICE I – DICIONÁRIO DE DADOS

**Tabela AnoEleitoral**

Nome Físico:	AnoEleitoral					
Nome Lógico:	AnoEleitoral					
Descrição:	Armazena os anos eleitoral					
Descrição	Nome	Tipo	Tam.	Nulo	PK	FK
Código do ano eleitoral	IdAno	INT		Não	Sim	
Descrição do ano	Descricao	VARCHAR	4	Não		
Observação	Obs	VARCHAR	500	Sim		
Estado da eleição: Início/Fim	EstadoEleicao	BIT	1	Não		
Data do registo	DataRegisto	DATETIME	4000	Não		
Estado do registo: Activo/Inactivo	Status	BIT		Não		
Código do usuário	IdUsuario	NVARCHAR		Não		Sim

**Tabela 8:**AnoEleitora/

**Tabela Eleitor**

Nome Físico:	Eleitor					
Nome Lógico:	Eleitor					
Descrição:	Armazena os dados do eleitor					
Descrição	Nome	Tipo	Tam.	Nulo	PK	FK
Código do eleitor	IdEleitor	INT		Não	Sim	
Nome do eleitor	Nome	VARCHAR	50	Não		
Número do cartão de eleitor	NRegisto	INT		Não		
Sexo do eleitor	Sexo	CHAR	1	Não		
Data de nascimento do eleitor	DataNascimento	Date		Não		
Local de nascimento(Provincia)	LocalNascimento	TYNINT		Não		Sim
Foto do eleitor	Foto	VARBINARY	MAX	Não		

Email do eleitor	Email	VARCHAR	80	Sim		
Número do telefone do eleitor	Telefone	VARCHAR	15	Sim		
Data de registo	DataRegisto	DATETIME		Não		
Estado do registo:Activo/Inactivo	Status	BIT		Não		
Código do grupo	IdGrupo	INT		Não		Sim
Código do usuário	IdUsuario	NVARCHAR		Não		Sim

**Tabela 9:**Eleitor

## Tabela BI

Nome Físico:	BI					
Nome Lógico:	BI					
Descrição:	Armazena dados do bilhete do eleitor					
Descrição	Nome	Tipo	Tam.	Nulo	PK	FK
Código do eleitor	IdEleitor	INT		Não	Sim	
Número do bilhete	NBI	VARCHAR	30	Não		
Entidade emissora	EntidadeEmissora	TINYINT		Não		Sim
Data de registo	DataRegisto	DATETIME		Não		
Estado do registo: Activo/Inactivo	Status	BIT		Não		
Código do usuário	IdUsuario	NVARCHAR		Não		Sim

**Tabela 10:**BI

## Tabela CadernoEleitoral

Nome Físico:	CadernoEleitoral					
Nome Lógico:	CadernoEleitoral					
Descrição:	Inclui os eleitores que podem votar, com a referência do local					
Descrição	Nome	Tipo	Tam.	Nulo	PK	FK
Código do caderno	IdCaderno	INT		Não	Sim	
Código do eleitor	IdEleitor	INT		Não		Sim
Código do ano eleitoral	IdAnoEleitoral	INT		Não		Sim
Código da rua	IdRua	INT		Não		Sim
Ponto de referência do local de voto	Referencia	VARCHAR	500	Não		
Data de registo	DataRegisto	DATETIME		Não		
Estado do registo: Activo/Inactivo	Status	BIT		Não		
Observação	Obs	VARCHAR	500	Sim		
Estado do eleitor	IdEstadoEleitor	TINYINT		Não		Sim
Código do usuário	IdUsuario	NVARCHAR		Não		Sim

**Tabela 11:**Caderno Eleitoral

## Tabela EstadoEleitor

Nome Físico:	EstadoEleitor					
Nome Lógico:	EstadoEleitor					
Descrição:	Armazena os possíveis estados de um eleitor no acto eleitoral					
Descrição	Nome	Tipo	Tam.	Nulo	PK	FK
Código do estado do eleitor	IdEstado	TINYINT		Não	Sim	
Descrição do estado	Descricao	VARCHAR	20	Não		
Data de registo	DataRegisto	DATETIME		Não		
Estado do registo: Activo/Inactivo	Status	BIT		Não		

**Tabela 12:**Estado Eleitor

## Tabela Mandato

Nome Físico:	Mandato					
Nome Lógico:	Mandato					
Descrição:	Armazena o mandato					
Descrição	Nome	Tipo	Tam.	Nulo	PK	FK
Código do mandato	IdMandato	TINYINT		Não	Sim	
Ano de mandato	Ano	CHAR	2	Não		
Data de registo	DataRegisto	DATETIME		Não		
Estado do registo: Activo/Inactivo	Status	BIT		Não		
Código do usuário	IdUsuario	NVARCHAR	128	Não		Sim

**Tabela 13:**Mandato

## Tabela Comissao

Nome Físico:	Comissao					
Nome Lógico:	Comissao					
Descrição:	Armazena as comissões					
Descrição	Nome	Tipo	Tam.	Nulo	PK	FK
Código da comissão	IdComissao	INT		Não	Sim	
Descrição da comissão	Descricao	VARCHAR	80	Não		
Código do tipo de comissão	IdTipoComissao	TINYINT		Não		Sim
Observação	Obs	VARCHAR	800	Sim		
Data de registo	DataRegisto	DATETIME		Não		
Código da rua em que se localiza	IdRua	INT		Não		Sim
Código do usuário	IdUsuario	NVARCHAR	128	Não		Sim
Estado do registo: Activo/Inactivo	Status	BIT		Não		

**Tabela 14:**Comissao

### Tabela TipoComissao

Nome Físico:	TipoComissao					
Nome Lógico:	TipoComissao					
Descrição:	Armazena os tipos de comissões					
Descrição	Nome	Tipo	Tam.	Nulo	PK	FK
Código do tipo de comissão	IdTipoComissao	TINYINT		Não	Sim	
Descrição do tipo de comissão	Descricao	VARCHAR	30	Não		
Data de registo	DataRegisto	DATETIME		Não		
Estado do registo: Activo/Inactivo	Status	BIT		Não		

**Tabela 15:** Tipo Comissao

### Tabela Provincia

Nome Físico:	Provincia					
Nome Lógico:	Provincia					
Descrição:	Armazena todas as provincias					
Descrição	Nome	Tipo	Tam.	Nulo	PK	FK
Código da província	IdProvincia	TINYINT		Não	Sim	
Descrição da província	Descricao	VARCHAR	50	Não		

**Tabela 16:** Provincia

## Tabela Municipio

Nome Físico:	Municipio					
Nome Lógico:	Municipio					
Descrição:	Armazena todos municípios					
Descrição	Nome	Tipo	Tam.	Nulo	PK	FK
Código do município	IdMunicipio	SMALLINT		Não	Sim	
Descrição do município	Descricao	VARCHAR	50	Não		
Código da província	IdProvincia	TINYINT		Não		Sim

**Tabela 17:**Município

## Tabela Bairro

Nome Físico:	Bairro					
Nome Lógico:	Bairro					
Descrição:	Armazena os bairros					
Descrição	Nome	Tipo	Tam.	Nulo	PK	FK
Código do bairro	IdBairro	INT		Não	Sim	
Descrição do bairro	Descricao	VARCHAR	50	Não		
Código do município	IdMunicipio	SMALLINT		Não		Sim

**Tabela 18:**Bairro

## Tabela Rua

Nome Físico:	Rua					
Nome Lógico:	Rua					
Descrição:	Armazena as ruas					
Descrição	Nome	Tipo	Tam.	Nulo	PK	FK
Código da rua	IdRua	INT		Não	Sim	
Descrição da rua	Descricao	VARCHAR	50	Não		
Código do bairro	IdBairro	INT		Não		Sim

**Tabela 19:**Rua

## Tabela Grupo

Nome Físico:	Grupo					
Nome Lógico:	Grupo					
Descrição:	Armazena os grupo					
Descrição	Nome	Tipo	Tam.	Nulo	PK	FK
Código do grupo	IdGrupo	INT		Não	Sim	
Descrição do grupo	Descricao	VARCHAR	50	Não		
Código da comissao	IdComissao	INT		Não		Sim
Observação	Obs	VARCHAR	800	Sim		
Código do usuário	IdUsuario	NVACHAR	128	Não		Sim
Código do bairro	IdBairro	INT		Não		Sim
Data de registo	DataRegisto	DATETIME		Não		
Estado do registo: Activo/Inactivo	Status	BIT		Não		

**Tabela 20:**Grupo



### Tabela Boletim

Nome Físico:	Boletim					
Nome Lógico:	Boletim					
Descrição:	Armazena os boletins de voto					
Descrição	Nome	Tipo	Tam.	Nulo	PK	FK
Código do boletim	IdBoletim	INT		Não	Sim	
Código do partido	IdPartido	TINYINT		Não		Sim
Código do ano eleitoral	IdAnoEleitoral	INT		Não		Sim
Observação	Obs	VARCHAR	800	Sim		
Código do usuário	IdUsuario	NVACHAR	128	Não		Sim
Data de registo	DataRegisto	DATETIME		Não		
Estado do registo: Activo/Inactivo	Status	BIT		Não		

**Tabela 21:**Boletim

### Tabela Partido

Nome Físico:	Partido					
Nome Lógico:	Partido					
Descrição:	Armazena os partidos políticos					
Descrição	Nome	Tipo	Tam.	Nulo	PK	FK
Código do partido	IdPartido	TINYINT		Não	Sim	
Descrição do partido	Descricao	VARCHAR	50	Não		
Proposta do partido	Proposta	VARCHAR	1000	Não		
Data de existência	DataExistencia	DATE		Não		
Observação	Obs	VARCHAR	800	Sim		
Bandeira do partido	Imagem	VARBINARY	MAX	Não		
Código do usuário	IdUsuario	NVACHAR	128	Não		Sim
Data de registo	DataRegisto	DATETIME		Não		
Estado do registo: Activo/Inactivo	Status	BIT		Não		

**Tabela 22:**Partido

## Tabela Voto

Nome Físico:	Voto					
Nome Lógico:	Voto					
Descrição:	Armazena os votos					
Descrição	Nome	Tipo	Tam.	Nulo	PK	FK
Código do voto	IdVoto	UNIQUEIDENTIFIER		Não	Sim	
Código do boletim	IdBoletim	INT		Não		Sim
Código do bairro	IdBairro	INT		Não		Sim
Data de registo	DataRegisto	DATETIME		Não		
Estado do registo: Activo/Inactivo	Status	BIT		Não		

**Tabela 23:**Voto

## Tabela CarregamentoSms

Nome Físico:	CarregamentoSms					
Nome Lógico:	CarregamentoSms					
Descrição:	Armazena os carregamentos de sms					
Descrição	Nome	Tipo	Tam.	Nulo	PK	FK
Código do carregamento	IdCarregamento	INT		Não	Sim	
Quantidade de mensagens	Quantidade	INT		Não		
Código do usuário	IdUsuario	NVARCHAR	128	Não		Sim
Data de registo	DataRegisto	DATETIME		Não		
Estado do registo: Activo/Inactivo	Status	BIT		Não		

**Tabela 24:**CarregamentoSms

### Tabela EnvioSms

Nome Físico:	EnvioSms					
Nome Lógico:	EnvioSms					
Descrição:	Armazena as sms enviadas					
Descrição	Nome	Tipo	Tam.	Nulo	PK	FK
Código da sms enviada	IdEnvioSms	INT		Não	Sim	
Código do usuário	IdUsuario	NVARCHAR	128	Não		Sim
Data de registo	DataRegisto	DATETIME		Não		
Estado do registo: Activo/Inactivo	Status	BIT		Não		

**Tabela 25:**EnvioSms

### Tabela CategoriaUsuario

Nome Físico:	CategoriaUsuario					
Nome Lógico:	CategoriaUsuario					
Descrição:	Armazena as categorias dos usuários					
Descrição	Nome	Tipo	Tam.	Nulo	PK	FK
Código da categoria	IdCategoria	TINYINT		Não	Sim	
Descrição da categoria	Descricao	VARCHAR	50	Não		
Data de registo	DataRegisto	DATETIME		Não		
Estado do registo: Activo/Inactivo	Status	BIT		Não		

**Tabela 26:**CategoriaUsuario

**Tabela AspNetUsers**

Nome Físico:	Usuario					
Nome Lógico:	AspNetUsers					
Descrição:	Armazena as credências de acesso dos  utilizadores do sistema					
Descrição	Nome	Tipo	Tam.	Nulo	PK	FK
Código do usuário	Id	NVARCHAR	128	Não	Sim	
Email do usuário	Email	NVARCHAR	MAX	Sim		
Confirmação do email	EmailConfirmed	BIT		Não		
Palavra-passe criptografada	PasswordHash	NVARCHAR	MAX	Não		
Segurança dos dados usuário	SecurityStamp	NVARCHAR	MAX	Não		
Telemóvel	PhoneNumber	NVARCHAR	MAX	Sim		
Confirmação do telemóvel	PhoneConfirmed	BIT		Não		
Habilitação de dois factores	TwoFactorEnabled	BIT		Não		
Data do término da sessão	LockoutEndUtc	DATETIME		Sim		
Habilitar controle do término de sessão	LockoutEnabled	BIT		Não		
Contar tentativa de acesso errada	AccessFailedCount	INT		Não		
Estado do registo:Activo/Inactivo	Status	BIT		Não		
Nome do usuário	UserName	NVARCHAR	256	Não		
Categoria do usuário	IdCategoria	TINYINT		Não		Sim
Data de registo	DataRegisto	DATETIME		Não		

**Tabela 27:**AspNetUsers

**Tabela Recenseador**

Nome Físico:	Recenseador					
Nome Lógico:	Recenseador					
Descrição:	Armazena os dados do recenseador					
Descrição	Nome	Tipo	Tam.	Nulo	PK	FK
Código do recenseador	IdRecenseador	INT		Não	Sim	
Nome do recenseador	Nome	VARCHAR	50	Não		
Número do bilhete do eleitor	BI	VACRHAR	15	Não		
Sexo do recenseador	Sexo	CHAR	1	Não		
Data de nascimento do recenseador	DataNascimento	Date		Não		
Foto do recenseador	Foto	VARBINARY	MAX	Não		
Email do recenseador	Email	VARCHAR	80	Sim		
Número do telefone do recenseador	Telefone	VARCHAR	15	Sim		
Data de registo	DataRegisto	DATETIME		Não		
Estado do registo:Activo/Inactivo	Status	BIT		Não		

**Tabela 28:**Recenseador

## APÊNDICE II – ORÇAMENTO

Tabela de orçamentos			
Descrição	Qtd	Valor	Total
Servidor de base de dados(HP ML350-6 núcleos)	2	823.140,00 kz	1.646.280,00 kz
Microsoft SqlServer 2016 Enterprise	3	Licença: 2.364.941,91 kz	7.094.823,00 kz
Windows Server 2016 R2(Data Center)	1	Licença: 1.021.138,64 kz	1.021.138,64 kz
Servidor de comunicação e backup(HP DL160-4 núcleos)	1	584.100,00 kz	584.100,00 kz
Storage Qnap Ts431 4x2tb	1	3.152.410 kz	3.152.410 kz
Computador HP 280 MT	135	183.600,00 kz	24.786.000,00 kz
Antivírus Kasperky 2017 Total Security	137	8.100 kz	1.109.700 kz
Equipa de desenvolvimento		20.000.000 kz	
Formação para uso do sistema		1.500.000 kz	
Manutenção emergente		2.000.000 kz	
Manutenção preventiva		1.000.000 kz	
Total		63.894.451 kz	

Tabela 29:Orçamento

### **APÊNDICE III – ROTEIRO DA ENTREVISTA**

- De onde começa o processo eleitoral?
- Quais são as regras que definem o intervalo do ano eleitoral?
- Quais são os documentos necessários para fazer o recenseamento eleitoral?
- Quantos partidos políticos existem em Angola?
- Como é feita a contagem dos votos?
- É possível fazer a recontagem dos votos?
- Como é mantida a singularidade do voto?
- Quantos eleitores existem actualmente?
- É possível votar antecipadamente?
- O que acontece com os cidadãos que se encontram no estrangeiro no momento da votação?
- Onde são guardados os votos?
- Quando é que é dado como terminado o processo eleitoral?
- Quanto tem custado financeiramente a organização do processo eleitoral?
- O que é o caderno eleitoral?

## APÊNDICE IV – EXEMPLO DA ENTREVISTA

### *Entrevista realizada ao funcionário da CNE*

**Romário Ângelo(RA):** O senhor poderia dizer-me de onde começa o processo eleitoral?

**Funcionário CNE(FC):** Sim, posso, o processo eleitoral começa pelo recenseamento eleitoral, esse processo nos permite incluir o eleitoral ao caderno eleitoral.

**RA:** O que é o caderno eleitoral?

**FC:**Na verdade, o caderno eleitoral é uma lista dos eleitores que vão exercer o sufrágio.

**RA:** Quais são as regras que definem o intervalo do ano eleitoral?

**FC:**Isso depende do mandato actual, se o mandato for de 4 anos, então o intervalo será de 4 anos também.

**RA:** Quantos partidos políticos existem no país?

**FC:**Existem 12 partidos, contando com aqueles que foram extintos.

**RA:** Como é feita a contagem dos votos?

**FC:** Os votos são contados manualmente, ou seja, após terminado o processo de votação, algumas pessoas reúnem-se para contar os boletins de forma agrupada.

**RA:** É possível fazer a recontagem dos votos?

**FC:** Sim, é possível, isto só acontece quando algum partido manifesta alguma insatisfação com o resultado e apresenta argumentos para obrigar a recontagem, visto que é um processo cansativo e demorado.

**RA:** Como é mantida a singularidade do voto?

**FC:** Após o eleitor exercer o sufrágio, lhe é pintado o dedo indicador, com um marcador e também apontamos na nossa lista, este é o meio que utilizamos para evitar que um eleitor vote mais de 1 vez num ano eleitoral.

**RA:** Quantos eleitores existem actualmente?

**FC:** Actualmente temos mais de 3 milhões de eleitores, esta informação não podemos lhe dar com precisão, mas o número tende a aumentar até ao término do recenseamento.



**RA:** É possível votar antecipadamente?

**FC:** Este é um caso que ainda não foi aprovado, está em discussão, mas a princípio não é possível.

**RA:** O que é que acontece com os cidadãos que se encontram no estrangeiro no momento de votação?

**FC:** Simples, o voto deles é dado como nulo, o sistema de votação actual obriga a presença física do eleitor na urna.

**RA:** Onde são guardado os boletim de votos preenchidos?

**FC:** Infelizmente essa informação não pode ser divulgada, é de carácter confidencial, mas o certo é que vão para um local seguro.

**RA:** Quando é dado como terminado o processo eleitoral?

**FC:** É daod como terminado quando são publicado os resultados.

**RA:** Quanto tem custado financeiramente a organização do processo eleitoral?

**FC:** Essa é uma informação sigilosa, mas como o senhor Adão de Almeida, que é o secretário de estado para assuntos da administração do território, divulgou esse valor, então eu vou me baseiar no que ele disse, o orçamento global é de 70 mil milhões de kwanzas, equivalente a 230 milhões de euros.

## **ANEXOS**