



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO SERTÃO
PERNAMBUCO
CURSO DE GESTÃO DA TECNOLOGIA DA INFORMAÇÃO**

VANDERLEI AFONSO DA SILVA

**SISTEMA CONTVOTO: UMA PROPOSTA DE AUTOMAÇÃO DO PROCESSO
ELEITORAL DO INSTITUTO FEDERAL DO SERTÃO PERNAMBUCANO**

FLORESTA-PE
2018

Dados Internacionais de Catalogação na Publicação (CIP)

S586s Silva, Vanderlei Afonso da
Sistema contvoto: uma proposta de automação do processo eleitoral do Instituto Federal do Sertão Pernambucano. / Vanderlei Afonso da Silva - Floresta, 2018.

144 f. il.

Orientador: Paulo Thiago Lima do Nascimento .
Trabalho de Conclusão de Curso – Tecnólogo em Gestão da Tecnologia da Informação Instituto Federal de Educação, Ciência e Tecnologia do Sertão Pernambucano – Campus Floresta.

1. Gerenciamento de software. 2. Sistema. 3. Programação.

I. Nascimento, Paulo Thiago Lima do . II. Título.

CDD: 005.629

VANDERLEI AFONSO DA SILVA

**SISTEMA CONTVOTO: UMA PROPOSTA DE AUTOMAÇÃO DO PROCESSO
ELEITORAL DO INSTITUTO FEDERAL DO SERTÃO PERNAMBUCANO**

Monografia apresentada como requisito para obtenção da graduação no curso de Gestão da Tecnologia da Informação, do Instituto Federal de Educação, Ciência e tecnologia do Sertão Pernambucano - Campus Floresta.

Orientador: Prof. Esp. Paulo Thiago Lima do Nascimento.

FLORESTA-PE

2018

VANDERLEI AFONSO DA SILVA

**SISTEMA CONTVOTO: UMA PROPOSTA DE AUTOMAÇÃO DO PROCESSO
ELEITORAL DO INSTITUTO FEDERAL DO SERTÃO PERNAMBUCANO**

Aprovado em: ___/___/_____

Nota: _____

BANCA EXAMINADORA

Professor Esp. Paulo Thiago Lima do Nascimento – Orientador
Instituto Federal do Sertão pernambucano – Campus Floresta

Professor Msc. Elismar Moraes dos Santos - Avaliador
Instituto Federal do Sertão pernambucano – Campus Floresta

Professor Msc. Severino do Ramo de Paiva - Avaliador
Instituto Federal do Sertão pernambucano – Campus Floresta

Dedico este trabalho à minha filha, esposa e familiares que de alguma forma me ajudaram nos momentos em que necessitei.

AGRADECIMENTOS

À minha esposa por está sempre comigo e entender minhas dificuldades para conciliar trabalho com estudos, situação que por muitas vezes acaba por nos distanciar.

À minha filha por ser o motivo de me manter firme e buscar sempre o melhor para nossa família, servindo de exemplo de vida para mesma no futuro.

E aos professores do Instituto Federal - *campus* Floresta, por influenciar na minha vida como educadores, me tornando uma pessoa melhor.

“Se o dinheiro for a sua esperança de independência, você jamais a terá. A única segurança verdadeira consiste numa reserva de sabedoria, de experiência e de competência.”

Henry Ford.

RESUMO

Com o uso da Engenharia de *Software* a Justiça Eleitoral Brasileira modernizou e otimizou seu processo de contabilização e apresentação de resultado das eleições no país. Esta evolução reflete as mudanças pela qual passa o mundo, sobre tudo nas sucessivas criações de novos sistemas para automatizar e facilitar a execução de tarefas que, de certa forma, são cansativas e demoradas. Este trabalho tem como objetivo mostrar os eventos que levaram a criação da urna eletrônica brasileira, as técnicas, ferramentas e linguagem de programação existentes para a construção de um sistema computacional similar ao equipamento da Justiça Eleitoral para o uso do mesmo nas eleições realizadas pelo Instituto Federal do Sertão conforme suas legislações, visto esta instituição, nesses tempos modernos, fazer uso de trabalhos manuais para escolha do pessoal que ocupa cargo eletivo em seu quadro.

Palavras chaves: Engenharia de *Software*, sistema, programação.

ABSTRACT

With the use of Software Engineering, the Brazilian Electoral Court modernized and optimized its accounting process and presentation of the results of the country's elections. This evolution reflects the changes that the world is going through, especially in the successive creations of new systems to automate and facilitate the execution of tasks that are, in a way, tiring and time-consuming. This paper aims to show the events that led to the creation of the Brazilian electronic ballot box, the techniques, tools and programming language available for the construction of a computer system similar to the equipment of the Electoral Justice for the use of the same in the elections held by the Federal Institute of the Sertão according to its legislations, since this institution, in these modern times, makes use of manual labor to choose the personnel that occupies elective position in its picture.

Key words: Software Engineering, system, programming.

LISTA DE ABREVIATURAS E SIGLAS

API - Application Programming Interface
Fa - Fluxo alternativo
FDD - Feature Driven Development
Fe - Fluxo de exceção
GTI - Gestão da Tecnologia da Informação.
GUI - *Graphical User Interface*.
IDE - Integrated Development Environment
IF - Instituto Federal.
JDBC - *Java Database Connectivity*.
JVM - *Java Virtual Machine*.
PC - Personal Computer
PDF - Portable Document Format
RF - Requisito funcional.
RNF - Requisito não funcional.
SGBD - Sistema de Gerenciamento de Banco de Dados.
SO - Sistema Operacional
SQL - Structured Query Language.
TRE - Tribunal Regional Eleitoral.
TSE - Tribunal Superior Eleitoral.
UML - *Unified Modeling Language*.
XP - Extreme Programming.

LISTA DE FIGURAS

Figura 1: Modelo em cascata.	23
Figura 2: Desenvolvimento incremental.	24
Figura 3: Especificações dirigidas a planos, e ágeis.	25
Figura 4: Ciclo de vida de um release em Extreme Programming.	26
Figura 5: Fluxo das atividades no Scrum.	28
Figura 6: Fluxo das atividades do FDD.	28
Figura 7: Usuários de um documento de requisitos.	31
Figura 8: Diagrama de caso de uso (simples).	34
Figura 9: Diagrama de caso de uso (include).	34
Figura 10: Diagrama de caso de uso (extend).	35
Figura 11: Diagrama de caso de uso (generalização).	35
Figura 12: Descrição detalhada de caso de uso.	36
Figura 13: Elementos de um diagrama de classes.	37
Figura 14: Diagrama de classes (características de uma associação).	38
Figura 15: Exemplo de classe associativa e associação ternária.	39
Figura 16: Exemplo de agregação e composição.	39
Figura 17: Exemplo de relacionamento de herança.	40
Figura 18: Elementos básicos de um diagrama de sequência.	41
Figura 19: Nome do diagrama, referências e mensagens com notação.	41
Figura 20: Diagrama de sequência com operadores.	42
Figura 21: Elementos de um diagrama de máquina de estados.	43
Figura 22: Elementos de um diagrama de utilização.	44
Figura 23: Tabelas de um banco de dados relacional.	45
Figura 24: Tabelas com atributos e relacionamentos.	46
Figura 25: Processo de criação de um arquivo executável.	47
Figura 26: Java multiplataforma.	48
Figura 27: Declaração de variáveis em Java.	49
Figura 28: Declaração de método em Java.	49
Figura 29: Criação e composição de uma classe.	50
Figura 30: Criação e manipulação de objetos em Java.	51
Figura 31: Relacionamento entre objetos em Java.	51
Figura 32: Manipulação de objetos com atributos protegidos.	52
Figura 33: Criando uma tela com Java.	53
Figura 34: Tela com JPanel, JTextField e JLabel.	54
Figura 35: Utilizando a classe JButton.	54
Figura 36: Java e JDBC.	55
Figura 37: Comando <i>create</i>	56
Figura 38: Comando <i>insert</i>	56
Figura 39: Comando <i>update</i>	57
Figura 40: Comando <i>select</i>	57
Figura 41: Comando <i>delete</i>	58
Figura 42: Fórmula (percentual de votos obtidos por um candidato).	62
Figura 43: Fórmula (percentual de votos em branco).	62
Figura 44: Fórmula (percentual de votos nulos).	63
Figura 45: Requisito Funcional (Impedir registros duplicados) do sistema ContVoto.	66
Figura 46: Requisito Não Funcional (Somente rede local) do sistema ContVoto.	66

Figura 47: Diagrama de utilização do sistema ContVoto.....	67
Figura 48: Diagrama de casos de uso do sistema ContVoto.	68
Figura 49: Tela principal do sistema ContVoto.....	69
Figura 50: Tela de identificação de eleitor do sistema ContVoto.....	69
Figura 51: Detalhamento do caso de uso (Autoidentificação de eleitor) do sistema ContVoto.	70
Figura 52: Diagrama de classes do sistema ContVoto.....	71
Figura 53: Modelo relacional do sistema ContVoto.	72
Figura 54: Diagrama de sequência (Votar) do sistema ContVoto.	73
Figura 55: Diagrama de máquina de estado das interfaces gráficas do módulo Servidor - Sistema ContVoto.	74
Figura 56: Diagrama de máquina de estado das interfaces gráficas do módulo Cliente - Sistema ContVoto.	74
Figura 57: Desenvolvimento do <i>software</i> na <i>IDE NetBeans</i>	76
Figura 58: Gerenciamento de banco de dados no <i>MySQL Workbench</i>	77
Figura 59: Tela " <i>Login</i> " do sistema ContVoto - Servidor.	78
Figura 60: Tela "principal" do sistema ContVoto - Servidor.....	79
Figura 61: Tela de "cadastro de eleitor" do sistema ContVoto - Servidor.....	80
Figura 62: Tela de "gerenciamento de candidato" do sistema ContVoto - Servidor. .	80
Figura 63: Tela "criar uma eleição" do sistema ContVoto - Servidor.....	81
Figura 64: Resultado de uma eleição do sistema ContVoto.....	82
Figura 65: Tela de configuração de conexão do sistema ContVoto - Cliente.	83
Figura 66: Tela de identificação do sistema ContVoto - Cliente.....	83
Figura 67: Tela de votação para o cargo de Reitor do sistema ContVoto - Cliente...	84
Figura 68: Tela de votação para o cargo de Diretor-Geral do sistema ContVoto - Cliente.	84

SUMÁRIO

1	INTRODUÇÃO.....	16
1.1	Justificativa	16
1.2	Objetivos	17
1.2.1	Objetivo geral	17
1.2.2	Objetivo específico.....	17
2	REFERENCIAL TEÓRICO.....	18
2.1	Tecnologia no processo eleitoral brasileiro	18
2.1.1	História	18
2.1.1.1	Iniciativa	18
2.1.1.2	Cadastro eleitoral informatizado.....	18
2.1.1.3	Totalização eletrônica de votos.....	19
2.1.1.4	Urna eletrônica.....	19
2.2	Engenharia de <i>Software</i>	20
2.2.1	Processo de software	21
2.2.1.1	Modelos de processo de software.....	22
2.2.2	Engenharia de requisitos.....	29
2.2.2.1	Requisitos funcionais	29
2.2.2.2	Requisitos não funcionais	30
2.2.2.3	Documento de requisitos.....	30
2.2.3	Modelagem de <i>software</i>	31
2.2.3.1	Paradigma de orientação a objetos.....	32
2.2.3.2	Linguagem de Modelagem Unificada (UML)	32
2.2.4	Modelagem de banco de dados	44
2.3	Programação de computadores	46
2.3.1	JAVA	47
2.3.1.1	Atributo	48
2.3.1.2	Método	49
2.3.1.3	Classe.....	49
2.3.1.4	Objeto.....	50
2.3.1.5	Relacionamento.....	51
2.3.1.6	Modificadores de acesso	51
2.3.1.7	Swing.....	53
2.3.1.8	Socket	54
2.3.1.9	JDBC.....	55
2.3.2	SQL.....	55
2.3.2.1	<i>Create</i>	56
2.3.2.2	<i>Insert</i>	56
2.3.2.3	<i>Update</i>	56
2.3.2.4	<i>Select</i>	57
2.3.2.5	<i>Delete</i>	57
2.4	Processo eleitoral no IF Sertão Pernambucano.....	58
2.4.1	Objetivo do processo	58
2.4.2	Cargos eletivos	59
2.4.2.1	Reitor	59
2.4.2.2	Diretor-Geral	59
2.4.3	Local de realização e pessoas envolvidas.....	59
2.4.3.1	Comissões eleitorais.....	59
2.4.3.2	Candidatos.....	60
2.4.3.3	Eleitores.....	60
2.4.4	Consulta.....	61

2.4.4.1	Votação	61
2.4.4.2	Segurança	61
2.4.4.3	Apuração e resultado	61
3	METODOLOGIA.....	64
3.1	Documento de requisitos	65
3.1.1	Requisitos.....	65
3.1.2	Organização do sistema	66
3.2	Projeto.....	67
3.2.1	Casos de uso	67
3.2.2	Interfaces Gráficas	68
3.2.3	Detalhamento dos casos de uso	70
3.2.4	Classes	70
3.2.5	Modelo relacional	71
3.2.6	Diagrama de sequência.....	72
3.2.7	Diagrama de máquina de estados	73
3.3	Implementação	74
3.3.1	<i>NetBeans</i>	75
3.3.2	<i>MySql Workbench</i>	76
4	RESULTADOS	78
4.1	Servidor	78
4.1.1	<i>Login</i>	78
4.1.2	Tela Principal	79
4.1.3	Cadastro de eleitor	79
4.1.4	Gerenciamento de candidatos	80
4.1.5	Iniciar uma eleição	81
4.1.6	Resultado de eleições	81
4.2	Cliente	82
4.2.1	Configuração de conexão com o servidor	82
4.2.2	Autoidentificação do eleitor	83
4.2.3	Escolha de candidatos e registro dos votos	84
5	CONSIDERAÇÕES FINAIS.....	85
5.1	Trabalhos futuros.....	86
	REFERÊNCIAS.....	87
	APÊNDICES.....	89

1 INTRODUÇÃO

A humanidade passa por um processo contínuo de evolução, evolução essa existente em todos os setores, sejam eles públicos ou privados. Nesse processo os sistemas informatizados são extremamente utilizados. Segundo Norton (1996, p. 418), “Um sistema é um conjunto de regras e procedimentos para o fornecimento preciso e oportuno de informação”.

Através desses sistemas e da Engenharia de *Software* para o desenvolvimento dos mesmos, o Brasil passou de uma apuração de votos de forma manual, cansativa e demorada para um resultado de uma eleição de forma automatizada, simples e rápida (TRIBUNAL REGIONAL ELEITORAL/RS, 2006). Engenharia de *Software* é uma disciplina de engenharia cujo foco está em todos os aspectos da produção de *software*, desde os estágios iniciais da especificação do sistema até sua manutenção (SOMMERVILLE, 2011).

Nesse trabalho é projetado um sistema de contabilização de votos para as eleições da Diretoria-Geral do IF Sertão Pernambucano, sendo implementado através da linguagem de programação JAVA, no paradigma de programação orientada a objetos e gerenciado através do método ágil XP. Como fazer uso da Engenharia de *Software*, das ferramentas e técnicas conhecidas e disponíveis, das linguagens de programação de computadores, ou seja, como fazer uso da Tecnologia da Informação para automatizar o processo eleitoral naquela instituição?

1.1 Justificativa

A disponibilização grátis de diversas técnicas e ferramentas para o desenvolvimento de sistemas, propondo a possibilidade de criar um *software* útil com um baixo custo. Além disso, o processo eleitoral atual no IF Sertão Pernambucano – Campus Floresta é realizado de forma manual, indo de encontro com o avanço tecnológico nos dias de hoje. Por fim, a necessidade particular de fazer e deixar algo útil, facilitando assim a vida de terceiros.

1.2 Objetivos

1.2.1 Objetivo geral

Desenvolver um sistema computacional para automação do processo eleitoral do IF Sertão Pernambucano, facilitando assim os trabalhos realizados por todos os envolvidos no evento.

1.2.2 Objetivo específico

- Analisar e conhecer as regras existentes nas legislações que regem o processo eleitoral do IF Sertão Pernambucano e, utilizando a Engenharia de *Software*, transformá-las em funcionalidades do sistema.
- Mostrar a importância do uso da Tecnologia da Informação para a simplificação de tarefas administrativas.
- Documentar e especificar as fases existentes no decorrer de um processo de desenvolvimento de *software*.
- Entregar um sistema funcional com seu projeto completo e código fonte comentado.

2 REFERENCIAL TEÓRICO

2.1 Tecnologia no processo eleitoral brasileiro

2.1.1 História

Até chegar a utilizar efetivamente a tecnologia no processo eleitoral, o Brasil contou com vários episódios ocorridos entre os anos 1930 e 1996, episódios esses que acabaram resultando no desenvolvimento da urna eletrônica, cujo papel fundamental é registrar e contabilizar votos, como também apresentar o resultado de uma determinada eleição.

2.1.1.1 Iniciativa

A tecnologia começou a ser debatida no TSE a partir do surgimento do Código Eleitoral de 1932, cujo texto previa a utilização de uma “máquina de votar” para o auxílio no processo de votação das eleições no Brasil, fazendo com que a Corte, no ano de 1937, criasse uma comissão para avaliar e elaborar parecer acerca de protótipos apresentados em sessão plenária, sendo no ato apreciado três projetos que de certo modo agradou os ministros do Tribunal. No entanto a tentativa de encontrar uma máquina que atendesse as expectativas da legislação foi interrompida pela decretação do Estado Novo, ditadura implantada pelo governo Getúlio Vargas que acabou extinguindo a Justiça Eleitoral (TRIBUNAL REGIONAL ELEITORAL/RS, 2006).

2.1.1.2 Cadastro eleitoral informatizado

Por muito tempo as iniciativas populares relativas à máquina de votar sumiram. No entanto, na década de 70, volta a ser discutida dentro e fora da Justiça Eleitoral a utilização de recursos tecnológicos no cadastramento de eleitores e contabilização de votos. Nesse sentido foi realizado o que Paulo César Camarão, ex-secretário de informática do TSE, classificou de o primeiro passo oficial do processo que levou à construção da urna eletrônica, o cadastramento eleitoral

eletrônico. Desenvolvido com base na lei nº 6.996/82, cuja mesma autorizava a utilização de recursos de informática no processamento de dados nas eleições, o cadastro eliminou uma grande quantidade de fichas mantidas nos TREs, sendo as mesmas substituídas por um banco de dados computadorizado (TRIBUNAL REGINAL ELEITORAL/RS, 2006). O sistema que inicialmente foi implantado no TRE do Rio Grande do Sul foi apresentado ao TSE e aprovado pelo mesmo, marcando aí o início efetivamente da utilização de tecnologias nas eleições brasileiras.

2.1.1.3 Totalização eletrônica de votos

Valendo-se também da Lei nº 6.996/82, o TRE do Rio de Janeiro contratou a empresa Proconsult com o objetivo de contabilizar eletronicamente os votos das eleições de 1982. No ato, parte da imprensa pôs a apuração eletrônica em suspeita, pois os números divulgados pela Proconsult não batiam com pesquisas realizadas no dia da votação. O problema na contabilização ocorreu pelo fato do processo ainda envolver serviços manuais e cédulas de papel. Após rumores de fraude envolvendo a utilização da tecnologia, a Polícia Federal concluiu uma investigação descartando tal possibilidade (TRIBUNAL REGINAL ELEITORAL/RS, 2006).

A fim de continuar com a tecnologia, o TRE de Santa Catarina, no ano de 1990, utilizou um microcomputador em cada zona eleitoral para assim melhorar o processo de totalização de votos. Um ano mais tarde repetiu o mesmo processo no plebiscito de emancipação política do Distrito de Cocal do Sul, sendo tal ato registrado pelo o mesmo tribunal como sendo a primeira votação totalmente eletrônica na América Latina (TRIBUNAL REGINAL ELEITORAL/RS, 2006).

2.1.1.4 Urna eletrônica

Com o aumento da utilização da informática e popularização de suas ferramentas, era, no meio da Justiça Eleitoral, cada vez mais crescente o desejo de automatizar com segurança e robustez as eleições brasileiras cujo eleitorado encontrava-se absurdamente aumentando. Com tal objetivo foram criadas duas comissões com funções distintas, mas com o mesmo objetivo, desenvolver a tão sonhada máquina de votar citada no Código Eleitoral de 1932. Da primeira, presidida

pelo o ministro Ilmar Galvão, composta por juristas e técnicos de informática de vários tribunais, deu origem a um Termo de Referência, documento que reunia, além de outros pontos, várias funcionalidades desejadas no futuro equipamento. Elaborado em 1995, o documento ainda previa a conclusão e utilização da urna eletrônica já nas eleições do ano seguinte. A segunda comissão foi composta por profissionais de informática de vários órgãos públicos, dentre eles o Instituto Tecnológico da Aeronáutica – ITA, tendo a mesma como papel projetar exatamente as especificações técnicas da máquina com base no documento da comissão anterior, nascendo aí a Urna Eletrônica. Após concluir todo o projeto o Tribunal Superior Eleitoral declarou que a eleição de 1998 seria a última a utilizar cédulas de papel (TRIBUNAL REGIONAL ELEITORAL/RS, 2006).

2.2 Engenharia de *Software*

Nos dias de hoje encontramos *software* praticamente em todos os lugares, seja nos painéis dos aeroportos, no caixa eletrônico de um banco ou na televisão que assistimos aos nossos programas favoritos. O mesmo pode ser criado de forma pessoal, ou seja, um programador o cria para dar suporte as suas próprias tarefas diárias sem documentar a sua estrutura, impossibilitando que terceiros o utilize por não existir um manual que o auxilie como também dificulta que outros programadores façam alterações no mesmo com a intenção de deixá-lo mais funcional e útil, pois o sistema não foi projetado utilizando as técnicas e ferramentas adequadas, tornando-o inútil, sendo necessário criar um novo programa do zero. A Engenharia de *Software* é utilizada para o desenvolvimento profissional de sistemas computacionais.

Para Sommerville (2011, p. 3):

A engenharia de software tem por objetivo apoiar o desenvolvimento profissional de software, mais do que a programação individual. Ela inclui técnicas que apoiam especificação, projeto e evolução de programas, que normalmente não são relevantes para o desenvolvimento de software pessoal.

A engenharia de *software* possibilita que os engenheiros de *software* – pessoas responsáveis por projetar o sistema – transcrevam para o projeto o que se quer de um programa, de modo que os programadores – pessoas responsáveis por utilizar uma determinada linguagem de programação para codificar o sistema, escrevam-no exatamente com as funcionalidades e qualidade descritas. O projetista é a ponte entre a vontade dos utilizadores do futuro sistema e a equipe técnica de desenvolvimento, planejando dentre outras atividades o custo com os trabalhos a serem realizados. Para efetivamente levar as dificuldades do mundo real para dentro dos sistemas, ou seja, fazer com que o sistema resolva os problemas detectados, a engenharia de *software* dispõe de técnicas e ferramentas que buscam desenvolver e entregar um produto com qualidade, menor gasto possível e que possibilite uma mudança e conseqüente evolução sem prejuízos maiores.

A comunicação entre os envolvidos no projeto ocorre através da escrita, como também de desenhos lógicos chamados de diagramas, cuja função é modelar as características e funcionalidades do sistema de maneira gráfica, sendo necessária para descrever algumas figuras e requisitos (BEZERRA, 2007).

2.2.1 Processo de software

Processo de *software* é uma metodologia, ou seja, como as atividades envolvidas no decorrer da elaboração de um sistema são abordadas. Ele define como as ferramentas e técnicas existentes serão utilizadas na busca da qualidade do *software* em desenvolvimento. Na engenharia de *software* existem diversos modelos de processo, podendo ser escolhido o mais adequado para o tipo de projeto a ser desenvolvido ou até mesmo combinar mais de um deles, focados em fases distintas mas com o mesmo objetivo, o de produzir um *software* de qualidade.

Os processos de *software* são basicamente caracterizados como: dirigidos a plano e processos ágeis. Aqueles têm todas as suas atividades planejadas com antecedência para serem utilizadas do início ao fim do projeto, tornando rígida a ocorrência de eventuais mudanças no decorrer dos trabalhos, enquanto que estes possuem seus planejamentos gradativos, possibilitando uma eventual alteração para refletir uma necessidade e desejo do cliente em qualquer fase do projeto (SOMMERVILLE, 2011).

2.2.1.1 Modelos de processo de software

Um modelo de processo de software é uma representação simplificada de um processo de software, visto cada um deles conter perspectivas particulares de um processo e fornecer informações parciais do mesmo (SOMMERVILLE, 2011).

Independente do modelo de processo escolhido para o desenvolvimento de um software, todos eles incluem quatro atividades fundamentais:

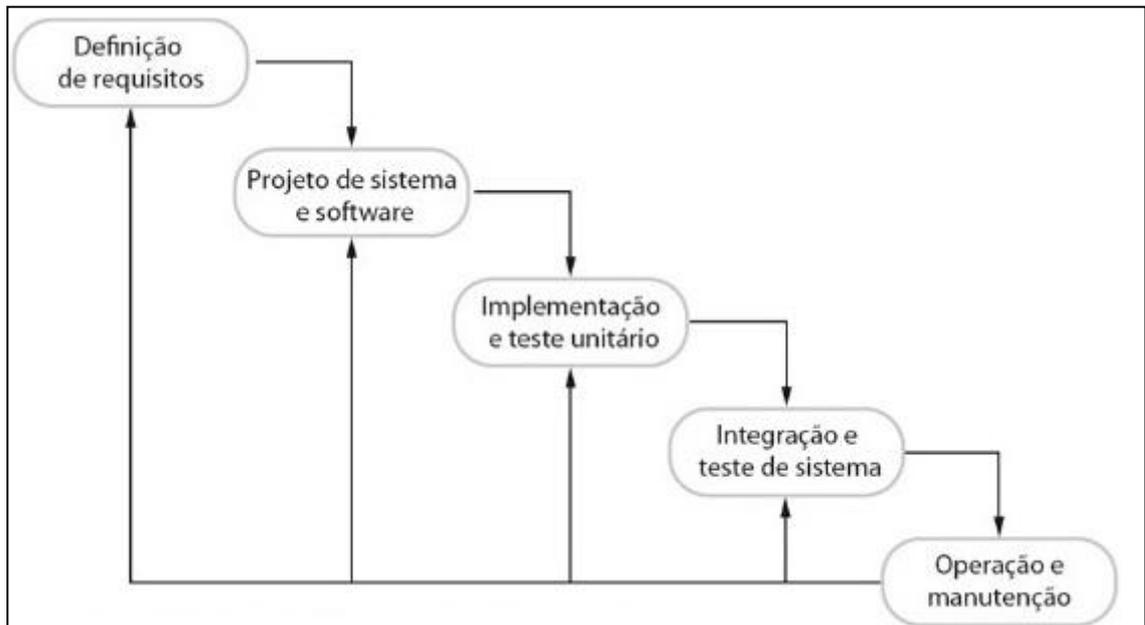
1. **Especificação de software:** Definição das funcionalidades do *software* com a especificação das restrições das mesmas;
2. **Projeto e implementação de software:** Produção do *software* respeitando às especificações definidas;
3. **Validação de software:** Validação do *software* para garantir que suas funcionalidades estejam coerentes com as especificações;
4. **Evolução de software:** Evolução para garantir que alterações de requisitos e funcionalidades sejam realizadas no *software*.

Como o processo aponta de maneira organizada o que deve ser feito, em que momento, quem deve fazer e quais ferramentas serão utilizadas, é de extrema importância definir um ou mais deles para assim manter a gerência do projeto mais simples e fácil de ser realizada. Modelo em cascata, desenvolvimento incremental e *extreme programming* (XP) são exemplos dos mais variados modelos existentes.

- **Modelo em cascata**

Pertencente ao grupo dos processos orientados a plano, o modelo em cascata conta com um encadeamento de fases, ligando uma a outra, conforme mostra a Figura 1.

Figura 1: Modelo em cascata.



Fonte: Sommerville (2011, p. 20).

Essas fases definem o ciclo de vida do *software*, sendo que cada uma delas produz documentos que servirão para alimentar a fase seguinte, não podendo um estágio ser iniciado enquanto o anterior não for concluído.

Parecidas com as atividades fundamentais do desenvolvimento de software, as fases desse modelo possuem as seguintes características:

1. **Definição de requisitos:** Por meio de consulta aos usuários do futuro sistema são elaboradas as especificações do mesmo;
2. **Projeto de sistema e software:** Alocação dos requisitos para a parte física e lógica do sistema, sendo elaborada uma arquitetura completa;
3. **Implementação e teste unitário:** O projeto é desenvolvido em partes chamadas de programas, sendo cada um deles testados a fim de constatar a sua coerência com o que foi planejado nas especificações;
4. **Integração e teste de sistema:** As unidades são integradas e testadas com o objetivo de verificar se as mesmas estão funcionando relacionadamente conforme o que foi definido com o cliente, sendo o sistema entregue a este.
5. **Operação e manutenção:** O sistema é instalado e utilizado, identificando eventuais erros que passaram por despercebidos, momento em que surgem novos requisitos.

Esse modelo é mais utilizado em projetos cujos requisitos do sistema sejam bem definidos e que alteração nos mesmos seja algo improvável de ocorrer no decorrer dos trabalhos, pois tudo é planejado no estágio inicial, tendo uma divisão inflexível em relação a mudanças.

- **Desenvolvimento incremental**

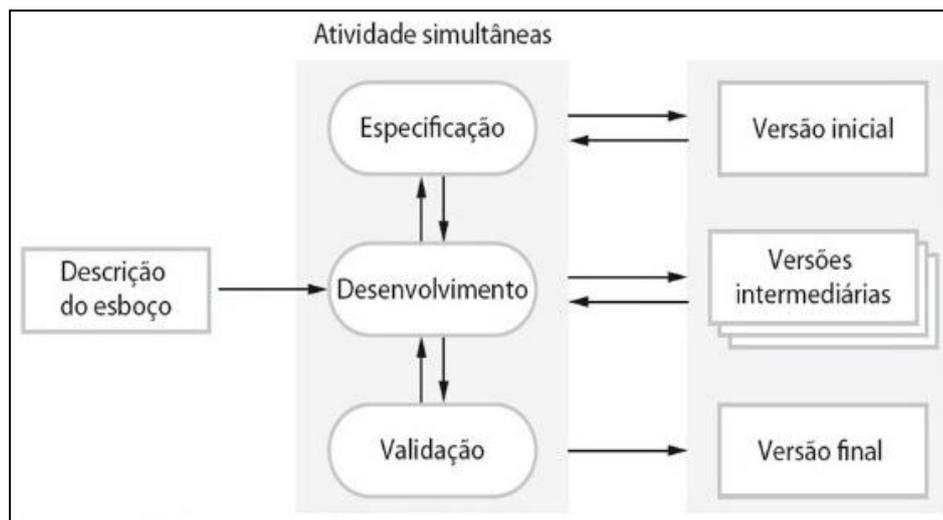
O desenvolvimento incremental é uma parte fundamental nos processos ágeis, por isso esse modelo pode ser caracterizado tanto como tal como com um modelo dirigido a planos, ou até mesmo combiná-los.

Para Sommerville (2011, p. 3):

Em uma abordagem dirigida a planos, os incrementos do sistema são identificados previamente; se uma abordagem ágil for adotada, os incrementos iniciais são identificados, mas o desenvolvimento de incrementos posteriores depende do progresso e das prioridades do cliente.

As atividades do desenvolvimento incremental são intercaladas, contando com um rápido *feedback* entre elas, conforme mostra a Figura 2.

Figura 2: Desenvolvimento incremental.



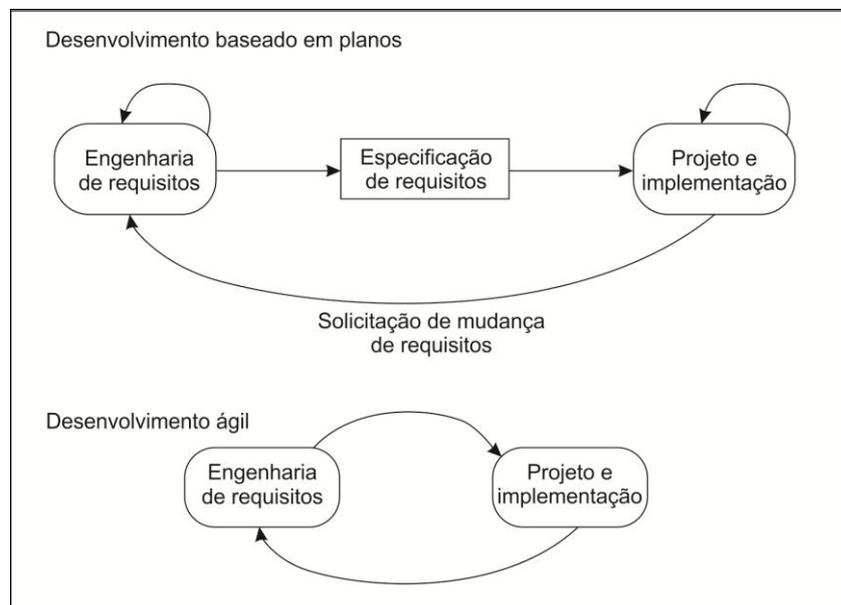
Fonte: Sommerville (2011, p. 22).

- **Extreme Programming (XP)**

Os modelos de processos dirigidos a planos são bastante eficientes no desenvolvimento de sistemas críticos cuja prioridade seja a segurança, como um software de controle de aeronaves, pois os mesmos têm suas atividades voltadas para documentação e gerenciam uma grande quantidade de equipes envolvidas. No entanto, quando o software não tem essas características, o processo de desenvolvimento fica bastante burocrático e caro, visto os mesmos não serem feitos para lidar com mudanças corriqueiras, principalmente nos desejos do cliente.

Diferentes desses, os processos ágeis de desenvolvimento têm o foco no programa funcional, ou seja, a entrega de programa funcionando ao cliente ao invés de vasta documentação que o mesmo sequer entende. O contato com ele ocorre periodicamente a fim de manter sempre a funcionalidade do software de acordo com seus desejos, mesmo que eles venham a mudar. A comparação entre os dois tipos de processos é mostrada na Figura 3.

Figura 3: Especificações dirigidas a planos, e ágeis.



Fonte: Sommerville (2011, p. 43).

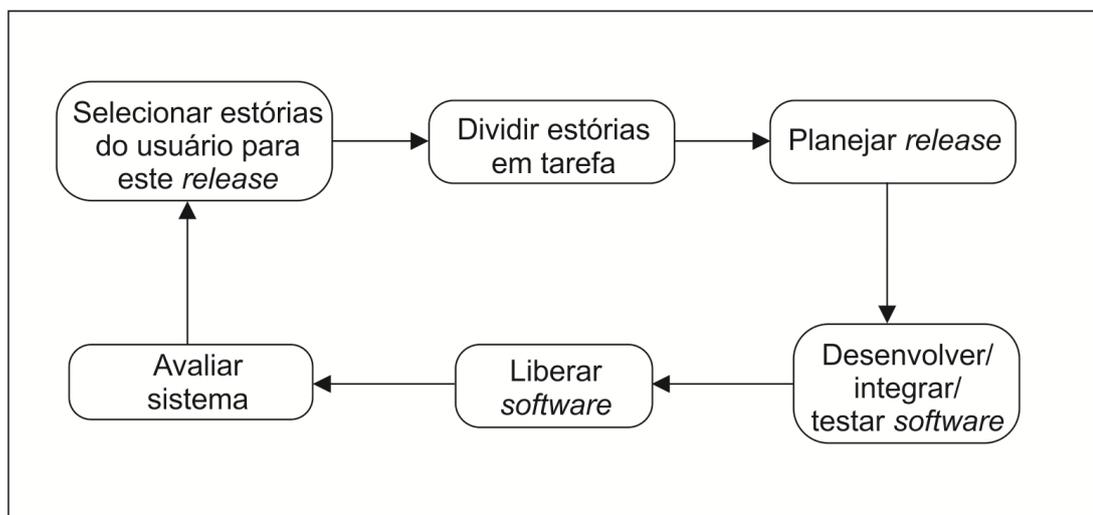
Extreme Programming (XP) é um dos mais variados modelos ágeis existentes. Aqui os requisitos do cliente são chamados de estórias do cliente, sendo os mesmos divididos em tarefas para assim serem executadas e codificadas.

Trabalhando em pares, os programadores desenvolvem testes para o sistema antes de codificar cada tarefa.

Tendo um foco maior na programação, o XP prevê a presença periódica do cliente no decorrer do projeto, considerando-o até mesmo como parte da equipe, pois em vários momentos é apresentado para o mesmo um conjunto de funcionalidades do sistema já codificadas e em execução (*release*), para que assim ele avalie e opine se está de acordo ou se necessita de alguma mudança.

Cada *release* leva cerca de duas semanas para ser desenvolvido e apresentado para avaliação do cliente, tendo seu ciclo de vida conforme mostra a Figura 4.

Figura 4: Ciclo de vida de um release em Extreme Programming.



Fonte: Sommerville (2011, p. 44).

Antes de iniciar a programação de um *release*, é definido um conjunto de testes unitários que servirão de base para validar as funcionalidades do mesmo, acabando também por orientar os serviços de codificação realizados pela equipe de programadores.

Além dos testes unitários (pequenos testes no ato da programação), é planejado um plano prevendo os testes de usuário, cujo objetivo é validar as funcionalidades do sistema com suas partes integradas, ficando tal teste, assim como o *release*, submetido à aprovação do proprietário do sistema.

- **Scrum**

Pertencente ao conjunto de processos ágeis, esse método é utilizado para gerenciar equipes no decorrer do desenvolvimento de um *software*, propondo diversos recursos com objetivos distintos, mas todos eles alinhados para se chegar a uma finalidade (entrega de um produto que esteja condizente com o planejado).

Para Pressman (2011, p. 95):

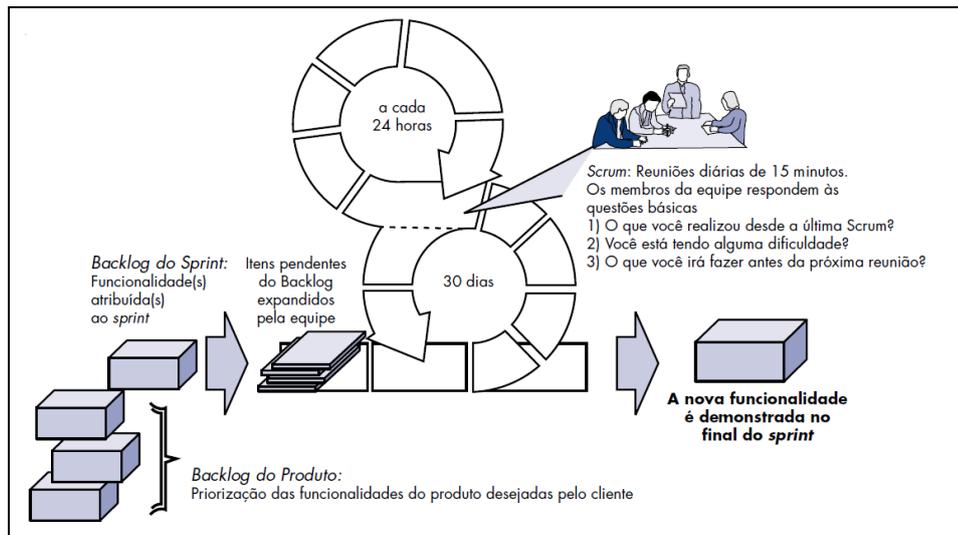
Os princípios do Scrum são consistentes com o manifesto ágil e são usados para orientar as atividades de desenvolvimento dentro de um processo que incorpora as seguintes atividades estruturais: requisitos, análise, projeto, evolução e entrega.

Esse processo contém diversos padrões para definir as funcionalidades, tarefas a serem executadas, e pessoal envolvido no desenvolvimento:

1. **Backlog:** Lista dos requisitos do sistema a serem implementados, podendo a mesma ser alterada no decorrer do processo;
2. **Sprint:** Trabalhos definidos com base na lista de requisitos que deverão ser realizados em um curto espaço de tempo, geralmente 30 dias. Diz respeito a uma parte do sistema que será codificada;
3. **Product Owner:** Proprietário do sistema ou pessoa por ele indicada. É responsável por indicar os *backlogs* desejados;
4. **Dev Team:** Time de desenvolvimento responsável por planejar e codificar as especificações das *sprints*, gerando partes funcionais do sistema;
5. **Scrum Master:** Responsável por garantir que as práticas do *Scrum* sejam aplicadas durante todo o processo de desenvolvimento. É ele quem orienta o proprietário e equipe de desenvolvimento em seus trabalhos.

Ainda são previstas reuniões diárias de curta duração, sendo que tudo isso ocorre conforme o fluxo mostrado na Figura 5.

Figura 5: Fluxo das atividades no Scrum.



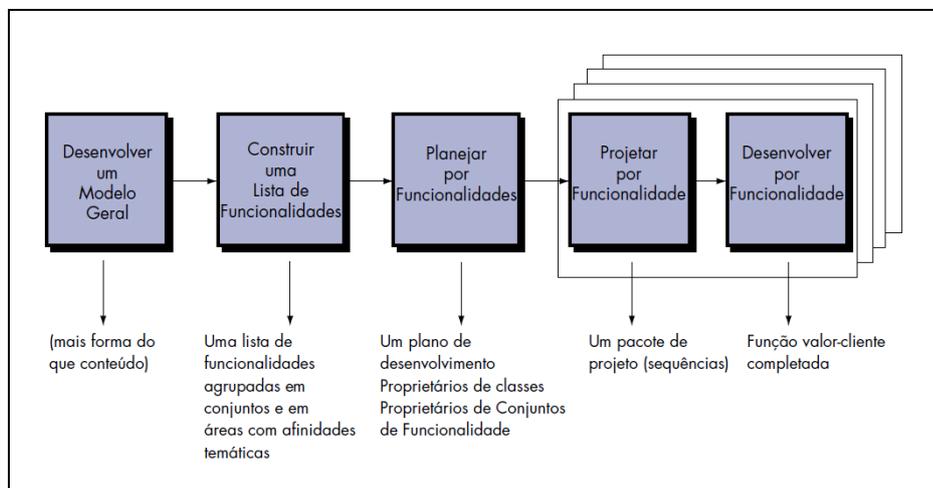
Fonte: Pressman (2011, p. 96).

- **Desenvolvimento Dirigido a Funcionalidades (FDD)**

Assim como os demais processos ágeis, o FDD tem seu foco nos desejos do cliente, no entanto o mesmo é voltado para a funcionalidade do software, como também na integração de seus incrementos, inspeção de código e auditoria para garantir a qualidade do produto.

Para Pressman (2011, p. 98), no contexto do FDD, funcionalidade “é uma função valorizada pelo cliente passível de ser implementada em duas semanas ou menos”. Essas funcionalidades são planejadas e projetadas conforme a Figura 6.

Figura 6: Fluxo das atividades do FDD.



Fonte: Pressman (2011, p. 96).

2.2.2 Engenharia de requisitos

Utilizada para identificar e definir as funcionalidades pretendidas de um sistema, a engenharia de requisitos é bastante eficiente para transformar a linguagem simples do cliente em uma linguagem técnica, possibilitando que o engenheiro de requisitos possa entender os desejos desse cliente acerca do sistema e transformar esses desejos em uma linguagem padrão e unificada de tal forma que todos os integrantes da equipe de desenvolvimento entendam o que o futuro *software* deve ter de funcionalidades.

Para Sommerville (2011, p. 57):

Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento. Esses requisitos refletem as necessidades dos clientes para um sistema que serve a uma finalidade determinada, como controlar um dispositivo, colocar um pedido ou encontrar informações.

Os requisitos de *software* são classificados como funcionais e não funcionais, sendo que aqueles são serviços e funcionalidades que o sistema deve oferecer, enquanto que estes são restrições aos serviços oferecidos por esse sistema.

2.2.2.1 Requisitos funcionais

Tendo suas características dependentes do tipo de *software* a ser desenvolvido, os requisitos funcionais descrevem o que o sistema deve fazer, quais ações e decisões o mesmo deve tomar para refletir os comandos vindos do usuário.

Para um sistema de contabilização de votos pode-se definir os seguintes requisitos funcionais como exemplo:

1. O sistema deve possibilitar que o usuário cadastre candidatos na base de dados;
2. O sistema deve possibilitar que o usuário emita relatórios com resultado de eleições anteriormente realizadas;
3. O sistema deve possibilitar que o administrador do mesmo encerre uma eleição em aberto;

4. O sistema deve emitir um relatório com o resultado da eleição assim que a mesma for encerrada;

2.2.2.2 Requisitos não funcionais

São requisitos que não estão relacionados com a funcionalidade do sistema, com ações que o mesmo executa. Dizem respeito ao desempenho, tempo de resposta, proteção ou disponibilidade.

Para o mesmo sistema de contabilização de votos pode-se definir os seguintes requisitos não funcionais como exemplo:

1. O sistema deve utilizar o gerenciador de banco de dados MySQL;
2. O sistema deve conter dois módulos (um de administração e outro de votação);
3. O sistema deve ser codificado com a linguagem de programação JAVA pelo o fato da mesma possibilitar que o *software* rode em vários sistemas operacionais diferentes;

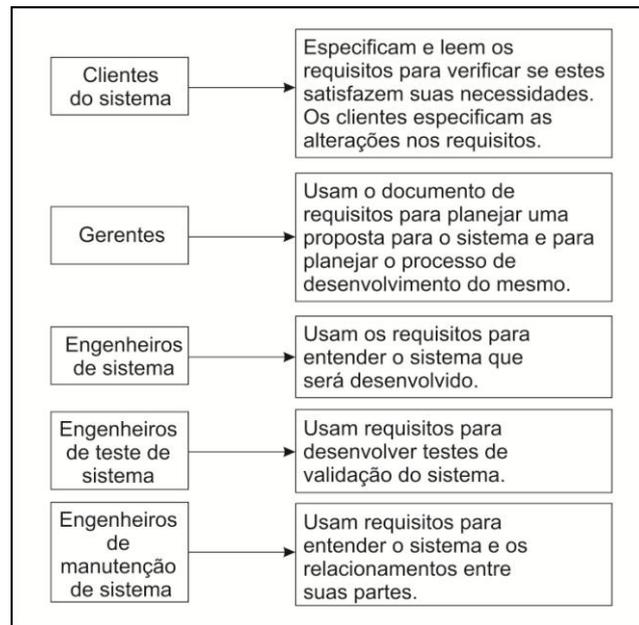
2.2.2.3 Documento de requisitos

Após o engenheiro fazer um levantamento dos requisitos do sistema, seja através de entrevistas com clientes, pessoas diretamente envolvidas com o *software*, programas similares ou consulta a documentos, é elaborado um documento com as especificações dos mesmos, o chamado documento de requisitos.

Para Sommerville (2011, p. 63), o documento de requisitos é uma declaração oficial de o que os desenvolvedores do sistema devem implementar.

O documento de requisitos orienta os envolvidos no projeto, mantendo uma comunicação entre os usuários da equipe, conforme mostra a Figura 7.

Figura 7: Usuários de um documento de requisitos.



Fonte: Adaptado de Sommerville, 2011.

2.2.3 Modelagem de *software*

O desenvolvimento de um *software*, comparado com uma construção civil, pode resultar na elaboração de algo que não foi planejado nem requisitado, fazendo com que uma parte ou até mesmo a totalidade do que já esteja pronto seja descartado, encarecendo e atrasando o projeto.

Para amenizar a ocorrência desses fatos, a engenharia civil conta com vários desenhos que representam a construção antes mesmo dela ser realizada, tais como planta baixa, projeto hidráulico e elétrico, detalhando de forma gráfica o que deve ser feito, facilitando assim a comunicação e conseqüentemente o entendimento acerca do processo de construção.

Semelhante à engenharia civil, a engenharia de *software* possibilita que os engenheiros desenvolvam diversos modelos de um mesmo sistema antes do mesmo ser implementado. Esses modelos são feitos utilizando diagramas e textos, sendo que cada tipo trata o mesmo sistema em uma perspectiva diferente, tais como: interação, comportamento, estrutura etc.

Esses diagramas são oriundos da UML (linguagem de modelagem unificada, do inglês *Unified Modeling Language*), utilizada para modelagem de sistemas

orientados a objetos, paradigma presente na maioria dos projetos de *software* nos dias atuais.

2.2.3.1 Paradigma de orientação a objetos

Em nosso cotidiano nos deparamos com diversos problemas que podem ser abordados de diferentes formas, mas que todas elas resultarão em algo igual, uma proposta de resolução desses. Paradigma, em engenharia de *software*, é justamente isso, a forma como os problemas a serem solucionados pelo sistema projetado são vistos e abordados, resultando em uma proposta de solução dos mesmos.

Dentre os diversos paradigmas existentes, os mais utilizados são o estruturado e o orientado a objetos, sendo que enquanto aquele trata os componentes do sistema como dados e processos, este os aborda como objetos independentes e encarregados de executar uma tarefa específica (BEZERRA, 2007).

Posterior ao estruturado, o paradigma orientado a objetos, de uma forma mais fácil de entender, aborda um *software* como o funcionamento de uma empresa, local composto por vários setores com funções diferentes, mas que trabalham em conjunto com um objetivo em comum. Em uma comparação, cada setor da empresa (recursos humanos, contabilidade, compras etc.) seria um objeto, elemento responsável por executar e solicitar que outro objeto realize uma tarefa, mantendo assim o funcionamento completo do sistema.

Esse paradigma foi criado com o objetivo de aproximar a modelagem de *software* ao que as pessoas fazem nos seus cotidianos, deixando o processo de transformação das funcionalidades desejadas do sistema em uma linguagem técnica um tanto que mais fácil de ser realizado.

2.2.3.2 Linguagem de Modelagem Unificada (UML)

A UML é uma linguagem utilizada para modelagem de sistema orientado a objetos. No processo de modelagem de *software*, um mesmo sistema pode ser representado de diversas perspectivas, ou seja, de visões diferentes. A UML

descreve essas perspectivas de forma gráfica, facilitando o entendimento do projeto em desenvolvimento.

Para Bezerra (2007, p. 16), a linguagem descreve o sistema nas seguintes visões:

Visão de Caso de Uso: *descreve o sistema de um ponto de vista externo com um conjunto de interações entre o sistema e os agentes externos ao sistema. Esta visão é criada inicialmente e direciona o desenvolvimento das outras visões do sistema.*

Visão de projeto: *ênfatisa as características do sistema que dão suporte, tanto estrutural como comportamental, às funcionalidades externamente visíveis ao sistema.*

Visão de implementação: *abrange o gerenciamento de versões do sistema, construídas pelo o agrupamento de módulos (componentes) e subsistemas.*

Visão de implantação: *corresponde à distribuição física do sistema em seus subsistemas e à conexão entre essas partes.*

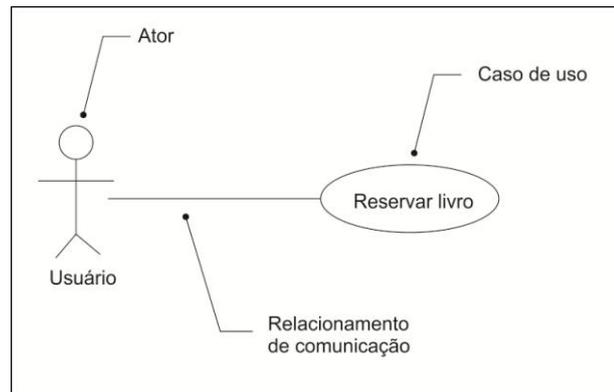
Visão de processo: *essa visão ênfatisa as características de correspondência (paralelismo), sincronização e desempenho do sistema.*

Para modelagem gráfica, a UML conta com diferentes diagramas que juntos possibilitam o fácil entendimento acerca do que precisa ser realizado, tanto no momento da construção do *software* como na manutenção do mesmo. Cada diagrama tem o seu foco voltado para uma visão específica do sistema, possibilitando que os engenheiros de *software* possam gerir a complexidade do projeto. Existe um total de 13 diagramas, no entanto não necessariamente todos são utilizados em um determinado projeto, podendo uma parte deles representar um sistema de forma completa.

- **Diagrama de casos de uso**

Utilizado para representar a interação de elementos externos com as funcionalidades do sistema, o diagrama de casos de uso é um dos primeiros a serem construídos em um projeto de *software*. Seu desenvolvimento é realizado com base nos requisitos, ou seja, nas funcionalidades pretendidas do sistema. O mesmo é composto por três elementos (caso de uso, ator e relacionamento), conforme mostra a Figura 8.

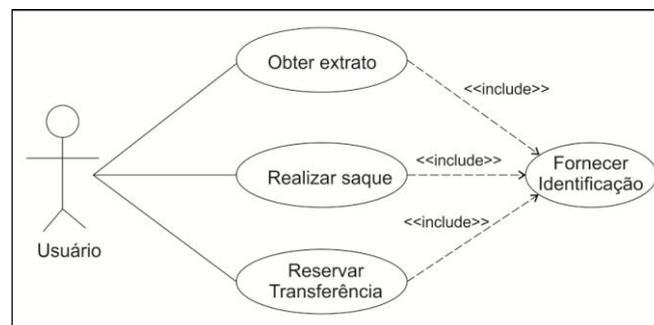
Figura 8: Diagrama de caso de uso (simples).



Fonte: Adaptado de Bezerra, 2007.

O ator que interage com uma funcionalidade (caso de uso) nem sempre é uma pessoa, podendo o mesmo ser qualquer coisa, tais como um *hardware* ou até mesmo outro *software*. Uma única comunicação pode envolver mais de um caso de uso, conforme mostra a Figura 9.

Figura 9: Diagrama de caso de uso (include).

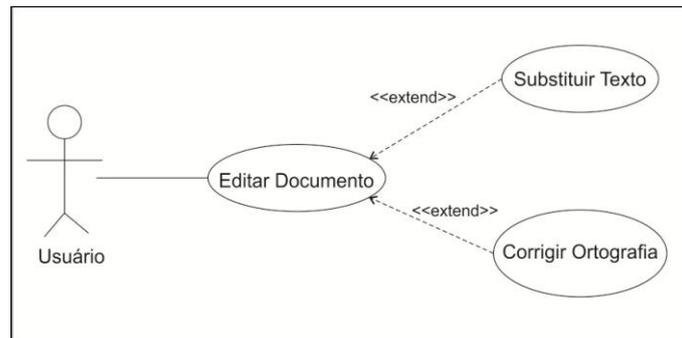


Fonte: Adaptado de Bezerra, 2007.

As funcionalidades com as quais o usuário está diretamente se comunicando incluem outra, a "Fornecer Identificação". Ou seja, quando o ator interagir com "Obter Extrato" estará também interagindo com "Fornecer Identificação", pois esta está incluída naquela.

O diagrama de casos de uso ainda conta com mais dois tipos de relacionamentos, o de extensão e generalização. Extensão, assim como inclusão, ocorre entre casos de uso, conforme mostra a Figura 10.

Figura 10: Diagrama de caso de uso (extend).

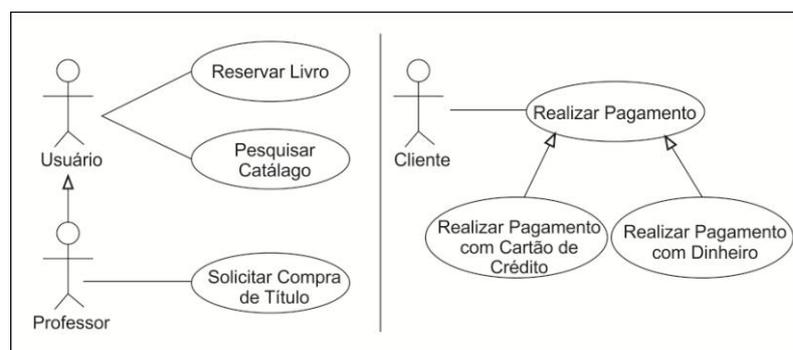


Fonte: Adaptado de Bezerra, 2007.

Enquanto que na inclusão o caso de uso incluído é automaticamente executado, na extensão o mesmo fica somente disponível para tal. No exemplo da Figura 8, quando o usuário editar um documento, nessa mesma ação ele pode substituir um texto ou corrigir a ortografia do mesmo, pois essas funcionalidades são extensões da outra, ou seja, são complementos.

Enquanto que a inclusão e extensão ocorrem somente entre casos de uso, a generalização pode ocorrer tanto entre esses como entre atores, conforme mostra a Figura 11.

Figura 11: Diagrama de caso de uso (generalização).



Fonte: Adaptado de Bezerra, 2007.

Quando essa relação ocorre entre os casos de uso, a mesma indica que uma determinada funcionalidade mais genérica é especializada por outras mais específicas, assim como ocorre entre os casos de usos “Realizar Pagamento”, “Realizar Pagamento com Cartão de Crédito” e “Realizar Pagamento com Dinheiro”. Enquanto que ela ocorre entre atores, como em “Professor” e “Usuário”, indica que aquele interage com qualquer caso de uso com que este venha interagir.

- **Descrição funcional de casos de uso**

Como um caso de uso define uma funcionalidade do sistema, visto o mesmo representar um requisito funcional, ele necessita de um nível mais aprofundado de detalhes para que o mesmo seja implementado conforme o desejo do projetista. Situações como: o que precisa está previamente pronto e disponível para que uma funcionalidade se inicie; sequência de passos a ser seguida; caminho a percorrer quando um determinado passo não ocorrer bem; o que muda após a conclusão de uma determinada funcionalidade; são uns dos assuntos a serem explicitados no projeto para um melhor entendimento do sistema em desenvolvimento.

Para desenvolver essa descrição, a UML dispõe de um fluxo de eventos no qual, através de tabelas e textos, ajuda a detalhar o processo de interação entre usuário e sistema para cada casa de uso, conforme mostra a Figura 12.

Figura 12: Descrição detalhada de caso de uso.

Caso de uso: Caso de uso que está sendo detalhado
Atores: Atores que interagem com o sistema na funcionalidade detalhada
Pré-condições: Situação necessária para que a funcionalidade possa ser executada (ex: emitir um extrato de uma conta é necessário que a máquina tenha papel)
Pós-condições: Situação do sistema após a execução da funcionalidade
Requisitos correlacionados: Requisitos que são executados no decorrer da execução da funcionalidade que está sendo detalhada (ex: autenticar um cliente antes do mesmo emitir um extrato de sua conta)
Variações Tecnológicas : Especifica restrições, funções auxiliares que serão executadas, permissões etc.
Prioridade: Especifica o grau de desejo da implementação da funcionalidade descrita (essencial, importante, desejável)
Fluxo principal: Especifica os passos principais no decorrer da execução da funcionalidade 1. O ator aciona um determinado botão. 2. O sistema apresenta uma tela. 3. O ator executa uma função proposta pela tela apresentada (Fa001) (Fa002). 4. O ator aciona o botão A (Fe002). 5. Funcionalidade concluída com sucesso.
Fluxo de exceções: (Fe001) Primeira exceção. O sistema apresenta uma mensagem com o erro A e volta para o fluxo principal no passo 3. (Fe002) Segunda exceção. O sistema apresenta uma mensagem com o erro B e volta para o fluxo principal no passo 3.
Fluxo alternativo: (Fa001) Primeira alternativa de execução. (Fa002) Segunda alternativa de execução. O ator aciona o botão B (Fe001) (Fe002).
GUI: Telas associadas (sigla de identificação das telas)

Fonte: Adaptado de Martins, 2010.

- **Diagrama de classes**

Característica marcante da orientação a objetos, uma classe representa algo do mundo real (ex: carro, pessoa, animal), descrevendo suas características (ex: cor, altura, peso, nome), como também suas ações (ex: beber, comer, correr). A cada carro, pessoa ou animal criado no sistema, dar-se o nome de objeto, sendo este uma instância da classe à qual pertence.

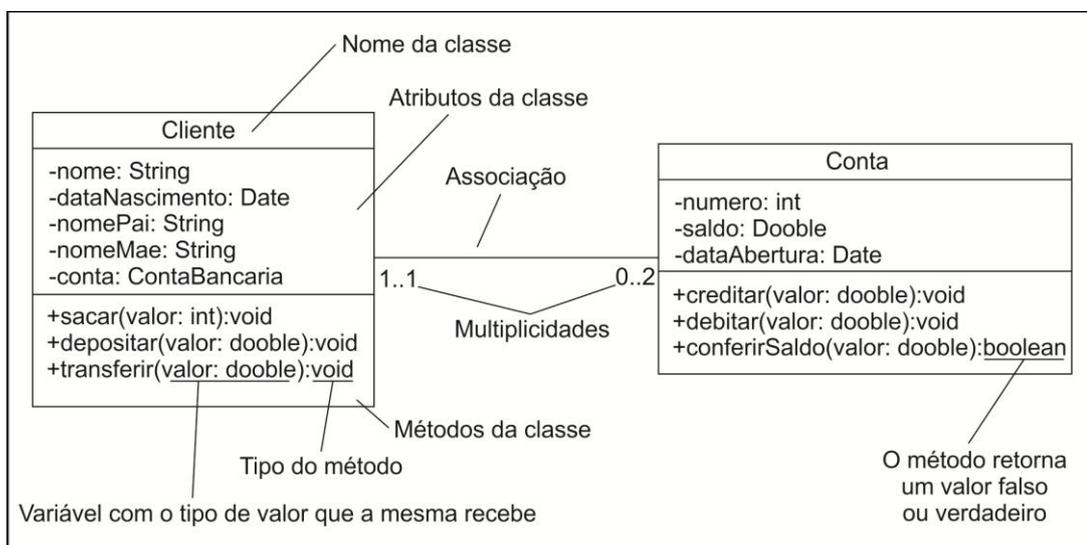
Para Bezerra (2007, p. 7):

Na terminologia da orientação a objetos, cada ideia é denominada classe de objetos, ou simplesmente classe. Uma classe é uma descrição dos atributos e serviços comuns a um grupo de objetos. Sendo assim, pode-se entender uma classe como sendo um molde a partir do qual objetos são construídos.

Visto os objetos serem criados a partir de uma classe e que a essência de um sistema orientado a objetos é a interação entre estes, é de extrema importância uma boa definição e modelagem das mesmas. Para tal, a UML dispõe do diagrama de classes, recurso visual que possibilita um melhor entendimento da estrutura do sistema no momento de seu desenvolvimento com uma linguagem de programação.

Este diagrama é composto por: classes com seus atributos e métodos; associações; e multiplicidades, conforme mostra a Figura 13.

Figura 13: Elementos de um diagrama de classes.

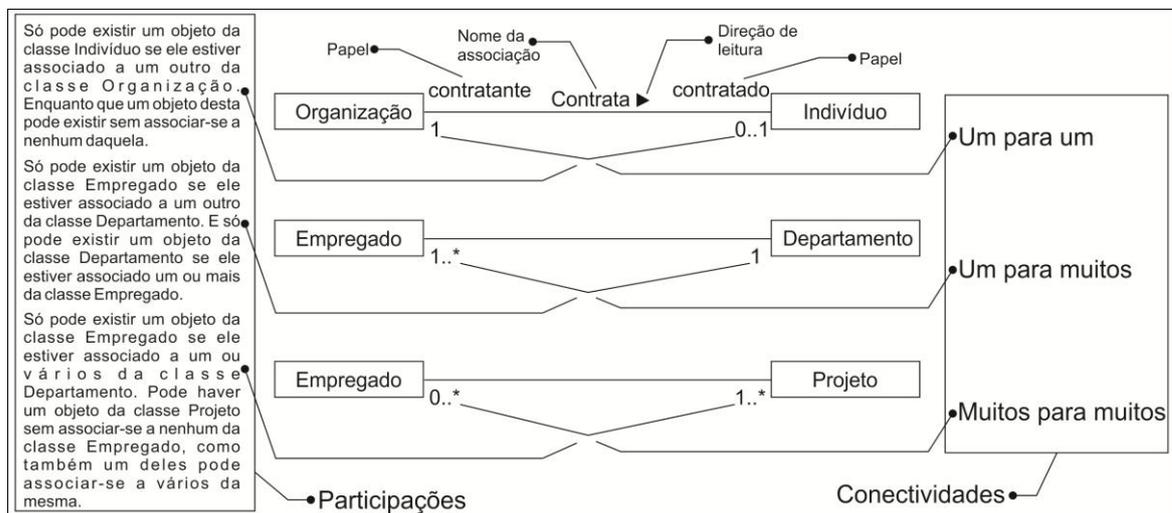


Fonte: Adaptado de Bezerra, 2007.

Fazendo parte das associações, a multiplicidade define a quantidade de instâncias de uma classe que se relaciona com um determinado objeto de outra. No exemplo da Figura 13, um objeto da classe “Cliente” pode não se relacionar com nenhum ou até dois objetos da classe “Conta”, enquanto que uma instância desta só pode referenciar uma daquela.

Além das multiplicidades, as associações podem ter nomes, direção de leitura, papéis, tipo de participação e conectividade (BEZERRA, 2007), conforme mostra a Figura 14.

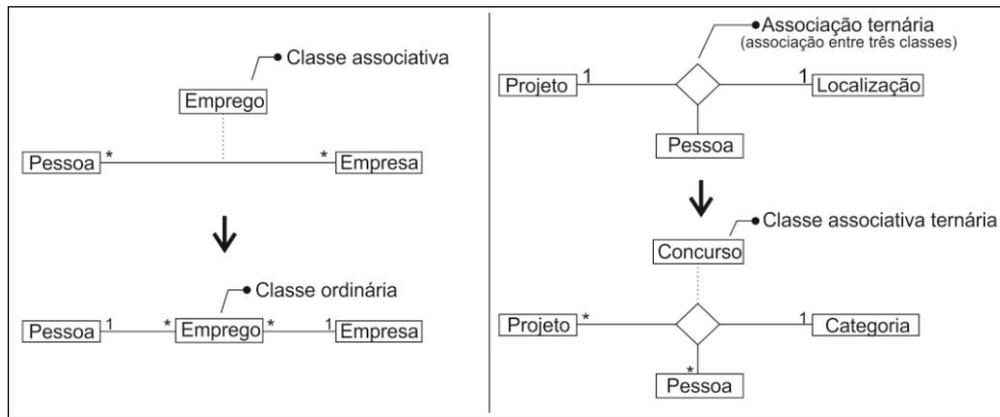
Figura 14: Diagrama de classes (características de uma associação).



Fonte: Adaptado de Bezerra, 2007.

Sendo a associação um tipo de relacionamento entre classes, a mesma pode, além de fazer essa ligação entre elas, relacionar uma classe a outra associação, caso chamado de classe associativa. Tal caso pode ter o seu relacionamento substituído por uma associação entre duas classes, transformando-se em uma ordinária, ou até mesmo manter-se como tal e se relacionar com uma associação ternária, conforme mostra a Figura 15.

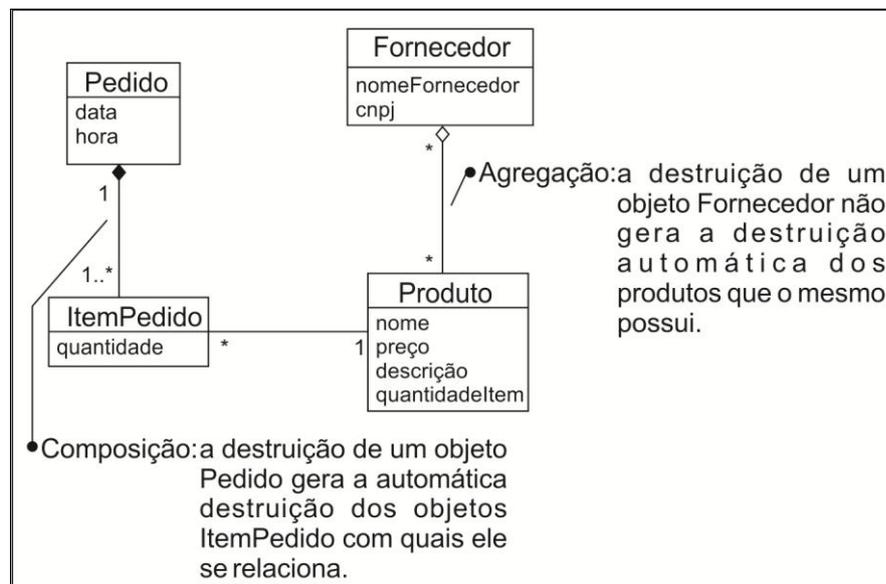
Figura 15: Exemplo de classe associativa e associação ternária.



Fonte: Adaptado de Bezerra, 2007.

As associações podem ter vários significados, significados esses que mudam conforme a mesma é representada graficamente, como ocorre com a linha que liga uma classe associativa, pois, diferente das demais, a mesma é tracejada. As mesmas ainda podem indicar que um objeto é formado por um conjunto de outros (agregação), ou que ele é parte de outro objeto (composição). Ambas com suas características conforme mostra a Figura 16.

Figura 16: Exemplo de agregação e composição.

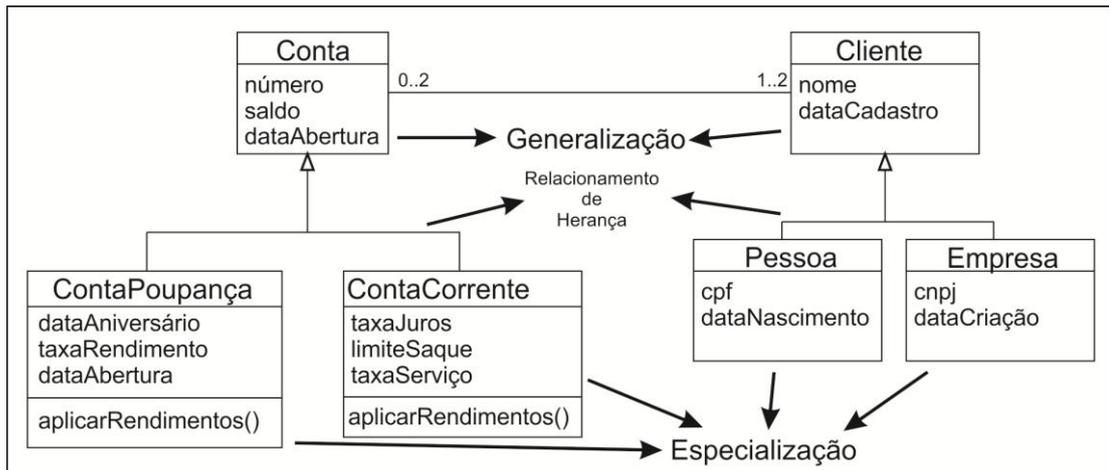


Fonte: Adaptado de Bezerra, 2007.

Apesar dos diagramas apresentarem relacionamento entre classes, a comunicação efetivamente ocorre entre objetos delas. No entanto existe um

relacionamento que representa a relação entre classes. A herança denota a existência de uma generalização e especialização, termos que definem uma classe mais genérica e outra mais específica, sendo que esta herda atributos e métodos daquela, conforme mostra a Figura 17.

Figura 17: Exemplo de relacionamento de herança.



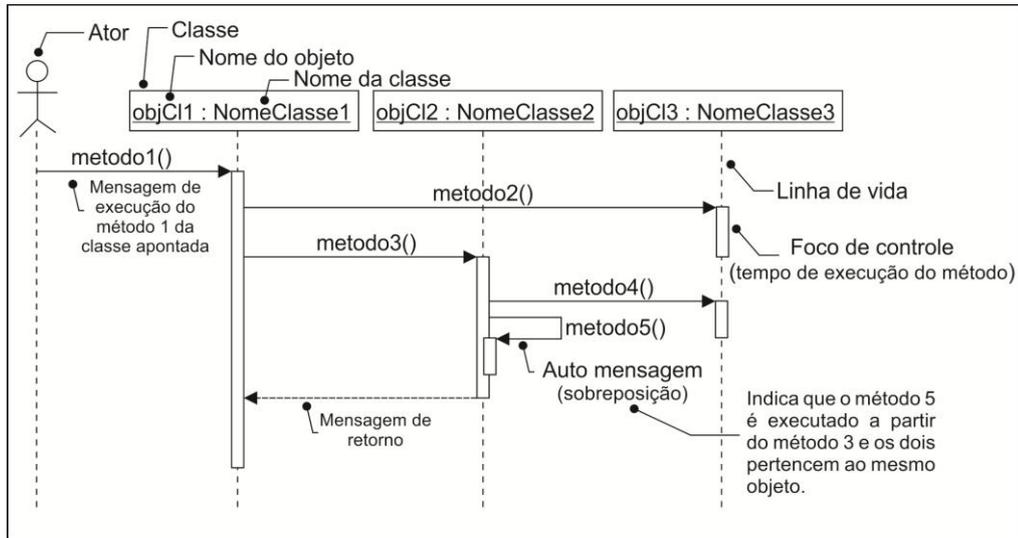
Fonte: Adaptado de Bezerra, 2007.

- **Diagrama de sequência**

A UML consegue representar um sistema tanto de forma estrutural como dinâmica. Enquanto o diagrama de classes modela estruturalmente um *software*, os de caso de uso e de sequência representam o mesmo de forma dinâmica. Apesar do modelo de casos de uso especificar a interação do sistema com seu ambiente externo, o mesmo não possui detalhe suficiente para modelar o funcionamento interno do mesmo, ou seja, como cada funcionalidade é executada. Para isso a UML dispõe do diagrama de sequência, responsável por detalhar como cada caso de uso é executado através da interação entre objetos pertencentes às classes do modelo estrutural.

Esse modelo diagramático mostra com qual objeto um determinado ator interage, qual funcionalidade desse objeto é ativada e a sequência de interação entre ações de outros objetos distintos até o retorno e apresentação de um resultado ao ator, conforme Figura 18.

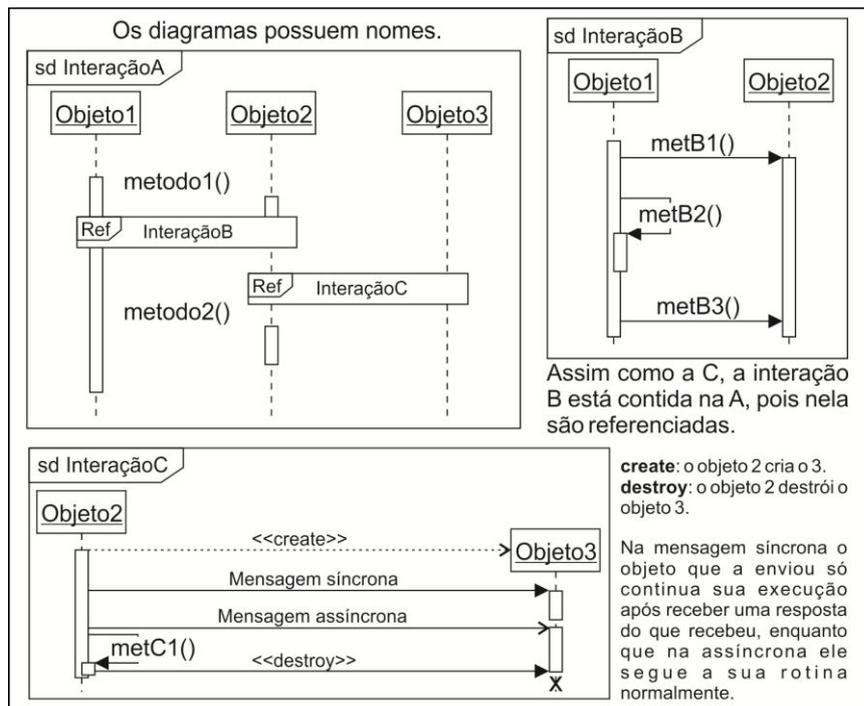
Figura 18: Elementos básicos de um diagrama de sequência.



Fonte: Adaptado de Bezerra, 2007.

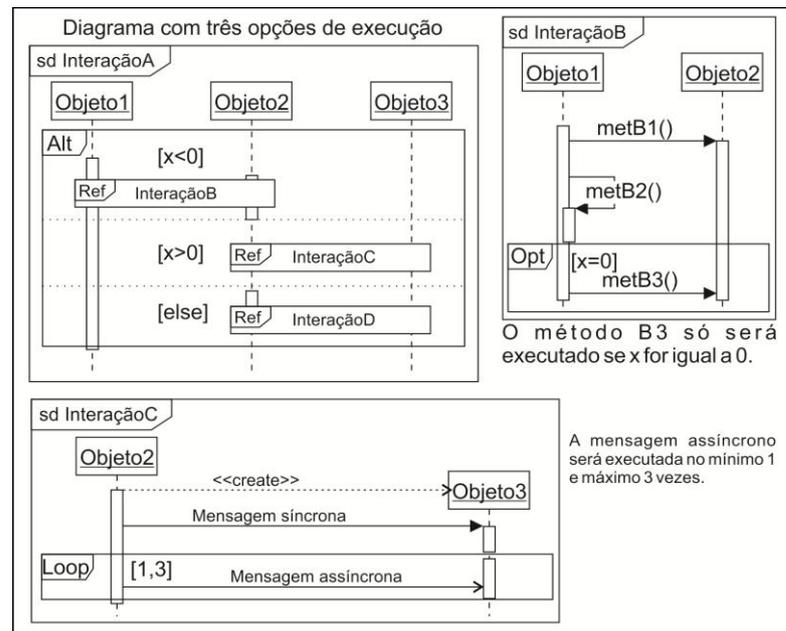
As mensagens têm diversas notações, podendo assim apresentar significados diferentes, como um objeto que cria ou destrói outro etc. Além delas, esse diagrama conta com outros elementos que possibilitam dar um nome a um determinado diagrama, referenciar outro dentro deste, como também definir condições para execução de interações, conforme mostram as Figuras 19 e 20.

Figura 19: Nome do diagrama, referências e mensagens com notação.



Fonte: Adaptado de Bezerra, 2007.

Figura 20: Diagrama de sequência com operadores



Fonte: Adaptado de Bezerra, 2007.

- **Diagrama de máquina de estado**

Durante todo o processo de utilização de um objeto oriundo de uma classe, o mesmo passa por diversos estados, ou seja, características que de algum modo os classifica de maneiras diferentes.

Um objeto chamado “compra” pode inicialmente ser classificado como “adicionando itens”, depois como “identificando forma de pagamento” e por fim como “compra finalizada”, estado este do qual o mesmo não sai mais.

Para (PEREIRA, 2011), os estados que um determinado objeto apresenta são caracterizados pela situação atual de seus atributos, como também pelos relacionamentos entre objetos da mesma classe.

Sendo assim, o diagrama de máquina de estados especifica os estados dos objetos pertencentes à mesma classe, e o valor dos atributos são determinantes para mudar o estado dos mesmos.

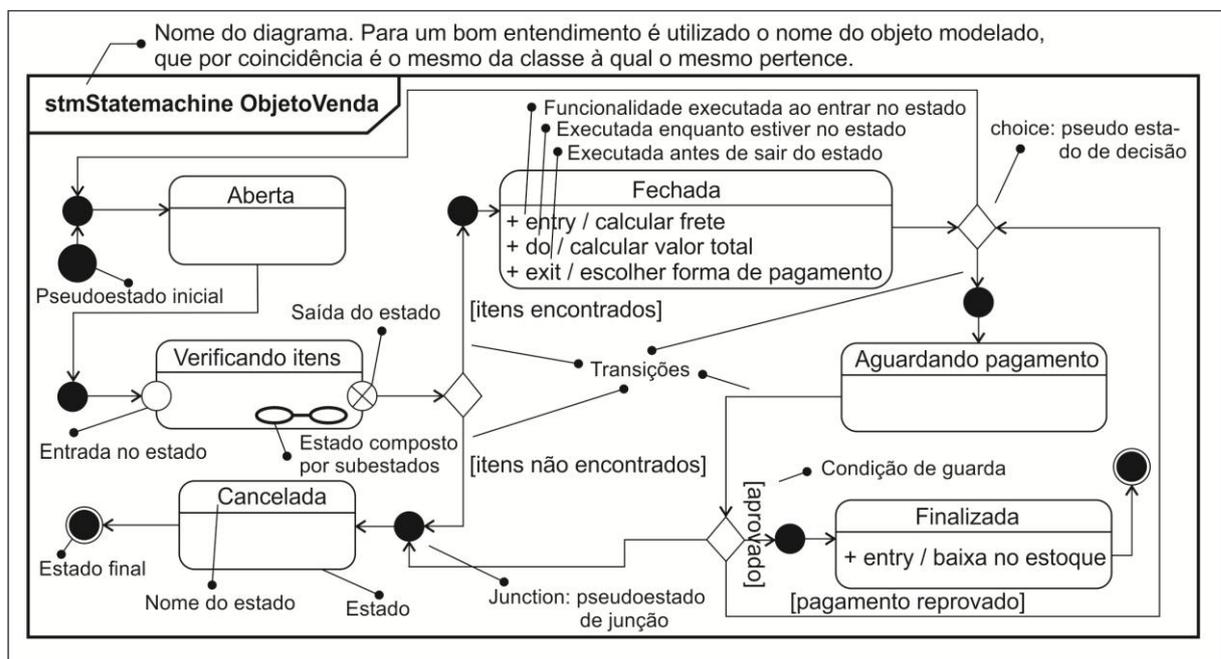
O desenvolvimento desse diagrama é realizado com base nas informações obtidas das classes do sistema, pois elas contêm a coleção de objetos. No entanto

nem todas elas necessitam de terem os estados de seus objetos detalhados.

Para (PEREIRA, 2011), este diagrama relata a história de um determinado objeto e que a mesma só vale a pena ser contada quando é vista como algo relevante na construção do sistema, caso contrário não há necessidade.

Seguindo a mesma lógica dos outros diagramas, a UML define diversos elementos que compõem o diagrama de máquina de estados, sendo os mesmos utilizados para auxiliar na interpretação daquele, conforme mostra a Figura 21.

Figura 21: Elementos de um diagrama de máquina de estados



Fonte: Adaptado de Pereira, 2011.

Além de modelar os estados de instâncias de classes, o diagrama de máquina de estados também é utilizado com outras finalidades, tal como modelar o estado de interfaces gráficas do sistema.

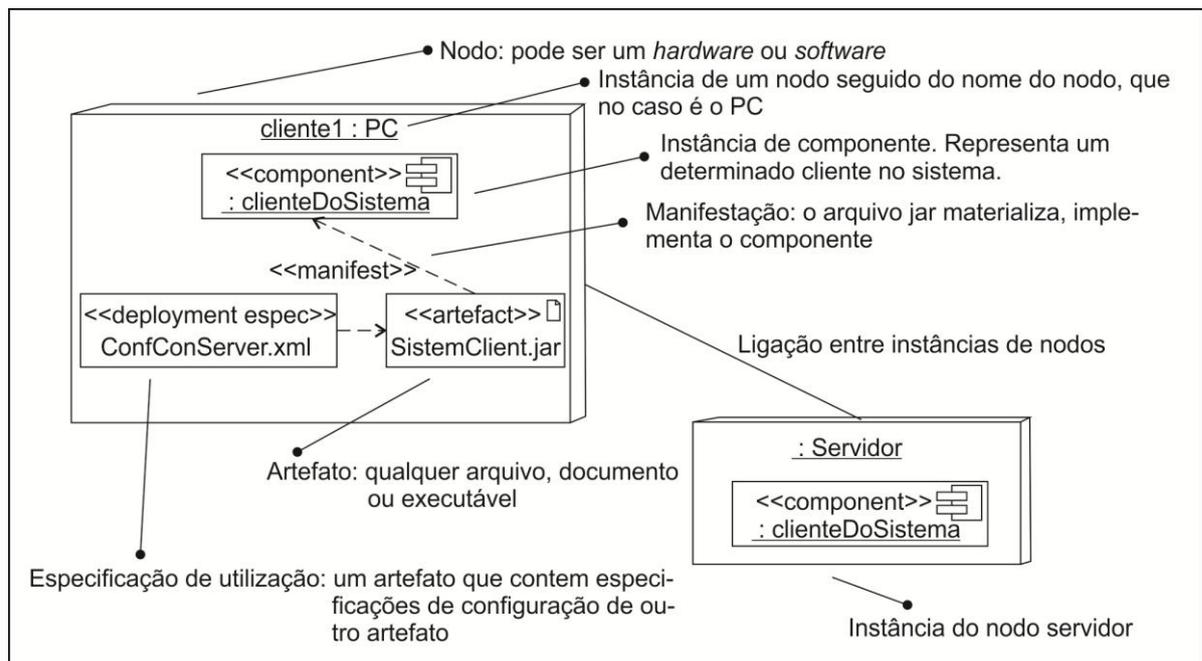
- **Diagrama de instalação ou utilização**

Em algumas situações o engenheiro de *software* necessita de modelar a organização de um sistema, definindo suas partes integrantes, tais como um leitor biométrico, sensor de temperatura, computador cliente, servidor etc., e suas

ligações, definindo como eles se colaboram para o funcionamento geral do projeto.

Para isso a UML conta com o *deployment diagram*, termo como originalmente é conhecido, no entanto ele pode aparecer como diagrama de utilização, instalação, configuração etc. O mesmo possui diversos elementos com significados que possibilitam definir um sistema com certa clareza, conforme Figura 22.

Figura 22: Elementos de um diagrama de utilização



Fonte: Adaptado de Pereira, 2011.

2.2.4 Modelagem de banco de dados

Um *software* recebe dados, processa os mesmos e apresenta um resultado. No paradigma de orientação a objetos, dados são armazenados em atributos de objetos. Um objeto só existe no sistema enquanto este está aberto, sendo destruído no momento de seu desligamento. No entanto, a maioria dos projetos de sistema demanda a necessidade de armazená-los de forma persistente, possibilitando sua utilização mesmo após o sistema ter sido fechado.

O banco de dados é um recurso que supre essa necessidade, armazenando objetos que possam ser utilizados em sessões diferentes de uso do sistema. Segundo (PEREIRA, 2011), um objeto pode ser classificado como transiente

(existente apenas enquanto o sistema é executado) e persistente (quando o mesmo é alocado na memória secundária e carregado na primária assim que o *software* necessita do mesmo).

Um banco de dados é formado por tabelas relacionadas entre si, sendo também chamado por banco de dados relacional. Cada tabela é composta por linhas, cuja função é representar um determinado objeto (instância). Cada instância de uma tabela se relaciona com a instância de outra, mantendo assim uma relação lógica comparada a de orientação a objetos. Uma linha ou instância possui diversos atributos que armazenam um determinado tipo de dado, conforme a Figura 23.

Figura 23: Tabelas de um banco de dados relacional

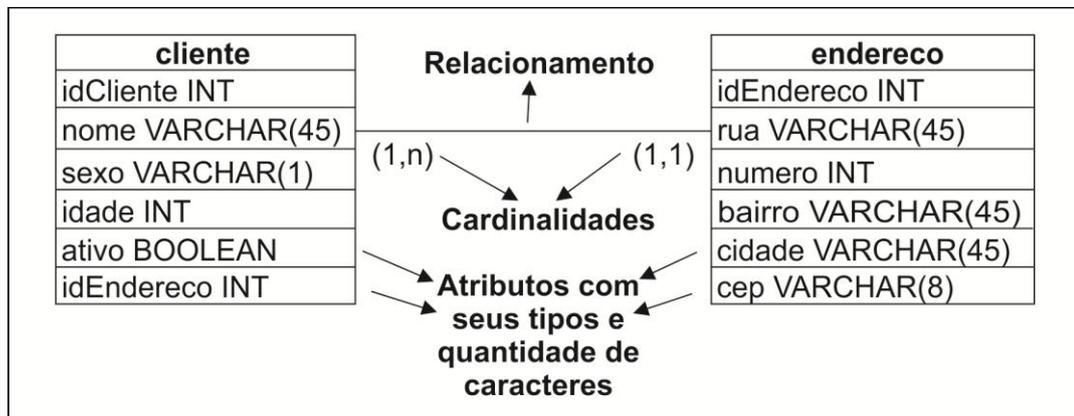


Fonte: Adaptado de Bezerra, 2007.

Visto as lógicas de orientação a objetos e banco de dados relacional serem diferentes, é de extrema importância, quando o projeto demandar a necessidade de um banco de dados, mapear os objetos de forma a distribuir seus elementos e relacionamentos entre si no modelo de banco de dados, ação conhecida como “modelagem de banco de dados”.

O diagrama UML utilizado como fonte de informações para o desenvolvimento do modelo relacional é o diagrama de classes, visto o mesmo conter elementos que compõem os objetos. Assim como esse diagrama, o modelo relacional conta com atributos, tipos de dados, relacionamentos e a quantidade de vezes que uma instância de uma tabela se relaciona com a instância de outra (cardinalidade), conforme Figura 24.

Figura 24: Tabelas com atributos e relacionamentos



Fonte: Adaptado de Bezerra, 2007.

O atributo utilizado como chave exerce uma função de suma importância em um banco de dados, pois é ele que possibilita a efetiva relação entre instâncias de tabelas diferentes. Cada instância de uma tabela possui uma chave que a identifica de forma única, chamada de chave primária. Quando esta chave é inserida em outra tabela ela passa a ser chamada de chave estrangeira.

Desenvolver um diagrama de forma que venha possibilitar a criação de um banco que armazene objetos de um sistema pode ser uma tarefa difícil quando realizada sem uma ferramenta adequada. A IDE *MySQL Workbench* possibilita a criação de esquemas relacionais com atributos, relações entre tabelas e restrições de implementação, como a criação de campos que não poderão conter valores nulos, gerando posteriormente a codificação na linguagem SQL de forma automática.

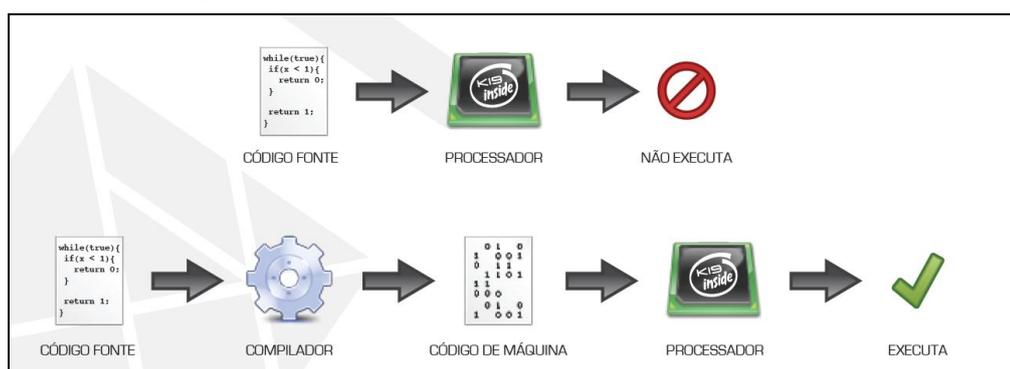
2.3 Programação de computadores

Uma vez desenvolvido um projeto de *software*, o mesmo necessita de efetivamente ser criado para assim executar as tarefas previamente planejadas. Esta criação se dá através da programação, ação de descrever, de forma escrita, instruções que a máquina (*hardware*) entenda, pois ela é limitada a uma única língua, a chamada linguagem de máquina, composta apenas pelos números 0 e 1.

Para facilitar o trabalho de programar um computador foram criadas diversas linguagens de programação, linguagens essas mais simples de serem compreendidas pelo o homem, visto ser um trabalho praticamente impossível escrever um programa utilizando a linguagem de máquina. Elas são classificadas como de baixo ou alto nível, conforme a compreensão das mesmas se aproxima da máquina ou do ser humano.

Para (K9 TREINAMENTOS, 2012), o local composto por instruções e comandos de uma linguagem de programação é chamado de código fonte. Mas eles não são compreendidos pela máquina que só trabalha com números binários (0 e 1). Todavia o algoritmo pode ser transformado em linguagem de máquina com a utilização de um compilador, *software* que cria um arquivo executável (conjunto de números binários) a partir do código fonte, tudo conforme a Figura 25.

Figura 25: Processo de criação de um arquivo executável



Fonte: K9 Treinamentos (2012, p. 4).

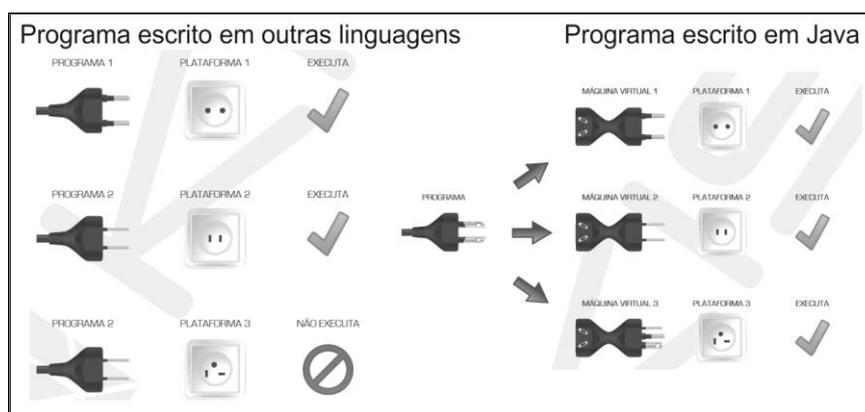
2.3.1 JAVA

Java é uma linguagem de programação de alto nível compatível com o paradigma de orientação a objetos. Utilizada tanto para desenvolvimento WEB quanto Desktop, um *software* desenvolvido com utilização da mesma pode ser executado em qualquer Sistema Operacional.

Para (CLARO, 2008), ao compilar um código fonte escrito em Java é criado um arquivo com códigos binários na extensão *.class*, sendo este interpretado e executado por uma máquina virtual, cuja função é acessar os recursos do *hardware* através do sistema operacional e efetivamente fazer o *software* funcionar.

A linguagem conta com uma máquina virtual específica, a JVM (Java Virtual Machine), sendo necessário que apenas ela seja compatível com o SO (Windows, Linux etc.), ficando o *software* executável em qualquer plataforma que a tenha instalada, tornando o Java diferente das demais linguagens, conforme a Figura 26.

Figura 26: Java multiplataforma



Fonte: Adaptado de K9 Treinamentos, 2012.

Por se tratar de uma linguagem de Orientação a Objetos, programar em Java é criar classes com seus atributos e métodos de forma que as mesmas gerem objetos e estes troquem mensagens entre si. Como tal, vários conceitos desse paradigma de programação são tratados nessa linguagem, tais como classe, atributos e objetos.

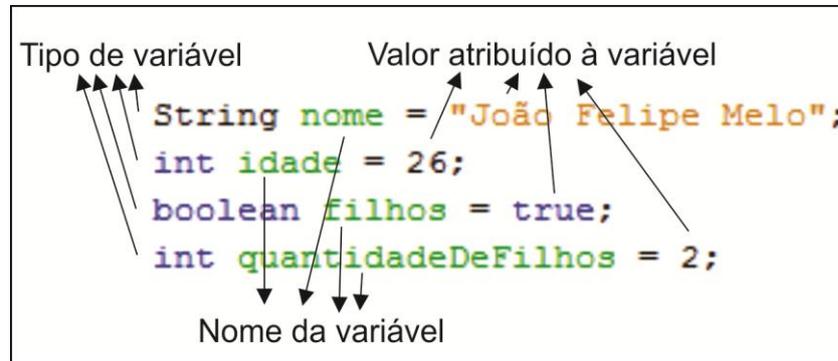
2.3.1.1 Atributo

Também chamado de variável, o atributo é responsável por armazenar um determinado tipo de dado, conforme necessidade do programador ou diagrama de classes elaborado pelo projetista.

Para (CLARO, 2008), o tipo de atributo determina o tipo de dado que pode ser armazenado no mesmo, não podendo ele receber outro tipo que não seja o declarado em seu escopo.

Visto o mesmo receber um dado sobre algo, o atributo é utilizado para definir as características de um objeto, tais como cor dos olhos de um animal ou nome de uma pessoa, sendo declarado e manipulado conforme a Figura 27.

Figura 27: Declaração de variáveis em Java



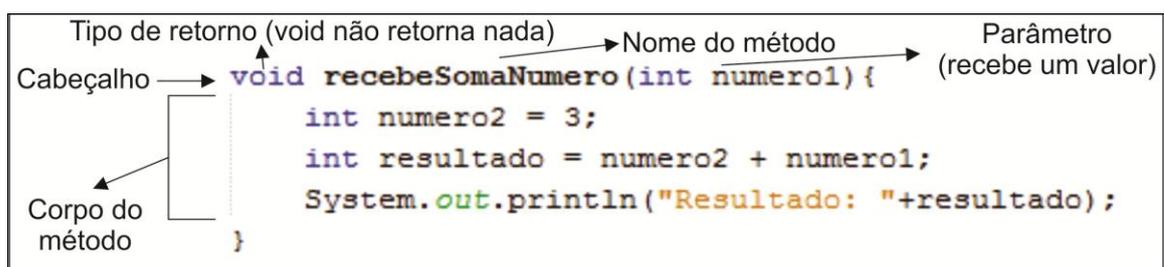
Fonte: Adaptado de Claro, 2008.

2.3.1.2 Método

Responsável por realizar uma ação, o método pode tanto receber como retornar um valor. Assim como o atributo, estes valores só podem ser do tipo declarado, rejeitando qualquer outro.

Para (CLARO, 2008), o método pode conter variáveis, comandos, retornar um determinado tipo de valor ou até mesmo não retornar nada, sendo declarado conforme a Figura 28.

Figura 28: Declaração de método em Java



Fonte: Adaptado de Claro, 2008.

2.3.1.3 Classe

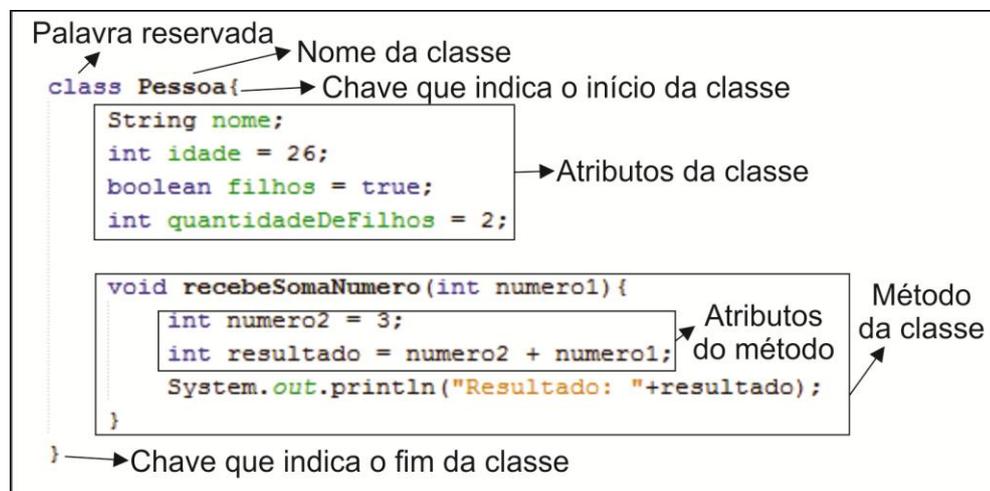
Composta pelos atributos e métodos, uma classe é um molde de onde são gerados objetos, objetos esses com as características e funções definidas no momento da declaração da classe pelo programador.

Para K9 Treinamentos (2012, p. 24):

Uma classe funciona como uma “receita” para criar objetos. Inclusive, vários objetos podem ser criados a partir de uma única classe. Assim como várias casas ou prédios poderiam ser construídos a partir de uma única planta; ou vários bolos poderiam ser preparados a partir de uma única receita; ou vários carros poderiam ser construídos a partir de um único projeto.

A criação de uma classe em Java é sempre iniciada pela palavra reservada “class” da linguagem, seguida pelo o nome da mesma, conforme Figura 29.

Figura 29: Criação e composição de uma classe



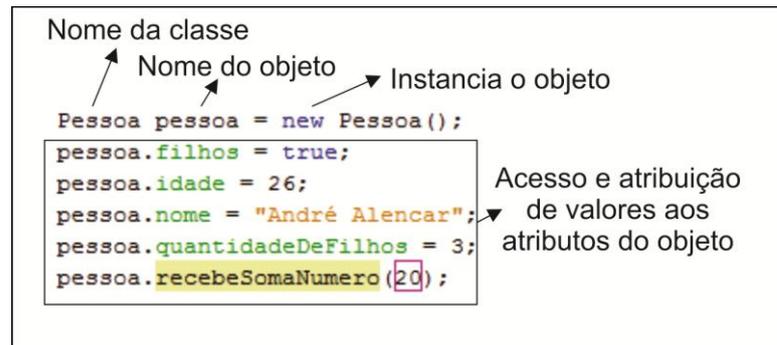
Fonte: Adaptado de K9 Treinamentos, 2012.

2.3.1.4 Objeto

Um objeto é uma instância de uma determinada classe, contendo os elementos que a mesma oferece (atributos e métodos), sendo que os atributos armazenam características do objeto, enquanto que os métodos representam as ações que o mesmo realiza.

A funcionalidade de um *software* orientado a objetos está na interação entre estes, ou seja, a troca de mensagens entre os objetos. Sendo assim, a criação e manipulação de objetos em um *software* são de suma importância, sendo essas ações realizadas na linguagem Java conforme a Figura 30.

Figura 30: Criação e manipulação de objetos em Java

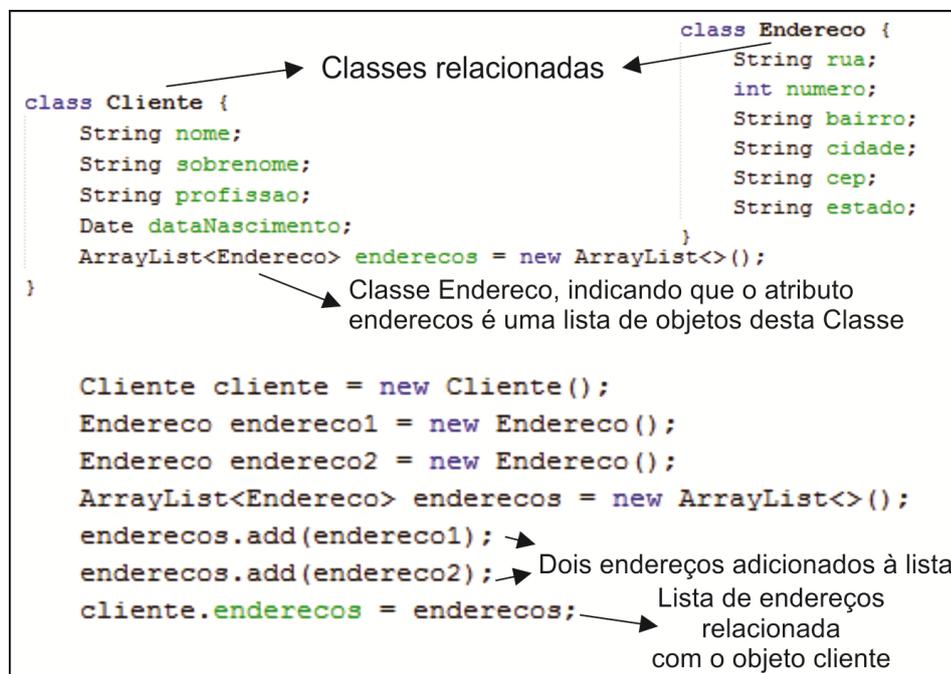


Fonte: Adaptado de K9 Treinamentos, 2012.

2.3.1.5 Relacionamento

Ligar objetos de tal forma que um possa ser interpretado como pertencente a outro é algo necessário e até mesmo obrigatório na programação. Um ou vários endereços pertencentes a um dado cliente é um exemplo de relacionamento na linguagem Java, sendo tal situação implementada conforme a Figura 31.

Figura 31: Relacionamento entre objetos em Java



Fonte: Adaptado de K9 Treinamentos, 2012.

2.3.1.6 Modificadores de acesso

Para Claro (2008, p. 37), modificadores de acesso ou regra de acesso são utilizados na declaração de classes, métodos e atributos. Sendo assim eles interferem diretamente na forma como os objetos são manipulados.

- **private**

Permite acesso aos atributos e métodos somente de dentro da própria classe, exigido o uso de método para manipulação dos objetos.

- **public**

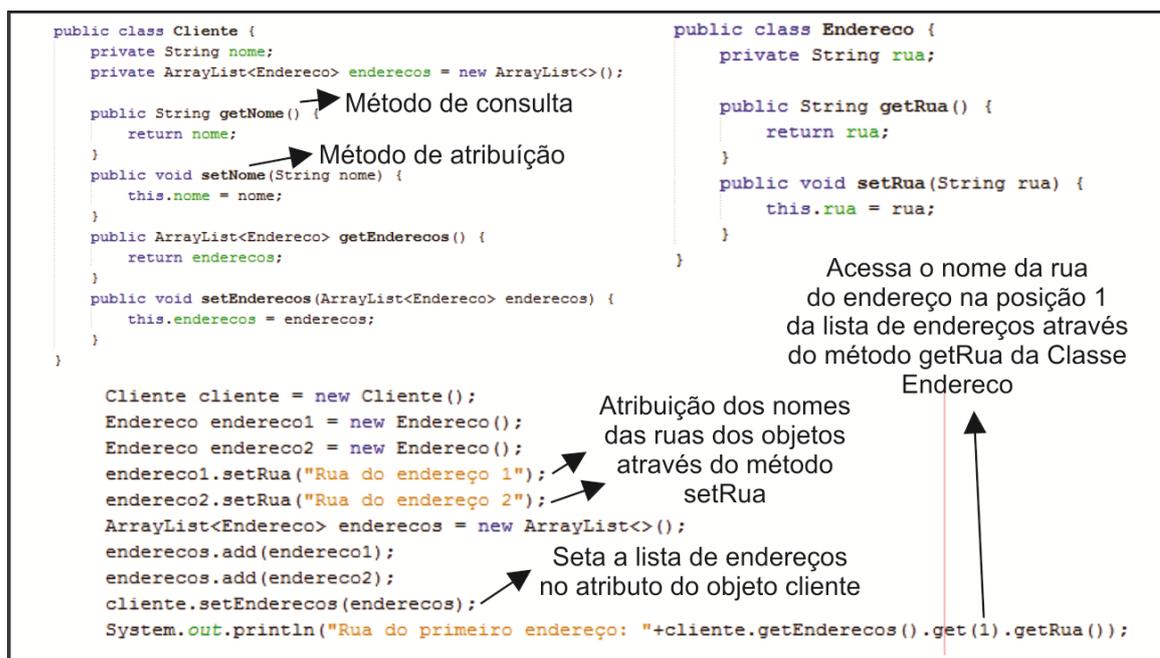
Permite o acesso aos atributos e métodos de fora da classe a qual esses pertencem. Quando utilizado na declaração de uma classe a mesma fica visível para outras classes mesmo que estas pertençam a pacotes distintos.

- **protected**

Permite acesso a classes do mesmo pacote.

Os atributos declarados como protegidos somente são manipulados com a utilização de métodos, pois é através deles que valores são atribuídos ou consultados nessas variáveis, conforme a Figura 32.

Figura 32: Manipulação de objetos com atributos protegidos.



Fonte: Adaptado de Claro, 2008.

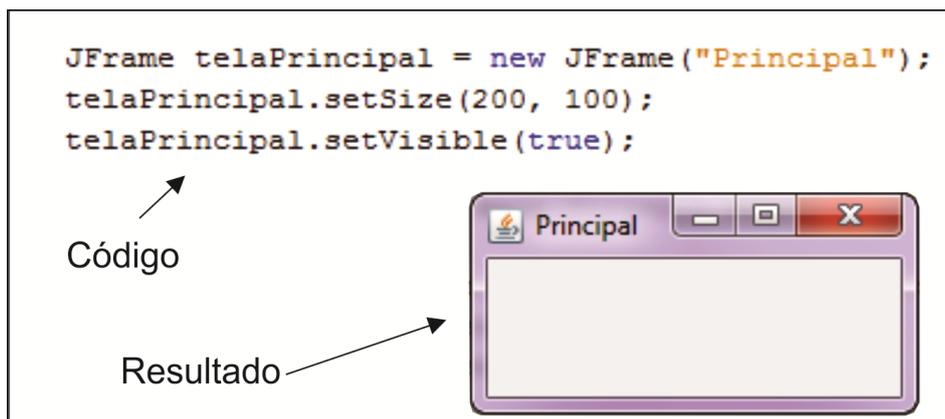
2.3.1.7 Swing

Para (K9 TREINAMENTOS, 2012), é um recurso da linguagem Java que cria interface gráfica com o usuário, possibilita a interação deste com o *software* através de seus componentes (tela, campo de texto, botão, lista, tabela), simplificando a utilização do sistema.

- **JFrame**

Classe utilizada para criar uma tela com título, botões de fechar, minimizar e maximizar, sendo criada e apresentada conforme a Figura 33.

Figura 33: Criando uma tela com Java.

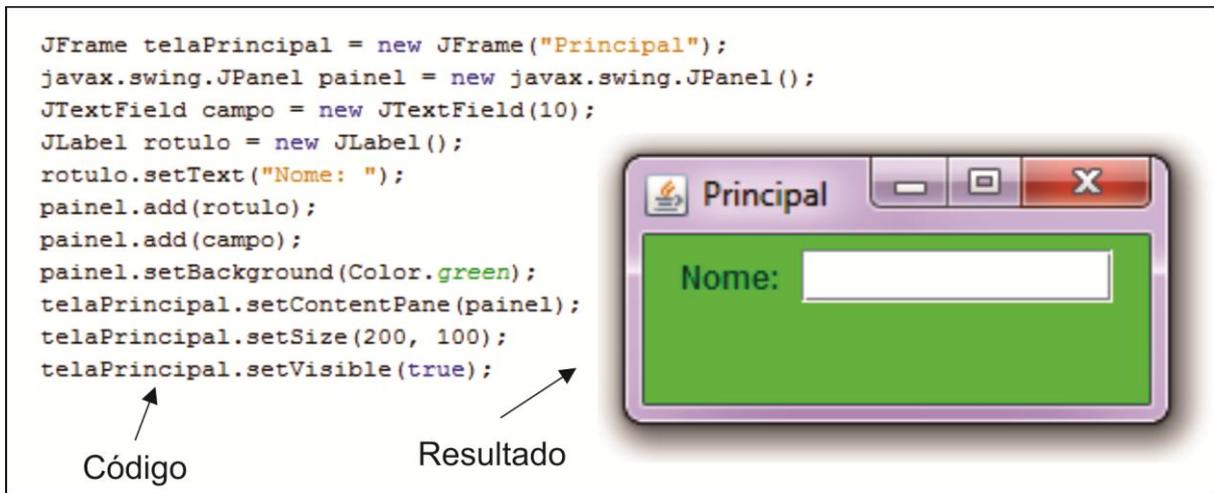


Fonte: Adaptado de K9 Treinamentos, 2012.

- **JPanel, JTextField e JLabel**

Podendo ser adicionado em um JFrame, o JPanel é utilizado para agrupar determinados componentes em uma tela, dividido-a em diversos setores com objetivos e cores diferentes. Enquanto que JTextField define um campo de entrada ou saída de texto e JLabel apresenta um rótulo, sendo eles componentes facilmente adicionados a um JPanel, tudo conforme Figura 34.

Figura 34: Tela com JPanel, JTextField e JLabel.

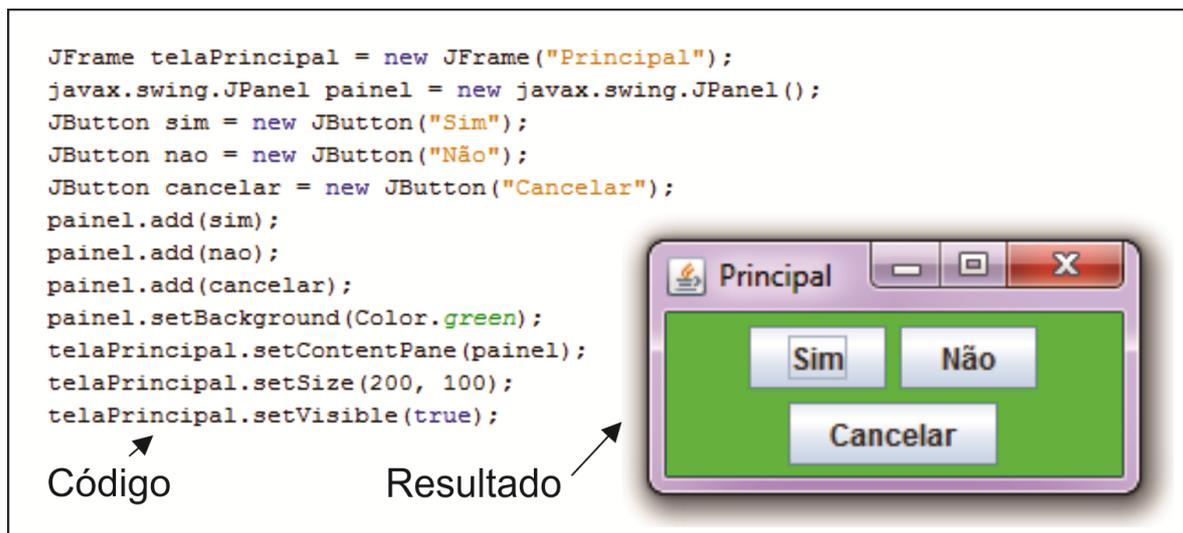


Fonte: Adaptado de K9 Treinamentos, 2012.

- **JButton**

Para (K9 TREINAMENTOS, 2012), é uma classe que possibilita a criação de botões para execução de determinadas ações quando acionados pelo usuário, sendo criado conforme Figura 35.

Figura 35: Utilizando a classe JButton.



Fonte: Adaptado de K9 Treinamentos, 2012.

2.3.1.8 Socket

Para Claro (2008, p. 77), um *socket* é uma conexão de dados transparente entre dois computadores em uma rede. Ele é identificado pelo endereço de rede dos computadores, seus pontos finais e uma porta em cada computador.

Java disponibiliza duas classes para desenvolvimento de comunicação entre dois computadores em uma rede, sendo elas *ServerSocket* e *Socket*. Enquanto esta cria um cliente, aquela implementa um servidor.

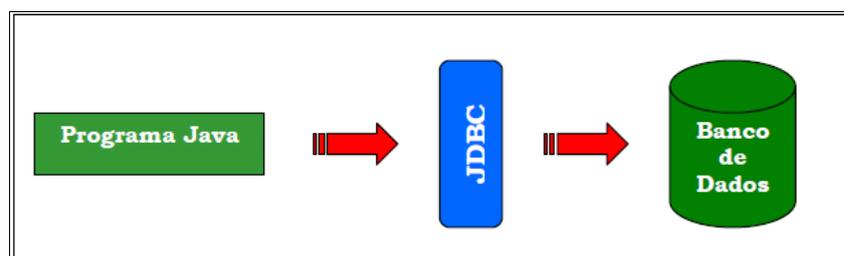
Segundo (CLARO, 2008), servidor é um programa rodando em um computador que fica aguardando requisições de outros programas (clientes) em outros computadores.

Através dos métodos das classes *ServerSocket* e *Socket* é possível o envio de mensagens e até objetos Java entre computadores diferentes, possibilitando o desenvolvimento de *software* em módulos distintos mas com objetivos comuns.

2.3.1.9 JDBC

Segundo (CLARO, 2008), JDBC (*Java Database Connectivity*) é uma API (*Application Programming Interface*) desenvolvida com o objetivo de criar uma ponte de conexão entre uma aplicação Java e um Gerenciador de Banco de Dados, possibilitando a manipulação de dados do banco via *software*, conforme Figura 36.

Figura 36: Java e JDBC.



Fonte: Claro (2008, p. 83).

2.3.2 SQL

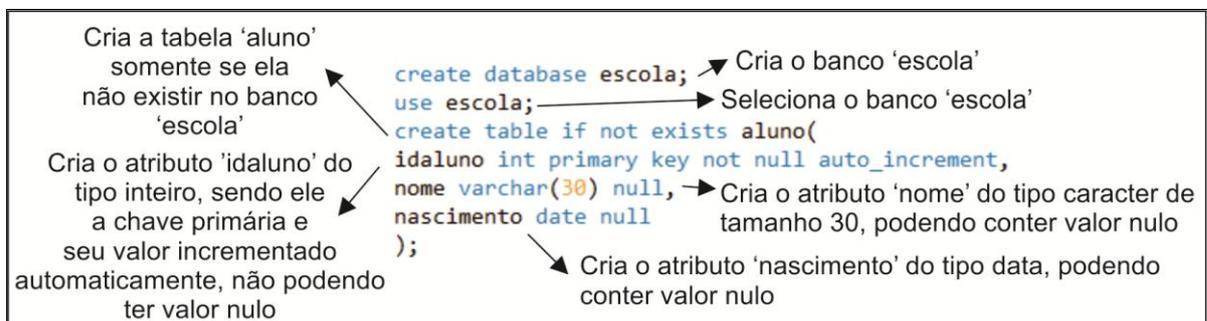
Para (ELMASRI, 2011), SQL é uma linguagem que possibilita o desenvolvimento e administração de bancos de dados quando combinado com um SGBD (Sistema Gerenciador de Banco de Dados). Através de seus comandos é

possível criar tabelas com relacionamentos entre elas, como também realizar consultas, exclusão e atualizações de dados nestas.

2.3.2.1 *Create*

Cria um banco de dados ou uma tabela de um banco, sendo neste último caso combinado com outros comandos com o objetivo de criar regras, conforme Figura 37.

Figura 37: Comando *create*.



Fonte: Adaptado de Elmasri, 2011.

2.3.2.2 *Insert*

Cria uma linha em uma determinada tabela, inserido valores nos atributos especificados, dispensando a inserção de dados no campo correspondente à chave primária, visto o mesmo ser autoincrementado. Sua sintaxe é mostrava na Figura 38.

Figura 38: Comando *insert*.

```
use escola;
insert into aluno (nome, nascimento) values ('João Felipe', '1986-02-27');
insert into aluno (nome, nascimento) values ('Maria Felipe', '1986-12-27');
```

Fonte: Adaptado de Elmasri, 2011.

2.3.2.3 *Update*

Utilizado para alterar valores de atributos, este comando possibilita atualizar, em uma única linha, mais de uma instância de tabela, conforme Figura 39.

Figura 39: Comando *update*.

```
use escola;
update aluno set nascimento = '1990-01-30' where idaluno > 1;
```

Atualiza o atributo 'nascimento' para o valor 1990-01-30 de todas as linhas cujo campo 'idaluno' seja maior que 1.

```
use escola;
update aluno set nascimento = '1990-01-30' where nascimento = '1986-02-27';
```

Atualiza o atributo 'nascimento' de todas as linhas com valores iguais a 1986-02-27 neste mesmo campo para 1990-01-30.

Fonte: Adaptado de Elmasri, 2011.

2.3.2.4 **Select**

Utilizado para consultar valores em tabelas de um banco de dados, este comando pode apresentar estrutura simples, como o retorno de todas as linhas e atributos de uma tabela, como também uma estrutura mais avançada, como o retorno de determinadas instâncias e dados específicos destas. Sua sintaxe é apresentada na Figura 40, mudando sua estrutura conforme a necessidade da consulta.

Figura 40: Comando *select*.

```
select * from aluno;
```

Retorna todas as instâncias com todos atributos da tabela aluno.

```
select * from aluno where nascimento = '1986-02-27';
```

Retorna as instâncias com todos atributos da tabela aluno cujo os valores de seus campos 'nascimento' sejam iguais a 1986-02-27.

```
select nome from aluno where nascimento = '1986-02-27';
```

Retorna as instâncias com o atributo 'nome' da tabela aluno cujo valores de seus campos 'nascimento' sejam iguais a 1986-02-27.

```
select * from aluno where nome like '%ão%';
```

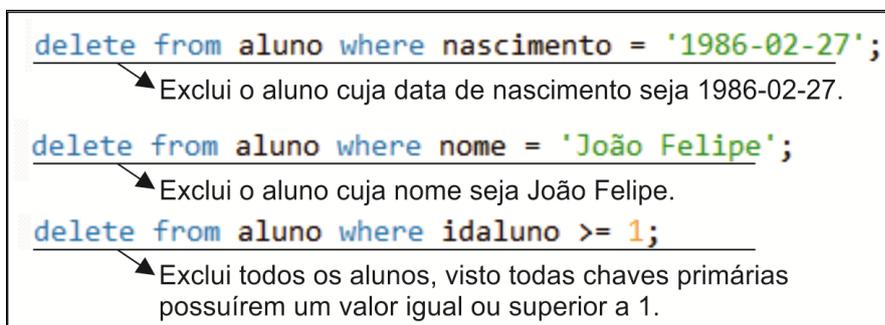
Retorna as instâncias com todos atributos da tabela aluno cujo valores de seus campos 'nome' tenha as letras ão, estejam elas posicionadas no início, meio ou no fim do conteúdo do atributo.

Fonte: Adaptado de Elmasri, 2011.

2.3.2.5 **Delete**

Utilizado para excluir linhas em uma tabela, esse comando pode eliminar uma, várias ou todas elas com uma única linha de comando, conforme Figura 41.

Figura 41: Comando *delete*.



Fonte: Adaptado de Elmasri, 2011.

2.4 Processo eleitoral no IF Sertão Pernambucano

Conforme o parágrafo único, artigo 1º da Lei nº 11.892, os Institutos Federais de Educação, Ciência e Tecnologia possuem natureza jurídica de autarquia, detentores de autonomia administrativa, patrimonial, financeira, didático-pedagógica e disciplinar. (BRASIL, 2008). Sendo assim, tal órgão necessita de responsáveis por gerenciar e administrar os recursos frutos desse status.

Além de ser criado com a natureza de autarquia, o IF Sertão Pernambucano, assim como os demais citados no artigo 5º da Lei nº 11.892, possui uma estrutura organizacional com cargos bem definidos, cargos estes responsáveis por exercer a autonomia dada pela legislação, dentre elas a financeira, sendo alguns deles responsáveis por nomear e preencher outros cargos. (BRASIL, 2008).

2.4.1 Objetivo do processo

O processo eleitoral faz-se com o objetivo de consultar a opinião da comunidade acadêmica diretamente envolvida acerca da pessoa candidata que terá o direito de ser nomeada em um cargo eletivo na estrutura organizacional do Instituto.

2.4.2 Cargos eletivos

Nem todos os cargos da organização são preenchidos através de eleições, tal como os Pró-Reitores, nomeados pelo Reitor do Instituto Federal sem a necessidade desta ação ser objeto de consulta prévia.

2.4.2.1 Reitor

Segundo o artigo 1º do Decreto Federal nº 6.986, os Institutos Federais são dirigidos por um Reitor, nomeados pelo Presidente da República, após a indicação da comunidade escolar respectiva. (BRASIL, 2008).

2.4.2.2 Diretor-Geral

Os campi que integram cada Instituto serão dirigidos por Diretores-Gerais, nomeados pelo Reitor após realização de consulta à comunidade acadêmica dos respectivos campus. (BRASIL, 2008).

2.4.3 Local de realização e pessoas envolvidas

Conforme artigo 2º do Decreto Federal nº 6.986, as consultas para indicação dos cargos de Reitor e Diretor-Geral são realizadas nos Institutos de forma simultânea a cada quatro anos. (BRASIL, 2008).

Esse processo deverá ser concluído em até noventa dias contados da data de seu início, sendo o mesmo conduzido por uma comissão eleitoral central e comissões eleitorais de cada campus.

2.4.3.1 Comissões eleitorais

As comissões são integradas por três representantes do corpo docente, três servidores técnico-administrativos e três do corpo discente, sendo indicados, em

reunião conjunta entre elas, os integrantes da comissão central. (BRASIL, 2008).

A comissão central tem, dentre outras, as atribuições de elaborar normas de inscrição de candidatos e votação, definir cronograma do processo, credenciar fiscais, coordenar a consulta para Reitor, incluindo deliberações recursais.

Além de executar as normas definidas pela comissão central, as comissões de campus coordenam as consultas de indicação a Diretor-Geral, devendo homologar inscrições de candidatos e publicar lista de eleitores votantes.

2.4.3.2 Candidatos

Conforme parágrafo 1º, artigo 12 da Lei nº 11.892, poderão candidatar-se ao cargo de reitor os docentes pertencentes ao quadro permanente e ativo de qualquer dos campi integrantes do Instituto Federal, desde que somem ao menos cinco anos de efetivo exercício em instituição federal e possuam o título de doutor ou outros requisitos definidos na mesma lei. (BRASIL, 2008).

Enquanto o cargo de Reitor é restrito aos professores, podem se candidatar ao cargo de Diretor-Geral do campus os servidores efetivos da carreira docente ou técnico-administrativo, desde que possuam ao menos cinco anos de efetivo exercício em instituição federal e preencham outros requisitos definidos nos incisos I, II e III da Lei nº 11.892.

2.4.3.3 Eleitores

Segundo o artigo 24 do Regulamento do processo eleitoral para escolha do(a) reitor(a) e diretores(as) gerais do instituto federal sertão pernambucano:

São eleitores todos os servidores que compõem o Quadro de Pessoal Ativo Permanente da Instituição, bem como os alunos regularmente matriculados nos cursos de ensino médio, técnico, de graduação e de pós-graduação, presenciais ou à distância.

Sendo assim os mesmos se dividem em três categorias: docente, discente e técnico-administrativo, tendo os mesmos o direito de votar uma única vez no

processo eleitoral.

O eleitor que pertencer a mais de uma categoria tem o direito de votar somente pela que contenha o menor número de eleitores. Desta forma, o discente que seja matriculado em mais de um curso votará utilizando a matrícula mais antiga.

2.4.4 Consulta

Os eleitores são consultados acerca dos candidatos levando em consideração a categoria às quais pertencem, devendo comparecer no ato da votação munido de documento de identificação com foto e assinar a lista nominal de votação.

2.4.4.1 Votação

A votação ocorre em sessões eleitorais, sendo uma para cada categoria (segmento) de eleitores, iniciando às 09:00 horas e encerrando às 21:00 do mesmo dia, contendo nelas uma lista previamente publicada com todos os eleitores aptos a votar.

Conforme o artigo 37 do Regulamento do processo eleitoral para escolha do(a) reitor(a) e diretores(as) gerais do instituto federal sertão pernambucano, a votação é efetuada em cédula única diferenciada por cor e nome dos segmentos, em processo manual, na qual constam nomes dos candidatos, em ordem alfabética.

2.4.4.2 Segurança

Os eleitores registram seus votos em cédulas, cujo conteúdo prevê uma lista em ordem alfabética com os nomes dos candidatos, e em seguida depositam tal cédula em urnas também de papelão localizadas em locais isolados por cabines, de modo a manter o sigilo do voto, sendo proibido que os mesmos carreguem consigo aparelho de celular ou outro equipamento que comprometa a lisura do processo.

2.4.4.3 Apuração e resultado

Conforme o artigo 23 do Regulamento do processo eleitoral para escolha do(a) reitor(a) e diretores(as) gerais do instituto federal sertão pernambucano, os votos são apurados por categoria, atribuindo um peso de 1/3 (um terço) à razão entre votos obtidos em um segmento e quantidade de eleitores aptos neste, sendo que a soma dos resultados em cada categoria resulta na porcentagem de votos obtidos pelo candidato, conforme fórmula apresentada na Figura 42.

Figura 42: Fórmula (percentual de votos obtidos por um candidato).

II – Percentuais de votos obtidos pelo Candidato C_i :

$$C_i = \left(\frac{1}{3} \times \frac{A_i}{A_t} + \frac{1}{3} \times \frac{D_i}{D_t} + \frac{1}{3} \times \frac{E_i}{E_t} \right) \times 100$$

A_i - números de votos obtidos pelo candidato C_i no segmento técnico-administrativo;
 D_i - número de votos obtidos pelo candidato C_i no segmento docente;
 E_i - número de votos obtidos pelo candidato C_i no segmento discente;
 A_t - número total de eleitores técnico-administrativos do IF SERTÃO PE;
 D_t - número total de eleitores docentes IF SERTÃO PE;
 E_t - número total de eleitores discentes IF SERTÃO PE.

Fonte: Brasil (2012, p. 7).

Assim como a apuração de votos de um candidato por categorias, também é calculado o percentual de brancos e nulos em cada uma destas, de tal forma que o resultado da soma entre esses valores apresente o percentual total desses votos no processo eleitoral, sendo suas fórmulas apresentadas nas Figuras 43 e 44.

Figura 43: Fórmula (percentual de votos em branco).

III – Percentual de votos brancos:

$$B = \left(\frac{1}{3} \times \frac{A_b}{A_t} + \frac{1}{3} \times \frac{D_b}{D_t} + \frac{1}{3} \times \frac{E_b}{E_t} \right) \times 100$$

A_b - números de votos brancos do segmento técnico-administrativo;
 D_b - número de votos brancos do segmento docente;
 E_b - número de votos brancos do segmento discente;
 A_t - número total de eleitores técnico-administrativos IF SERTÃO PE;
 D_t - número total de eleitores docentes IF SERTÃO PE;
 E_t - número total de eleitores discentes IF SERTÃO PE.

Fonte: Brasil (2012, p. 8).

Figura 44: Fórmula (percentual de votos nulos).

IV – Percentual de votos nulos:

$$N = \left(\frac{1}{3} \times \frac{A_n}{A_t} + \frac{1}{3} \times \frac{D_n}{D_t} + \frac{1}{3} \times \frac{E_n}{E_t} \right) \times 100$$

A_n - números de votos nulos do segmento técnico-administrativo;
 D_n - número de votos nulos do segmento docente;
 E_n - número de votos nulos do segmento discente;
 A_t - número total de eleitores técnico-administrativos IF SERTÃO PE;
 D_t - número total de eleitores docentes IF SERTÃO PE;
 E_t - número total de eleitores discentes IF SERTÃO PE.

Fonte: Brasil (2012, p. 8).

Conforme parágrafo 3º do artigo 23 do regulamento do processo eleitoral para escolha do(a) reitor(a) e diretores(as) gerais do Instituto Federal do Sertão pernambucano, ao calcular o índice percentual, caso a terceira casa decimal após a vírgula contenha um número maior ou igual a 5, arredonda-se a segunda casa decimal em uma unidade a mais, descartando as demais, caso contrário a segunda casa permanecerá com o mesmo valor, descartando as demais (BRASIL, 2012).

Desta forma, consagra-se vencedor do certame quem obtiver o maior percentual de votos, sendo apresentado, ainda, no resultado, o número de abstenções de eleitores.

3 METODOLOGIA

Este trabalho nasce da necessidade de uma automação do processo eleitoral identificado em uma eleição para os cargos de Reitor e Diretor-Geral realizado no decorrer do meu curso (Gestão da Tecnologia da Informação), na qual participei como eleitor.

Identificada essa necessidade, foi realizada uma pesquisa nas diversas legislações e regulamentos previstos para este fim, na qual foram identificados os possíveis problemas, dando possibilidade de conhecê-los e assim propor uma solução para os mesmos.

Tal pesquisa pode ser classificada, quanto a sua abordagem, como qualitativa, pois segundo Deslandes (1994, p. 21) a mesma responde argumentos muito particulares, preocupando-se com um nível de realidade que não pode ou não deve ser quantificado.

Quanto aos objetivos, a pesquisa classifica-se como exploratória, sendo a mesma definida por Gil (2002, p. 41) como tendo o objetivo de proporcionar mais familiaridade com o problema, com vistas a torná-lo mais explícito ou a construir hipóteses.

Quanto ao procedimento, a mesma classifica-se como documental, visto que ela se baseia em conteúdo de documentos, pois para Lakatos (2003, p. 174), a característica da pesquisa documental é a fonte de coleta de dados e está restrita a documentos, escritos ou não.

Para a realização do projeto e efetivamente a programação do sistema com vistas a propor uma solução dos problemas identificados na pesquisa, foi definido e utilizado o método ágil Extreme Programming (XP), sendo o mesmo definido como o mais adequado para a situação.

Seguindo as orientações propostas pelo XP, antes de elaborar cada "estória" do cliente era definido um plano de testes unitários para assim validar as funcionalidades da mesma no decorrer da programação, como também era definido o teste de aceitação do usuário para validar a integração do sistema.

O uso desse método ágil teve de ser adaptado à realidade da situação, visto existir apenas um programador, indo de encontro as propostas do mesmo, pois ele

prevê uma programação em pares, além de não existir um usuário do sistema para validar cada "estória" implementada.

Para preencher essa lacuna a programação e os testes unitários eram realizados por mim, sendo em seguida, após completar a codificação, solicitado a minha esposa que a mesma fizesse uso da parte implementada enquanto eu, como projetista e programador, ficava observando e identificando eventuais problemas.

Utilizando das propostas e métodos desse método ágil, o projeto, programação, como também os testes unitários ocorriam de forma paralela, dando suporte nisso tudo as ferramentas *Microsoft Word*, *Astah Community*, *MySQL Workbench* e *NetBeans*.

3.1 Documento de requisitos

Este documento foi elaborado com vistas a obter o máximo de informações das funcionalidades desejadas e necessárias do sistema, de forma a embasar o desenvolvimento do projeto que representasse o máxima da realidade.

Deram suporte nessa etapa a *IDE NetBeans* no desenho das telas gráficas e o *Astah Community* para o desenvolvimento do diagrama de utilização e o *Microsoft Word* para a digitação e formatação do documento.

3.1.1 Requisitos

Os mesmos foram identificados e separados como funcionais e não funcionais, como também receberam uma sigla que os distingue nessas categorias, sigla essa seguida de uma numeração que o diferencia de outros requisitos da mesma categoria, conforme Figuras 45 e 46.

Figura 45: Requisito Funcional (Impedir registros duplicados) do sistema ContVoto.

<u>RF-020</u>
Nome: Impedir registros duplicados
Descrição: O sistema deverá impedir que um determinado usuário, eleitor ou candidato sejam cadastrados mais de uma vez, tomando como base de análise os documentos de identificação dos mesmos, tal como CPF, número de matrícula e título de eleitor.
Prioridade: Importante.

Fonte: O Autor.

Figura 46: Requisito Não Funcional (Somente rede local) do sistema ContVoto.

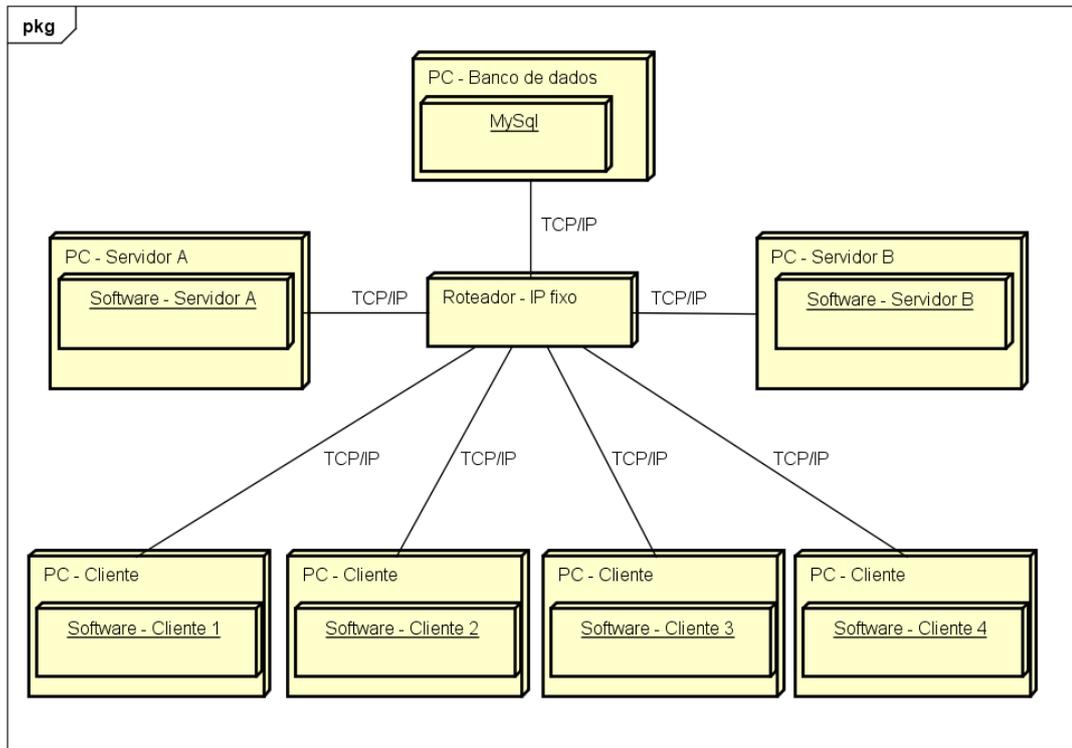
<u>RNF-09</u>
Nome: Somente rede local
Descrição: O sistema deverá ser conectado em uma rede local feita exclusivamente para troca de mensagens entre seus módulos.
Prioridade: Importante.

Fonte: O Autor.

3.1.2 Organização do sistema

Visto ser identificado que o sistema necessitaria de ser dividido em módulos e que esses módulos deveriam manter constante conectividade através de outros equipamentos, para um melhor entendimento do mesmo foi especificado a sua organização estrutural utilizando o diagrama de implantação elaborado na ferramenta *Astah Community*, conforme Figura 47.

Figura 47: Diagrama de utilização do sistema ContVoto.



powered by Astah

Fonte: O Autor.

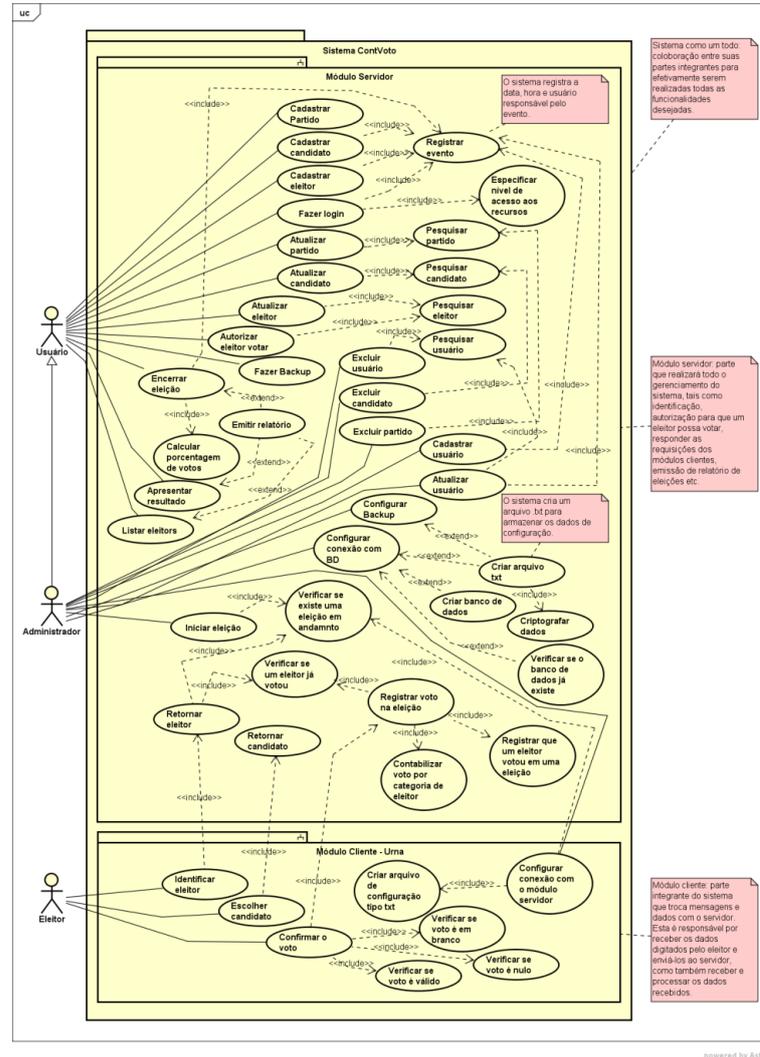
3.2 Projeto

Elaborado com vistas a propor uma solução para resolução dos problemas identificados no levantamento dos requisitos, esta fase passou por diversas etapas, etapas essas compostas por modelagens focadas em visões diferentes do sistema.

3.2.1 Casos de uso

Para esta modelagem foi utilizado o diagrama de casos de uso definido na ferramenta *Astah Community*, ferramenta esta que disponibiliza bastante recursos para a elaboração de um bom diagrama, conforme Figura 46.

Figura 48: Diagrama de casos de uso do sistema ContVoto.



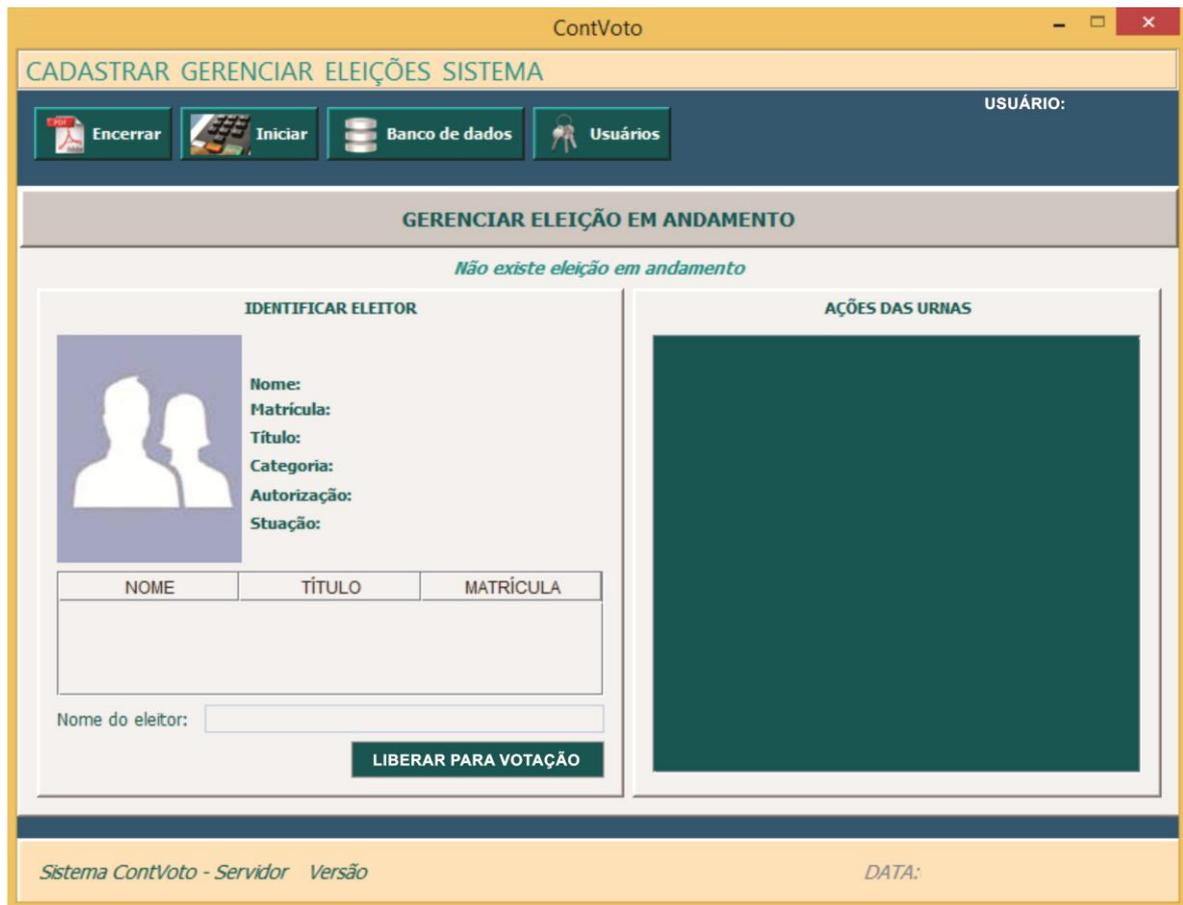
Fonte: O Autor.

3.2.2 Interfaces Gráficas

Esta camada foi criada com vistas a especificar os diversos elementos que fazem a iteração homem-máquina, definindo o processo de manipulação das funcionalidades do sistema pelo seus usuários.

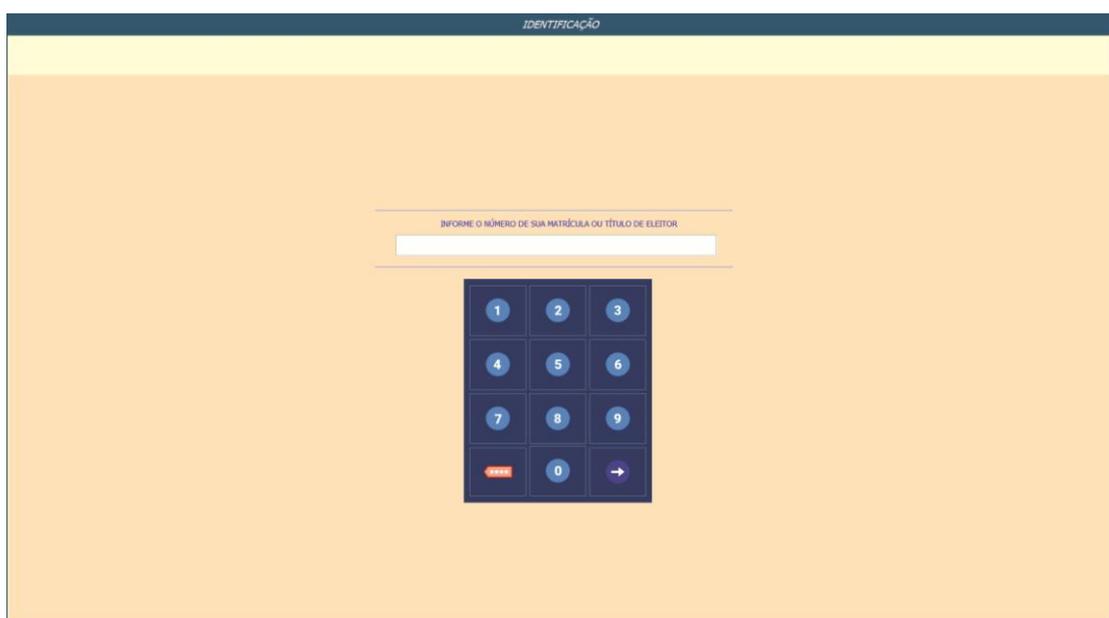
Para tanto foi utilizada a *IDE NetBeans* através dos recursos disponibilizados pela linguagem de programação Java, cujo esforço gerou várias telas, sendo mostrada algumas delas nas Figuras 49 e 50.

Figura 49: Tela principal do sistema ContVoto.



Fonte: O Autor.

Figura 50: Tela de identificação de eleitor do sistema ContVoto.



Fonte: O Autor.

3.2.3 Detalhamento dos casos de uso

Esta modelagem possibilita especificar os passos a serem percorridos no processo de manipulação do sistema. Ela tem como objetivo especificar cada caso de uso existente no diagrama de casos de uso. Para tal foi utilizado como ferramenta de suporte o *Microsoft Word*, resultando em vários detalhes, dentre eles o da Figura 51.

Figura 51: Detalhamento do caso de uso (Autoidentificação de eleitor) do sistema ContVoto.

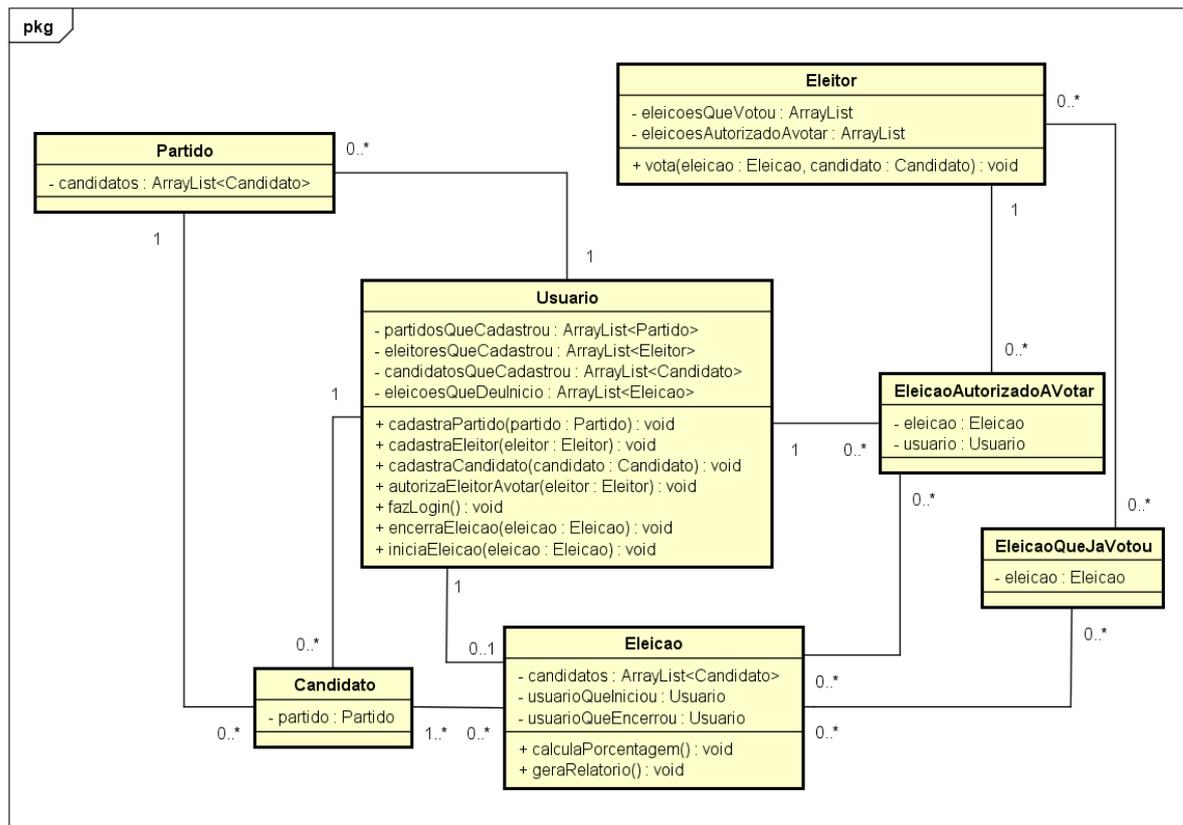
Autoidentificação de eleitor
Atores: Eleitor
Pré-condições: O sistema (cliente) encontra-se instalado na máquina, funcionando, conectado ao módulo servidor e com a tela de identificação aberta.
Pós-condições: Eleitor identificado.
Requisitos correlacionados: RF001, RF007, RF011, RF034, RF036, RF037 e RF038.
Variações Tecnológicas: Ao identificar o usuário o sistema deverá verificar se o mesmo já votou na eleição em andamento e impedi-lo de votar.
Prioridade: Importante.
Fluxo principal: 1. O ator aciona os botões do teclado. 2. O sistema preenche o campo de identificação com os dados do teclado. 3. O ator aciona o botão "->" (Fe001) (Fe002) (Fe003) (Fe004) (Fe005). 4. O sistema apresenta a tela de votação.
Fluxo de exceções: (Fe001) Campo "identificação" vazio. O sistema apresenta uma mensagem informando que o campo de identificação precisa ser preenchido e volta para o fluxo principal no passo 1. (Fe002) Sem eleição em andamento. O sistema apresenta uma mensagem informando que não existe eleição em andamento e volta para o fluxo principal no passo 1. (Fe003) Eleitor não encontrado. O sistema apresenta uma mensagem informando que o eleitor não foi encontrado e volta para o fluxo principal no passo 1. (Fe004) Eleitor não autorizado a votar. O sistema apresenta uma mensagem informando que o eleitor não está autorizado a votar na eleição em andamento e volta para o fluxo principal no passo 1. (Fe005) Eleitor já votou. O sistema apresenta uma mensagem informando que o eleitor já votou na eleição em andamento e volta para o fluxo principal no passo 1.
GUI: "Identificação".

Fonte: O Autor.

3.2.4 Classes

As classes do sistema foram definidas utilizando o diagrama de classes da linguagem UML na ferramenta *Astah Community*, conforme Figura 52.

Figura 52: Diagrama de classes do sistema ContVoto.



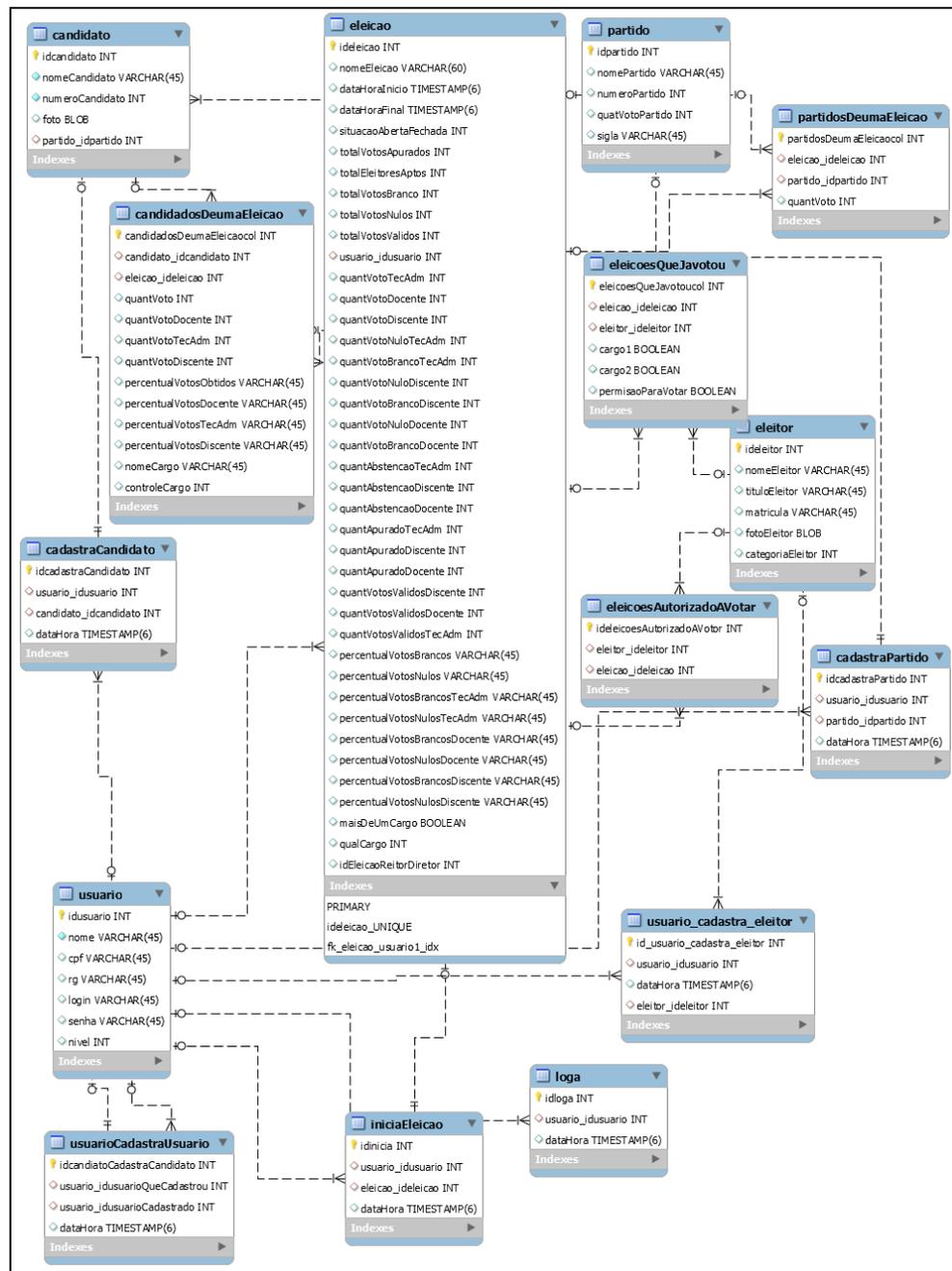
powered by Astah

Fonte: O Autor.

3.2.5 Modelo relacional

O Modelo Relacional foi criado com o auxílio do MySQL Workbench, ferramenta que possibilita a transformação automática deste modelo em códigos SQL, facilitando a implantação do banco de dados, ficando tal modelo conforme Figura 51.

Figura 53: Modelo relacional do sistema ContVoto.

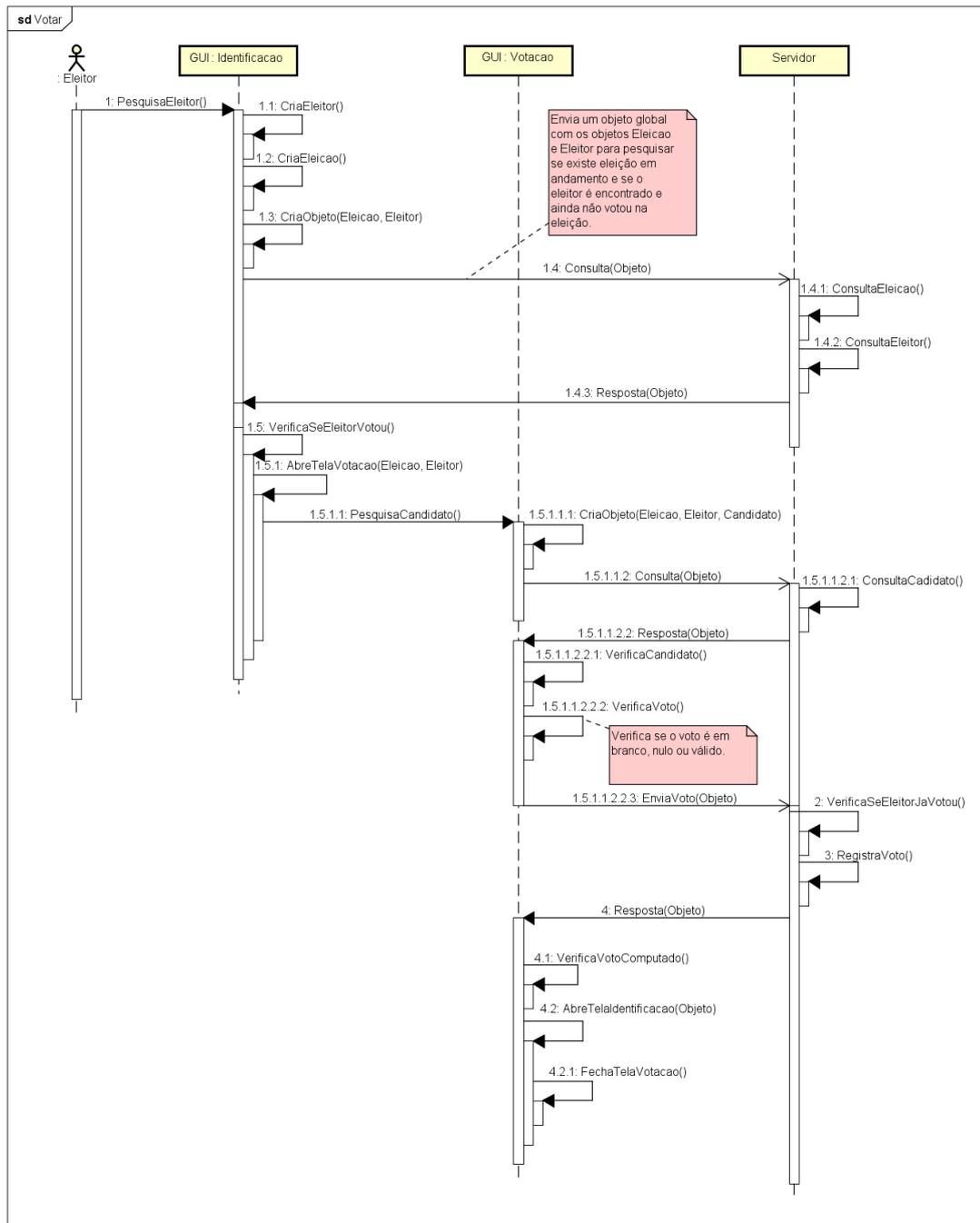


Fonte: O Autor.

3.2.6 Diagrama de sequência

Com o auxílio da ferramenta *Astah Community*, este diagrama foi desenvolvido para detalhar a comunicação de objetos do sistema no processo de manipulação do *software*, sendo gerado um para cada caso de uso definido como necessários para o bom entendimento do projeto, conforme Figura 54.

Figura 54: Diagrama de sequência (Votar) do sistema ContVoto.

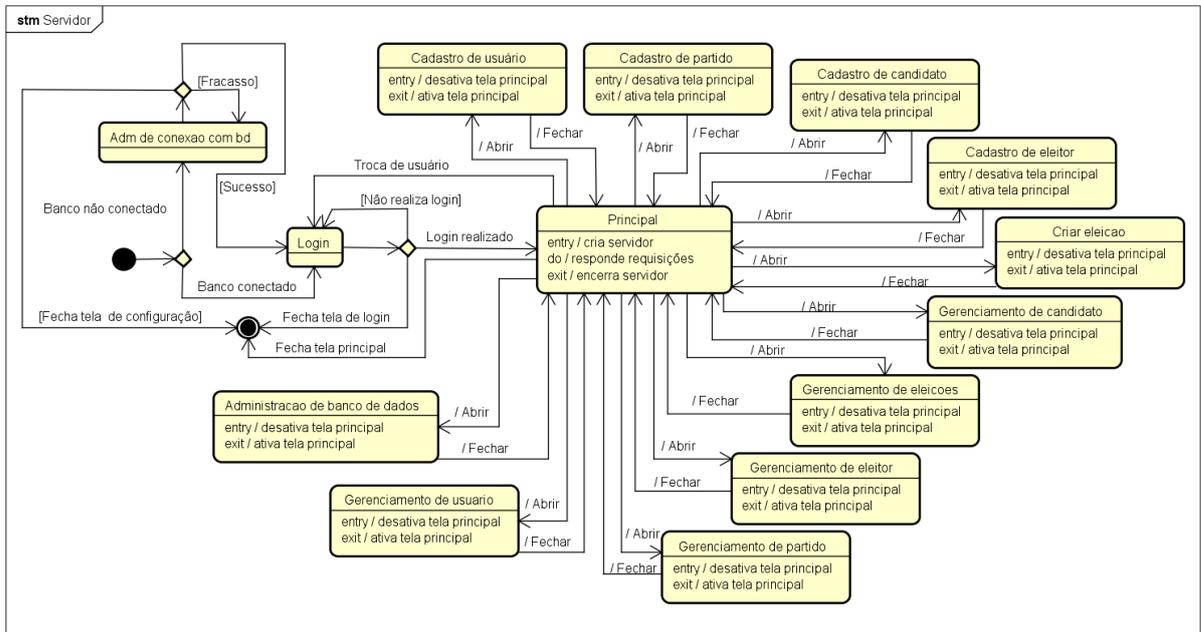


Fonte: O Autor.

3.2.7 Diagrama de máquina de estados

Também desenvolvido com a ferramenta *Astah Community*, este diagrama foi utilizado para modelar o comportamento das interfaces gráficas do sistema, resultado nos diagramas apresentados nas Figuras 55 e 56.

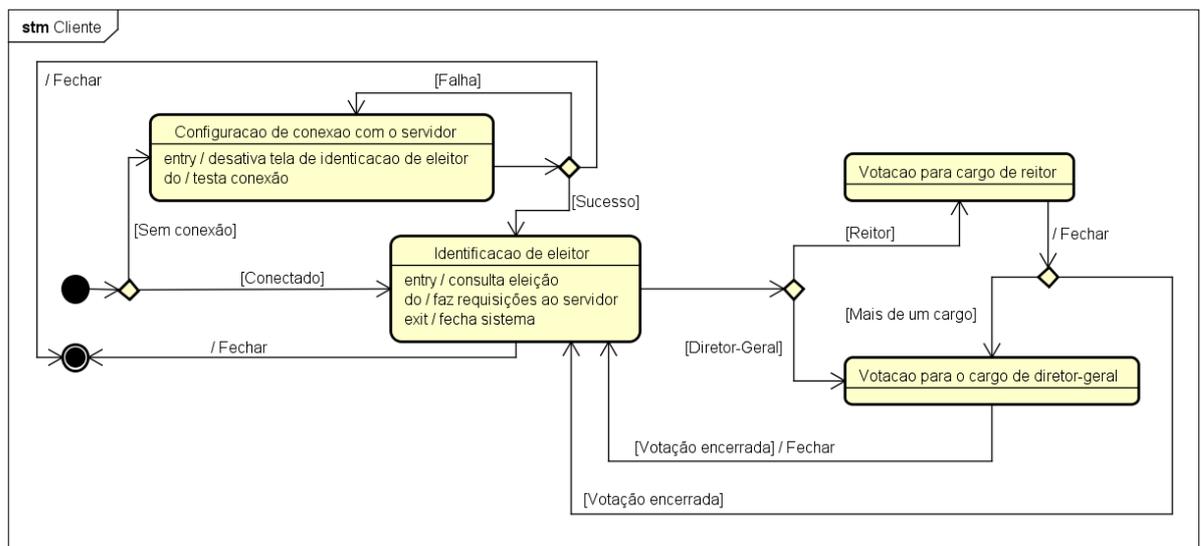
Figura 55: Diagrama de máquina de estado das interfaces gráficas do módulo Servidor - Sistema ContVoto.



powered by Astah

Fonte: O Autor.

Figura 56: Diagrama de máquina de estado das interfaces gráficas do módulo Cliente - Sistema ContVoto.



powered by Astah

Fonte: O Autor.

3.3 Implementação

A implementação do *software* foi realizada utilizando a linguagem de programação Java na IDE NetBeans e SQL no servidor MySQL para o desenvolvimento do banco de dados, como também sucessivas alterações realizadas no mesmo.

Todo esse processo ocorreu em paralelo ao desenvolvimento do projeto, pois em diversos momentos foi identificada a necessidade de atributos e métodos que não encontravam no projeto, resultando em alteração nos diagramas, código fonte e tabelas do banco de dados.

Essas alterações constantes eram facilitadas com a utilização das ferramentas (*NetBeans* e *MySQL Workbench*), simplificando todo o processo de desenvolvimento do *software*, além das duas serem perfeitamente compatíveis para a conexão entre *software* e banco de dados.

3.3.1 NetBeans

A IDE possibilita utilizar bibliotecas que não fazem parte originalmente da linguagem Java, tais como *MySQL-Connector*, *JasperReports*, e diversas outras necessárias para o desenvolvimento de um sistema profissional. Somado a isso todas as classes criadas pelo programador são organizadas de forma a facilitar a procura pelas mesmas, além de criar um arquivo executável que será aberto em qualquer computador, dispensando a utilização da IDE para abrir o sistema.

- ***MySQL-Connector***

Esta biblioteca contém o JDBC (*Java Database Connectivity*), uma API (*Application Programming Interface*) responsável por criar a ponte de conexão entre uma aplicação Java e o Gerenciador de Banco de Dados MySQL.

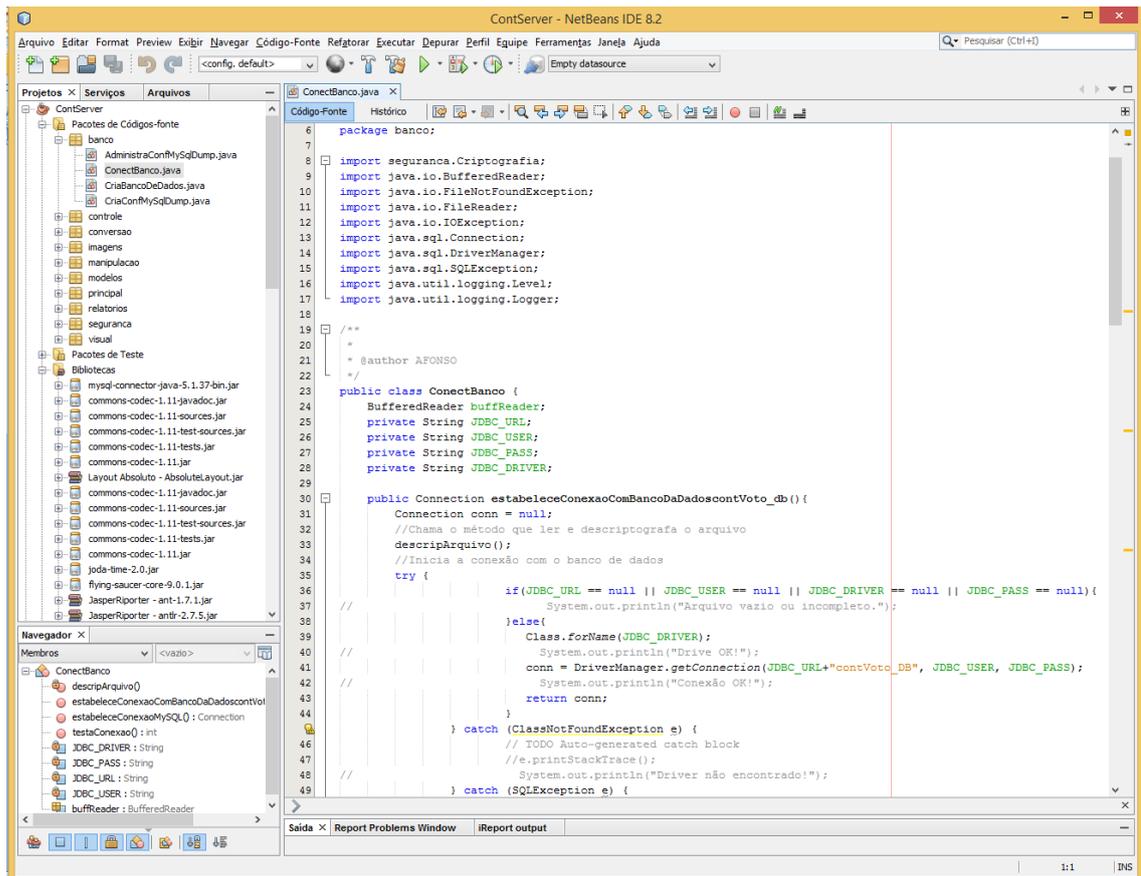
- ***JasperReports***

Desenvolvida para possibilitar a emissão de relatórios com imagens, tabelas,

listas e demais elementos necessários, esta biblioteca cria documentos em PDF compostos por dados extraídos do sistema.

Todos esses recursos, pacotes, classes e demais elementos da linguagem Java são organizados na ferramenta como mostra a Figura 57.

Figura 57: Desenvolvimento do software na IDE NetBeans.



Fonte: O Autor.

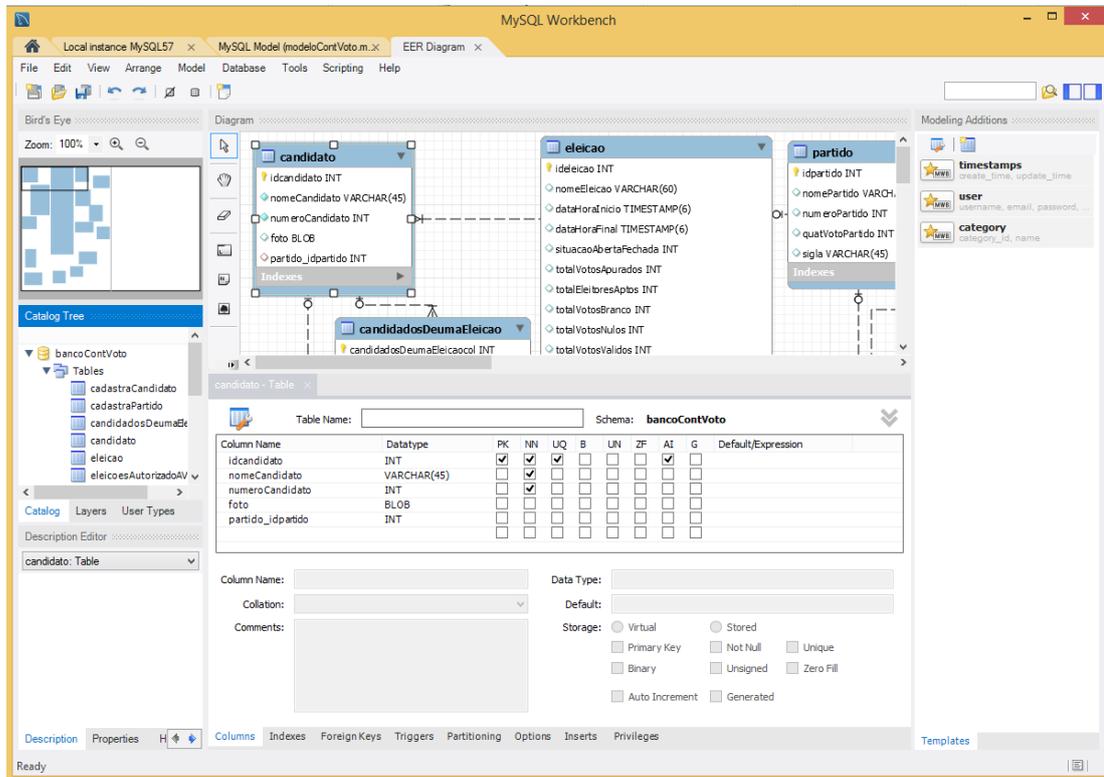
3.3.2 MySQL Workbench

Usada para manipulação de diversos bancos de dados de forma gráfica, dispensando a utilização constante de codificação SQL, esta ferramenta faz parte do MySQL Server, Gerenciador de Banco de Dados pertencente à empresa Oracle Corporation.

Através dela é possível a criação de modelos relacionais e automaticamente transformar esses modelos em um documento com códigos SQL, o qual pode ser executado no servidor e criar o banco projetado. Além disso a mesma possui

ferramentas para exclusão, inclusão de tabelas e um banco existente, como também a criação de relações entre essas tabelas. Sua estrutura é mostrada na Figura 58.

Figura 58: Gerenciamento de banco de dados no *MySQL Workbench*.



Fonte: O Autor.

4 RESULTADOS

O sistema ficou dividido em dois módulos (Servidor e Cliente), sendo o Servidor responsável por cadastros, atualizações, consultas, emissão de resultados de eleição etc., enquanto que o Cliente faz requisições com vistas a identificar eleitores, candidatos e enviar votos para serem computados no banco de dados através do Servidor.

Sendo assim, a apresentação dos resultados será dividida em duas partes, a primeira composta pelas principais telas do Servidor e em seguida as integrantes do módulo Cliente, sempre com o *software* em funcionamento e uma eleição em andamento, incluindo consulta a candidatos e registro de votos.

4.1 Servidor

4.1.1 Login

A tela *Login*, Figura 59, é a primeira a ser executada quando o *software* é aberto através de seu ícone. No entanto se o sistema não estiver devidamente conectado ao SGBD, antes dela é aberta a tela de configuração de conexão com o banco de dados. Aqui terá acesso aos recursos do *software* somente quem possuir cadastro e realize o *login*.

Figura 59: Tela "Login" do sistema ContVoto - Servidor.

Sistema ContVoto - Servidor Versão 1.03

LOGIN:

SENHA:

Por Vanderlei Afonso da Silva - (87)99628-5697

Fonte: O Autor.

4.1.2 Tela Principal

Ao efetivar a devida autenticação, o sistema abre a tela principal, Figura 60, e disponibiliza seus recursos conforme o usuário "logado" (comum ou administrador). Nela é realizado o gerenciamento de uma eleição em aberto, podendo pesquisar eleitores, comparar seu documento com suas informações e caso seja comprovada a sua autenticidade, libera o mesmo para votar. Além dessas funcionalidades e diversas outras, essa tela permite o acompanhamento de ações realizadas nos módulos clientes que encontram-se conectados.

Figura 60: Tela "principal" do sistema ContVoto - Servidor.

The screenshot displays the main interface of the ContVoto system. At the top, there is a navigation bar with the menu items 'CADASTRAR', 'GERENCIAR ELEIÇÕES', and 'SISTEMA'. Below this, there are four buttons: 'Encerrar', 'Iniciar', 'Banco de dados', and 'Usuários'. The user is logged in as 'ADMINISTRADOR: admin'. The main content area is titled 'GERENCIAR ELEIÇÃO EM ANDAMENTO' and shows the current election: 'Eleição em andamento: Reitor do IF Sertão - 2018'. The interface is split into two main sections. The left section, 'IDENTIFICAR ELEITOR', features a photo of a man, his personal details (Name: Vanderlei Afonso da Silva, Matrícula: 2289, Título: . . -, Categoria: Discente, Autorização: Não, Situação: Ainda não votou), and a table listing other voters. The table has columns for 'NOME', 'TÍTULO', and 'MATRÍCULA'. The right section, 'AÇÕES DAS URNIAS', shows a list of actions performed by the system, such as 'ELEITOR NÃO ENCONTRADO', 'ELEITOR NÃO AUTORIZADO', 'ELEITOR VOTANDO...', 'VOTO COMPUTADO', and 'ESTÁ LIVRE'. At the bottom of the left section, there is a search field for the voter's name and a 'LIBERAR PARA VOTAÇÃO' button. The footer of the window displays 'Sistema ContVoto - Servidor Versão 1.03' and the date and time 'DATA:12/09/2018 HORA: 23:45'.

NOME	TÍTULO	MATRÍCULA
Telma Feitosa Filho	65689854546454	125
Thiago Livino	13321231544688	4585
Verônica Filho	00212116546589	5654
Vanderlei Afonso da...		2289

Fonte: O Autor.

4.1.3 Cadastro de eleitor

Através do menu "Cadastrar" da tela principal é acionada a tela de cadastro de eleitor, Figura 61, interface elaborada para colher, além dos devidos dados, a imagem do eleitor armazenada no computador, transferindo-a para o banco de dados.

Figura 61: Tela de "cadastro de eleitor" do sistema ContVoto - Servidor.

Fonte: O Autor.

4.1.4 Gerenciamento de candidatos

No menu "Gerenciar" é possível acessar a tela de gerenciamento de candidatos cadastrados, possibilitando pesquisar um específico, atualizar seus dados e até mesmo excluí-lo, conforme Figura 62.

Figura 62: Tela de "gerenciamento de candidato" do sistema ContVoto - Servidor.

NOME	NOME/URNA	NÚMERO	PARTIDO	CARGO
João Mário Andrade	João	132	PT	Diretor-Geral
Lúcio Amorim Feliciano	Feliciano	451	PSDB	Diretor-Geral

Fonte: O Autor.

4.1.5 Iniciar uma eleição

Através desse módulo é possível iniciar uma eleição, definir sua data e horário de início, nome da mesma, cargos a serem consultados e candidatos participantes, sendo a tela com tais funcionalidades acionada através do menu "Eleições" ou do botão "Iniciar". A mesma é apresentada na Figura 63.

Figura 63: Tela "criar uma eleição" do sistema ContVoto - Servidor.

ContVoto

CADASTRAR GERENCIAR ELEIÇÕES SISTEMA

ADMINISTRADOR: admin

Encerrar Iniciar Banco de dados Usuários

CRIAR UMA ELEIÇÃO

Nome e início da eleição:

*Nome: Reitor do IF Sertão - 2018

*Data de início: 12/09/2018 *Hora de início: 23:45

Tipo de cargo:

Diretor-Geral Reitor

Escolha de candidatos:

Feliciano/13

João/45=Reitor

>>>

<<<

CANCELAR CRIAR

Nome d Sistema ContVoto - Servidor Versão 1.03

Sistema ContVoto - Servidor Versão 1.03 DATA:12/09/2018 HORA: 23:41

Fonte: O Autor.

4.1.6 Resultado de eleições

Através do botão "Encerrar" é possível finalizar uma eleição e emitir o relatório do seu resultado, sendo o mesmo apresentado conforme a Figura 64.

Figura 64: Resultado de uma eleição do sistema ContVoto.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SERTÃO PERNAMBUCANO Campus Floresta		SERVIÇO PÚBLICO FEDERAL MINISTÉRIO DA EDUCAÇÃO SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SERTÃO PERNAMBUCANO		
Reitor do IF Sertão - 2018				
Início: 13/09/18 23:45		Encerramento: 13/09/18 03:04		Total aptos: 21
Total nulos: 3		Total apurados: 13		Total válidos: 7
APURAÇÃO POR CATEGORIA				
	DOCENTE	TEC. ADM.	DISCENTE	TOTAL
APTOS	5	8	8	21
APURADOS	4	5	4	13
Nº BRANCOS	0	2	1	3
Nº NULOS	2	1	0	3
VÁLIDOS	2	2	3	7
ABSTENÇÕES	1	3	4	8
% BRANCOS	0,0%	8,33%	4,17%	12,5%
% NULOS	13,33%	4,17%	0,0%	17,5%
APURAÇÃO POR CANDIDATO				
Lúcio Amorim Feliciano				
	DOCENTE	TEC. ADM.	DISCENTE	TOTAL
Nº DE VOTOS OBTIDOS	0	1	1	2
PORCENTAGEM	0,0%	4,17%	4,17%	8,34%
João Mário Andrade				
	DOCENTE	TEC. ADM.	DISCENTE	TOTAL
Nº DE VOTOS OBTIDOS	2	1	2	5
PORCENTAGEM	13,33%	4,17%	8,33%	25,83%

Fonte: O Autor.

Caso a eleição em andamento foi iniciada para realizar uma consulta com os dois cargos (Reitor e Diretor-Geral), é emitido um relatório com os resultados para cada uma dessas categorias.

4.2 Cliente

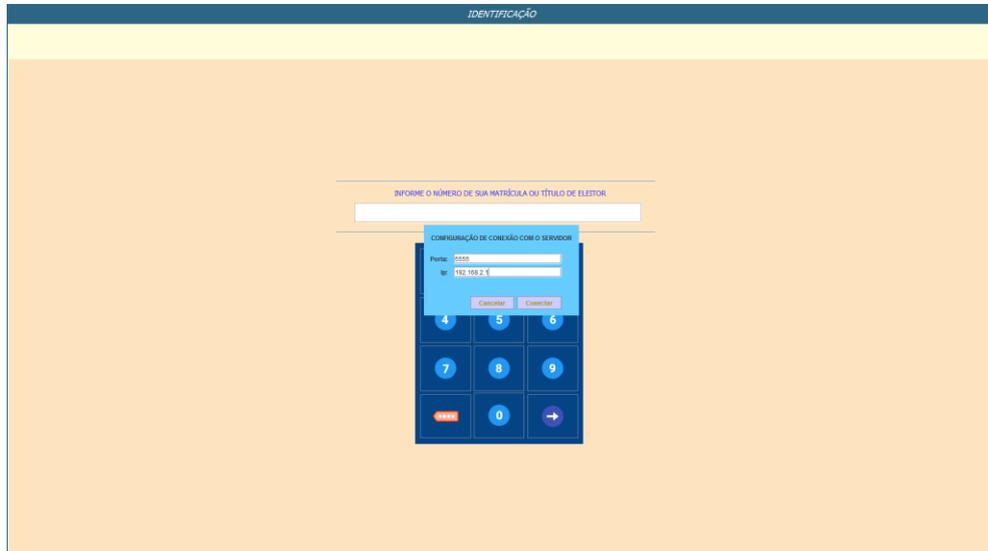
No módulo cliente é possível que o eleitor insira seu título de eleitor ou número de matrícula para que o mesmo seja identificado, como também faça a escolha de seu candidato e registre sua opção de voto.

4.2.1 Configuração de conexão com o servidor

Ao abrir esse módulo é verificada a conexão do mesmo com um módulo servidor, sendo aberta a tela de configuração, Figura 65, caso tal conexão não se

confirme. Nela são inseridos a porta de comunicação e o endereço IP do servidor.

Figura 65: Tela de configuração de conexão do sistema ContVoto - Cliente.



Fonte: O Autor.

4.2.2 Autoidentificação do eleitor

Confirmada a conexão, o sistema libera a tela de identificação, Figura 66, responsável por autenticar o eleitor para que o mesmo possa votar. Para conseguir votar o eleitor antes tem que se identificar diante do mesário no servidor, módulo responsável por fazer essa autorização.

Figura 66: Tela de identificação do sistema ContVoto - Cliente.

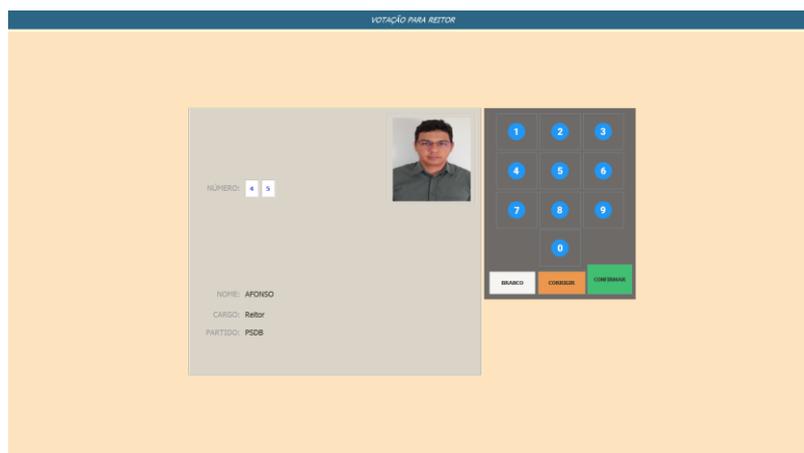


Fonte: O Autor.

4.2.3 Escolha de candidatos e registro dos votos

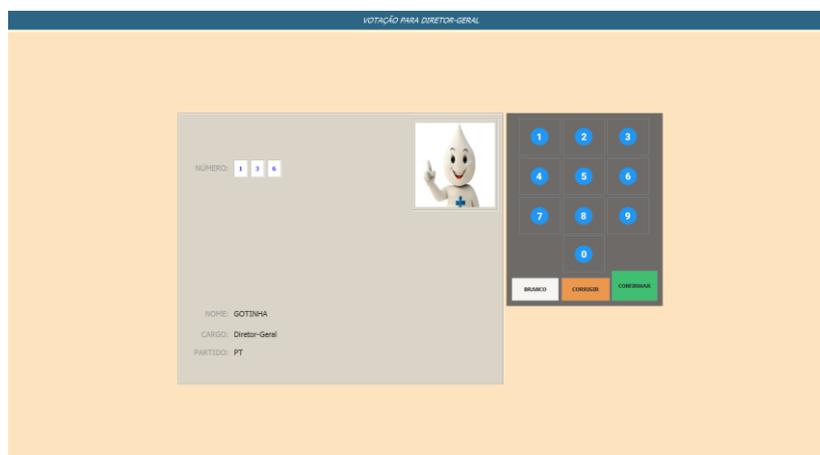
As telas de votação, Figuras 67 e 68, são acionadas através da tela de identificação após esta realizar a devida autenticação do eleitor. Antes da efetiva exibição, o sistema verifica se a eleição que encontra-se em andamento tem com objetivo realizar uma consulta para os dois cargos (Reitor e Diretor-Geral) ou apenas para um deles, sendo que na primeira opção é aberta inicialmente a tela de votação para o cargo de Reitor e em seguida esta abre a próxima. Sendo a eleição para apenas um cargo, o sistema aciona apenas a tela cuja votação é compatível com o cargo a ser consultado.

Figura 67: Tela de votação para o cargo de Reitor do sistema ContVoto - Cliente.



Fonte: O Autor.

Figura 68: Tela de votação para o cargo de Diretor-Geral do sistema ContVoto - Cliente.



Fonte: O Autor.

5 CONSIDERAÇÕES FINAIS

Com conhecimentos adquiridos no curso de GTI, me dei o objetivo de, assim como a Justiça Eleitoral Brasileira, desenvolver este sistema para facilitar o processo eleitoral do IF Sertão Pernambucano, processo esse composto por regras definidas em leis, decretos e regulamentos.

Toda a modelagem e desenvolvimento me deram a oportunidade de por em prática o que foi ministrado durante meu curso, sobre tudo nas disciplinas Engenharia e Qualidade de *Software*, Segurança e Auditoria de Sistemas, Análise e Projetos de Sistemas e em especial Programação Orientada a Objetos.

Ao término dos trabalhos senti a vontade de participar de uma equipe completa, visto todos os documentos e implementação serem realizados somente por mim, situação bem diferente do desenvolvimento profissional de *software*, cuja equipe envolve diversos profissionais com funções diferentes.

Enfim, apesar das dificuldades passadas no decorrer dos mais de 12 meses neste trabalho, sinto-me melhor como pessoa por desenvolver algo que, de certa forma, irá facilitar a vida de terceiros, e mais realizado ainda se o mesmo vier a ser evoluído por outros colaboradores, sejam eles professores ou alunos.

5.1 Trabalhos futuros

- Fazer o manual do sistema;
- Integrar o sistema a um leitor biométrico;
- Integrar o sistema a uma *webcam*, a fim de importar imagens da mesma;
- Criar um arquivo de *log* que registre todos os eventos ocorridos nas urnas;
- Criptografar todos os dados enviados ao banco de dados;
- Testar o *software* em uma eleição real.

REFERÊNCIAS

BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML**. – Rio de Janeiro: Elsevier, 2007.

BRASIL. Decreto n. 11.892, de 20 de outubro de 2008. **Regulamenta os arts. 11, 12 e 13 da Lei no 11.892, de 29 de dezembro de 2008, que institui a Rede Federal de Educação Profissional, Científica e Tecnológica e cria os Institutos Federais de Educação, Ciência e Tecnologia, para disciplinar o processo de escolha de dirigentes no âmbito destes Institutos**. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2007-010/2009/decreto/d6986.htm> Acesso em: 10 de out. 2017.

BRASIL. Lei n. 11.892, de 29 de dezembro de 2008. **Institui a Rede Federal de Educação Profissional, Científica e Tecnológica, cria os Institutos Federais de Educação, Ciência e Tecnologia, e dá outras providências**. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2007-2010/2008/lei/l11892.htm>. Acesso em: 10 de out. 2017.

BRASIL. Resolução 13 do Conselho Superior de 08 de Outubro de 2012. **Regulamento do processo eleitoral para escolha do(a) reitor(a) e diretores(as) gerais do instituto federal sertão pernambucano**. Disponível em: <<http://siteantigo.ifsertao-pe.edu.br/reitoria/images/editais/eleicao/regulamento%20final%20da%20elio%20reitor%20e%20diretores%20if%20serto%20pe%202.pdf>>. Acesso em: 10 de out. 2017.

CLARO, D.B.; SOBRAL, J.B.M.. **Programação em JAVA**. Faculdade de Educação Tecnológica do Rio de Janeiro, 2008. Disponível em: <<http://www.faeterj-rio.edu.br/downloads/bbv/0031.pdf>>. Acesso em: 07 de out. 2017. Acesso em: 15 de out. 2018.

DESLANDES, Suely Ferreira. **Pesquisa social: teoria, método e criatividade**. 21 ed. – Petrópolis, RJ: Vozes, 1994. Disponível em: <<https://wp.ufpel.edu.br/franciscovargas/files/2012/11/pesquisa-social.pdf>>.

ELMASRI, Ramez. **Sistemas de banco de dados**. – São Paulo: Pearson Addison Wesley, 2011.

GIL, Antônio Carlos.. **Como elaborar projetos de pesquisa**. 4. ed. - São Paulo: Atlas, 2002. Disponível em: <https://professores.faccat.br/moodle/pluginfile.php/13410/mod_resource/content/1/como_elaborar_projeto_de_pesquisa_-_antonio_carlos_gil.pdf>. Acesso em: 15 de out. 2018.

HEUSER, C.A.. **Projeto de banco de dados**. 4. Ed – Instituto de Informática da UFRGS, 1998. Disponível em: <http://www.fernandozaidan.com.br/pit-grad/Diversos/Livros_Disciplinas/Projeto_de_Banco_de_Dados_-_Carlos_Alberto_Heuser.pdf>. Acesso em: 07 de out. 2017.

K9 TREINAMENTOS. **Orientação a objetos em JAVA**. São Paulo, 2012. Disponível

em: <<http://www.inf.furb.br/~marcel/k19-k11-orientacao-a-objetos-em-java.pdf>>. Acesso em: 07 de out. 2017.

LAKATOS, Eva Maria. **Fundamentos de metodologia científica**. Marina de Andrade Marconi, Eva Maria Lakatos. 5. Ed. – São Paulo: Atlas, 2003. Disponível em: <https://docente.ifrn.edu.br/olivianeta/disciplinas/copy_of_historia-i/historia-ii/china-e-india>. Acesso em: 13 de out. 2018.

MARTINS, J. C. C.. **Gerenciando projetos de desenvolvimento de software com PMI, RUP e UML**. 5. ed. – Rio de Janeiro: Brasport, 2010.

NORTON, Peter. **Introdução à informática**. – São Paulo: Makron Books, 1996.

PEREIRA, Luiz Antônio de Moraes. **Análise e modelagem de sistemas com a UML: com dicas e exercícios resolvidos**. 1. Ed. – Rio de Janeiro: Luiz Antônio M. Pereira, 2011. Disponível em: <<http://www.luizantoniopereira.com.br/downloads/publicacoes/AnaliseEModelagemComUML.pdf>>. Acesso em: 15 de fev. 2018.

PRESSMAN, Roger S. **Engenharia de software [recurso eletrônico]: uma abordagem profissional**. Roger S. Pressman; tradução Ariovaldo Griesi; revisão técnica Reginaldo Arakaki, Julio Arakaki, Renato Manzan de Andrade. 7. Ed. – Dados eletrônicos – Porto Alegre: AMGH, 2011. Disponível em: <<https://docero.com.br/doc/n8e5xv>>. Acesso em: 13 de out. 2018.

SILVA, P.T.; CARVALHO, Hugo; TOREES, B.C.. **Segurança dos sistemas de informação: Gestão estratégica da segurança empresarial**. Centro Atlântico, 2003. Disponível em: <<https://docente.ifrn.edu.br/rodrigotertulino/livros/sistema-de-seguranca-da-informacao>>. Acesso em: 07 de out. 2017.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. – São Paulo: Pearson Prentice Hall, 2011.

STALLINGS, William. **Criptografia e segurança de redes**. 4. ed. – São Paulo: Pearson Prentice Hall, 2008.

TRIBUNAL REGIONAL ELEITORAL/RS. **Voto eletrônico: 10 anos da urna eletrônica; 20 anos do cadastramento eleitoral**. Porto Alegre, 2006. Disponível em: <http://www.tse.jus.br/hotsites/catalogo-publicacoes/pdf/Voto_Eletronico.pdf>. Acesso em: 06 de out. 2017.

TRIBUNAL SUPERIOR ELEITORAL. **Eleições: uma história de 500 anos**. Brasília, 2014. Disponível em: <<http://www.tse.jus.br/hotsites/catalogo-publicacoes/pdf/tse-eleicoes-no-brasil-uma-historia-de-500-anos-2014.pdf>>. Acesso em: 06 de out. 2017.

APÊNDICES

Apêndice A: Documento de requisitos

INTRODUÇÃO

Este documento especifica os requisitos do sistema (ContVoto), fornecendo aos programadores e analistas interessados em somente entendê-lo ou até mesmo aperfeiçoá-lo uma visão geral de suas funcionalidades.

Visão geral deste documento

Esta introdução fornece as informações necessárias para fazer um bom uso deste documento, explicitando seus objetivos e as convenções que foram adotadas no texto, além de conter uma lista de referências para outros documentos relacionados. As demais seções apresentam a especificação do sistema (ContVoto) e estão organizadas como descrito abaixo.

- Capítulo 1 – Descrição geral do sistema: apresenta uma visão geral do sistema, caracterizando qual é o seu escopo e descrevendo seus usuários.
- Capítulo 2 – Requisitos funcionais: especifica as funcionalidades desejadas e suas utilidades para o funcionamento geral do sistema.
- Capítulo 3 – Requisitos não funcionais: especifica todos os requisitos não funcionais do sistema, divididos em requisitos de usabilidade, confiabilidade, desempenho, segurança, distribuição, adequação a padrões e requisitos de hardware e *software*.
- Capítulo 4 – Organização física do sistema: especifica os equipamentos necessários, distribuição dos mesmos e como eles se colaboram para o funcionamento geral do sistema.

Convenções, termos e abreviações

A correta interpretação deste documento exige o conhecimento de algumas convenções e termos específicos, que são descritos a seguir.

Identificação dos Requisitos

Os requisitos estão identificados através de siglas, sendo elas:

- RF: Requisito funcional;
- RNF: Requisito não funcional.

Prioridades dos Requisitos

Para estabelecer a prioridade dos requisitos foram adotadas as denominações “essencial”, “importante” e “desejável”.

- Essencial é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.
- Importante é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.
- Desejável é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis são requisitos que podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

Documentos de referência

Documentos relacionados a esse projeto:

1. BRASIL. **Decreto n. 11.892, de 20 de outubro de 2008.** Regulamenta os arts. 11, 12 e 13 da Lei no 11.892, de 29 de dezembro de 2008, que institui a Rede Federal de Educação Profissional, Científica e Tecnológica e cria os Institutos Federais de Educação, Ciência e Tecnologia, para disciplinar o processo de escolha de dirigentes no âmbito destes Institutos. **Disponível em:** <http://www.planalto.gov.br/ccivil_03/_ato2007-2010/2009/decreto/d6986.htm>.

Acesso em: 10 de out. 2017;

2. BRASIL. **Lei n. 11.892, de 29 de dezembro de 2008.** Institui a Rede Federal de Educação Profissional, Científica e Tecnológica, cria os Institutos Federais de Educação, Ciência e Tecnologia, e dá outras providências. **Disponível em:**

<http://www.planalto.gov.br/ccivil_03/_ato2007-2010/2008/lei/l11892.htm>.

Acesso em: 10 de out. 2017;

3. **BRASIL.** Resolução 13 do Conselho Superior de 08 de Outubro de 2012. Regulamento do processo eleitoral para escolha do(a) reitor(a) e diretores(as) gerais do instituto federal sertão pernambucano. **Disponível em:**

<[http://siteantigo.ifserto-](http://siteantigo.ifserto-pe.edu.br/reitoria/images/editais/eleicao/regulamento%20final%20da%20elio%20reitor%20e%20diretores%20if%20serto%20pe%202.pdf)

pe.edu.br/reitoria/images/editais/eleicao/regulamento%20final%20da%20elio%20reitor%20e%20diretores%20if%20serto%20pe%202.pdf>. **Acesso em: 10 de out. 2017.**

4. Código-fonte: projeto inicialmente elaborado utilizando a linguagem JAVA na IDE NetBeans 8.2 e MySql Workbench por Vanderlei Afonso da Silva.

Abrangência e sistemas relacionados

O sistema aqui especificado tem como objetivo geral automatizar as eleições realizadas pelo IF Sertão Pernambucano para escolha de candidatos pré-cadastrados que aspiram a um cargo eletivo na instituição, pois tal processo é realizado utilizando urnas convencionais e cédulas de papel.

Desta forma busca-se, através deste sistema instalado em computadores, equiparar tais eleições às realizadas pela Justiça Eleitoral do Brasil, na qual os eleitores registram seus votos em equipamentos eletrônicos e a apuração dos mesmos é processado da mesma forma, dando mais simplicidade e rapidez nos trabalhos envolvidos.

Este sistema deverá possibilitar a realização de cadastro prévio de candidatos, partidos aos quais os mesmos pertençam, votos obtidos em uma determinada eleição, um módulo para identificação do eleitor e outro para que o mesmo possa escolher seu candidato e registrar seu voto etc., tudo conforme legislações federais envolvidas.

Descrição dos usuários

Este sistema é direcionado exclusivamente aos funcionários do IF Sertão Pernambucano para utilizarem nas eleições da instituição, pois o mesmo é projetado conforme suas legislações.

Esses requisitos descrevem o que o sistema deve efetivamente fazer quando manipulado pelo usuário, sendo aqui cada um deles identificado e detalhado separadamente, conforme mostra abaixo:

Grupo de Gerenciamento

RF-001

Nome: Testar conexão com o Gerenciador de Banco de Dados

Descrição: O sistema deverá analisar se o mesmo está conectado com um Gerenciador de Banco de Dados através de uma leitura de um arquivo tipo .txt com as informações de conexão com o mesmo (*login*, senha, endereço do gerenciador etc.).

Prioridade: Essencial.

RF-002

Nome: Analisar existência de banco de dados

Descrição: Após realizar a conexão com o gerenciador o sistema deverá verificar se o banco de dados do mesmo está devidamente criado.

Prioridade: Essencial.

RF-003

Nome: Criar documento de configuração

Descrição: Ao tentar ler o documento .txt com os dados de conexão com o Gerenciador de Banco de Dados o mesmo não existir, o sistema deverá disponibilizar uma tela para que o usuário possa inserir as informações e posteriormente criar tal documento com essas informações.

Prioridade: Essencial.

RF-004

Nome: Criar banco de dados

Descrição: Ao conectar com o Gerenciador de Banco de Dados o sistema identificar que o seu banco de dados não existe em tal gerenciador deverá criá-lo automaticamente sem ter a necessidade do usuário acessar o gerenciador.

Prioridade: Essencial.

RF-005

Nome: Criar usuário padrão

Descrição: Ao criar o banco de dados o sistema deverá automaticamente cadastrar um usuário do tipo administrador com *login* e senha, possibilitando assim o primeiro acesso.

Prioridade: Essencial.

RF-006

Nome: Realizar conexão entre módulos

Descrição: O sistema deverá possibilitar o envio de dados entre módulos diferentes do mesmo (cliente/servidor).

Prioridade: Essencial.

RF-007

Nome: Verificar conexão entre módulos

Descrição: O sistema deverá verificar constantemente se suas partes estão devidamente conectadas (cliente conectado ao servidor).

Prioridade: Essencial.

Grupo de Segurança

RF-008

Nome: Criptografar dados

Descrição: O sistema deverá criptografar os dados antes dos mesmos serem inseridos em documentos .txt, como também descriptografá-los quando eles forem devidamente lidos.

Prioridade: Importante.

RF-009

Nome: Backup

Descrição: O sistema deverá possibilitar que o usuário consiga fazer cópia de sua base de dados, como também restaurá-lo quando desejar.

Prioridade: Importante.

RF-010

Nome: Nível de acesso

Descrição: O sistema deverá identificar o usuário que está acessando e quais recursos o mesmo está autorizado a manipular, bloqueando-os conforme o caso.

Prioridade: Essencial.

RF-011

Nome: Identificação de eleitor no servidor

Descrição: O sistema deverá identificar o eleitor no ato de seu comparecimento (por seu nome), apresentando suas informações quando encontrado (título de eleitor, matrícula, nome, foto etc.).

Prioridade: Essencial.

RF-012

Nome: Autorização para votar

Descrição: O sistema deverá, quando devidamente identificado o eleitor, autorizá-lo a votar na eleição em que se encontra em andamento.

Prioridade: Essencial.

RF-013

Nome: Bloqueio de voto

Descrição: O sistema deverá bloquear qualquer tentativa de voto que não esteja autorizado para tal ou que já tenha votado na eleição em que estiver em andamento.

Prioridade: Essencial.

Grupo de Configuração

RF-014

Nome: Conexão com banco de dados

Descrição: O sistema, em seu módulo servidor, deverá possibilitar que um administrador insira as informações de conexão com o Gerenciador de Banco de Dados (login, senha, caminho etc.) guardando-as de forma persistente para serem acessadas sempre que necessário.

Prioridade: Essencial.

RF-015

Nome: Conexão com o módulo servidor

Descrição: O sistema, em seu módulo cliente, deverá possibilitar que um administrador insira as informações de conexão com o servidor (porta, IP etc.), guardando-as de forma persistente para que possam ser utilizadas quando necessário.

Prioridade: Essencial.

RF-016

Nome: *Backup do MySQL*

Descrição: O sistema, em seu módulo servidor, deverá possibilitar que um administrador insira as configurações de Backup do *MySQL*, possibilitando que os usuário consigam realizar as tarefas de cópia e restauração da base de dados.

Prioridade: Importante.

Grupo de controle

RF-017

Nome: Somente letras

Descrição: O sistema deverá impedir que o usuário insira números onde o mesmo só necessite de letras.

Prioridade: Importante.

RF-018

Nome: Somente números

Descrição: O sistema deverá impedir que o usuário insira letras onde o mesmo só necessite de números.

Prioridade: Importante.

RF-019

Nome: Formatação de campos

Descrição: O sistema deverá formatar automaticamente os dados inseridos pelo usuário de forma que os mesmos sejam iguais aos utilizados no mundo real (CPF, RG, número de candidato etc).

Prioridade: Importante.

RF-020

Nome: Impedir registros duplicados

Descrição: O sistema deverá impedir que um determinado usuário, eleitor ou candidato sejam cadastrados mais de uma vez, tomando como base de análise os documentos de identificação dos mesmos, tal como CPF, número de matrícula e título de eleitor.

Prioridade: Importante.

RF-021

Nome: Registrar voto por categoria

Descrição: O sistema deverá registrar os votos de um determinado candidato em uma determinada eleição pela categoria à qual o eleitor votante pertença.

Prioridade: Importante.

RF-022

Nome: Voto somente em um candidato por categoria

Descrição: O sistema deverá impedir que o eleitor vote em mais de um candidato por categoria (Diretor e Reitor) em uma eleição, possibilitando que o mesmo deposite seu voto no candidato desejado em cada categoria.

Prioridade: Importante.

RF-023

Nome: Quantidade de categoria por eleição

Descrição: O sistema deverá possibilitar que o administrador do sistema possa escolher se uma determinada eleição será realizada para escolha de candidatos de uma única categoria ou das duas existentes (Reitoria e Diretoria).

Prioridade: Importante.

RF-024

Nome: Controle de cadastro

Descrição: O sistema deverá registrar qual usuário fez um determinado cadastro no sistema, como também a data e horário dessa ocorrência.

Prioridade: Importante.

RF-025

Nome: Controle de autorização

Descrição: O sistema deverá registrar qual usuário autorizou um determinado eleitor a votar, registrando somente a data dessa ocorrência.

Prioridade: Importante.

RF-026

Nome: Registro de voto

Descrição: O sistema deverá registrar que um determinado eleitor votou em uma determinada categoria em uma determinada eleição, sendo que após o mesmo votar em todas devesa ser registrado que o mesmo já votou na eleição em questão.

Prioridade: Importante.

RF-027

Nome: Candidato tem que pertencer a um partido

Descrição: O sistema deverá impedir que um candidato seja cadastrado sem pertencer a um determinado partido.

Prioridade: Importante.

RF-028

Nome: Candidato de um partido não concorre com outro do mesmo

Descrição: O sistema deverá impedir que um candidato seja concorrente de outro quando ambos façam parte de um mesmo partido.

Prioridade: Importante.

RF-029

Nome: Número único

Descrição: O sistema deverá impedir que um candidato tenha seu número igual a outro.

Prioridade: Importante.

RF-030

Nome: Padrão de número

Descrição: O sistema deverá definir padrões de números dos candidatos por categoria, de tal forma que o eleitor possa identificar que um padrão refere-se uma determinada categoria.

Prioridade: Importante.

RF-031

Nome: Número de um partido

Descrição: O sistema deverá definir um número para cada partido, sendo esse composto por dois algarismos.

Prioridade: Importante.

RF-032

Nome: Impedir número de candidato diferente do partido

Descrição: O sistema deverá impedir que um candidato tenha um número diferente do número de seu partido.

Prioridade: Importante.

RF-033

Nome: Registrar *login*

Descrição: O sistema deverá registrar quem, data e horário em que um determinado usuário acessou o sistema.

Prioridade: Importante.

RF-034

Nome: Verificar se existe uma eleição em andamento

Descrição: O sistema deverá analisar se existe uma eleição em andamento, impedindo que outra seja iniciada sem que esta seja encerrada.

Prioridade: Importante.

RF-035

Nome: Identificar quem encerrou uma eleição

Descrição: O sistema deverá registrar qual usuário, data e horário em que uma eleição foi encerrada.

Prioridade: Importante.

RF-036

Nome: Informar estados do módulo cliente

Descrição: O sistema, em seu módulo de servidor, deverá disponibilizar de mecanismos que o usuário (mesário) consiga identificar quando um dos módulos clientes (urnas) estão em votação (com eleitor votando), tal como informar quando foi desocupada (votação encerrada).

Prioridade: Desejável.

RF-037

Nome: Informar quando um eleitor não autorizado tentar votar

Descrição: O sistema, em seu módulo de servidor, deverá disponibilizar de mecanismos que o usuário (mesário) consiga identificar quando um eleitor está tentando votar sem autorização para tal.

Prioridade: Desejável.

RF-038

Nome: Informar quando um eleitor tentar votar pela segunda vez

Descrição: O sistema, em seu módulo de servidor, deverá disponibilizar de mecanismos que o usuário (mesário) consiga identificar quando um eleitor está tentando votar pela segunda vez em um dos módulos clientes (urnas).

Prioridade: Desejável.

Grupo de cadastro**RF-039**

Nome: Cadastro de partido

Descrição: O sistema deverá possibilitar o cadastro de partidos contendo os dados que os mesmos passam ser identificados (sigla, nome e número).

Prioridade: Importante.

RF-040

Nome: Cadastro de candidato

Descrição: O sistema deverá possibilitar o cadastro de candidatos com os seguintes dados: foto, nome completo, nome que aparecerá na urna, partido, categoria à qual concorrerá e seu número.

Prioridade: Importante.

RF-041

Nome: Cadastro de eleitor

Descrição: O sistema deverá possibilitar o cadastro de eleitores contendo os seguintes dados: foto, nome completo, título de eleitor, matrícula e categoria à qual o mesmo pertence.

Prioridade: Importante.

RF-042

Nome: Cadastro de usuário

Descrição: O sistema deverá possibilitar o cadastro de usuário para a manipulação do mesmo com as seguintes informações: nome, cpf, rg, *login*, senha e sua categoria, podendo ser usuário ou administrador.

Prioridade: Importante.

RF-043

Nome: Iniciar eleição

Descrição: O sistema deverá possibilitar o administrador inicie uma eleição, definindo seu nome, data e horário de início, quantas e quais categorias, como também seus candidatos.

Prioridade: Importante.

Grupo de atualização

RF-044

Nome: Atualizar dados de partido

Descrição: O sistema deverá possibilitar a atualização dos dados de um determinado partido.

Prioridade: Importante.

RF-045

Nome: Atualizar dados de candidato

Descrição: O sistema deverá possibilitar a atualização dos dados de um determinado candidato.

Prioridade: Importante.

RF-046

Nome: Atualizar dados de eleitor

Descrição: O sistema deverá possibilitar a atualização dos dados de um determinado eleitor.

Prioridade: Importante.

RF-047

Nome: Atualizar dados de usuário

Descrição: O sistema deverá possibilitar a atualização dos dados de um determinado usuário.

Prioridade: Importante.

Grupo de exclusão

RF-048

Nome: Excluir partido

Descrição: O sistema deverá possibilitar a exclusão de um determinado partido.

Prioridade: Importante.

RF-049

Nome: Excluir candidato

Descrição: O sistema deverá possibilitar a exclusão de um determinado candidato.

Prioridade: Importante.

RF-050

Nome: Excluir eleitor

Descrição: O sistema deverá possibilitar a exclusão de um determinado eleitor.

Prioridade: Importante.

RF-051

Nome: Excluir usuário

Descrição: O sistema deverá possibilitar a exclusão de um determinado usuário.

Prioridade: Importante.

Grupo de votação

RF-052

Nome: Autoidentificação de eleitor

Descrição: O sistema, em seu módulo de cliente, deverá dispor de uma tela na qual o eleitor irá digitar sua identificação (título de eleitor ou matrícula), buscando na base de dados através do módulo servidor se o mesmo está autorizado a votar.

Prioridade: Importante.

RF-053

Nome: Identificação de candidato

Descrição: O sistema, em seu módulo de cliente, deverá dispor de uma tela na qual o eleitor irá digitar inserir o número de seu candidato, sendo retornado, caso o mesmo seja encontrado, todos os dados do mesmo e apresentados na mesma tela.

Prioridade: Importante.

RF-054

Nome: Registro de voto em branco

Descrição: O sistema deverá dispor que o candidato possa registrar sem voto em branco, como também registrar o somatório desse tipo de voto em uma determinada eleição por categoria de eleitor.

Prioridade: Importante.

RF-055

Nome: Registro de voto nulo

Descrição: O sistema deverá registrar os votos que se categorizem como nulo, contabilizando os mesmos por cada categoria de eleitor.

Prioridade: Importante.

RF-056

Nome: Registro de votos válidos

Descrição: O sistema deverá registrar os votos válidos em uma determinada eleição por categoria de eleitor, como também o somatório de todos eles.

Prioridade: Importante.

RF-057

Nome: Registro de faltosos

Descrição: O sistema deverá registrar a quantidade de eleitores que faltaram a eleição (abstenção).

Prioridade: Importante.

RF-058

Nome: Cálculo de porcentagem

Descrição: O sistema deverá calcular a porcentagem de votos nulos, em branco e válidos por cada categoria de eleitor, como também a porcentagem obtida por candidato.

Prioridade: Importante.

RF-059

Nome: Emissão de lista com eleitores aptos a votar

Descrição: O sistema deverá possibilitar que seja emitida uma lista com todos os eleitores divididos por categoria que estão aptos a votar.

Prioridade: Importante.

RF-060

Nome: Emissão de resultado

Descrição: O sistema deverá possibilitar que seja emitido um relatório de uma determinada eleição contendo o resultado da mesma, facilitando assim a identificação dos ganhadores.

Prioridade: Importante.

Grupo Usabilidade**RNF-01**

Nome: Fácil utilização e entendimento

Descrição: O sistema deve ser intuitivo, dispensando um grande estudo para entendê-lo, sendo obrigatório que os registros possam ser realizados em poucas telas.

Prioridade: Importante.

RNF-02

Nome: O sistema deve ser projetado para ser utilizado somente em desktop

Descrição: O sistema deverá ser implementado para rodar apenas em computadores de mesa.

Prioridade: Importante.

Grupo Confiabilidade**RNF-03**

Nome: O sistema deve fazer cópias de segurança

Descrição: O sistema deverá realizar cópias de segurança da base de dados, como também restauração da mesma a partir de uma cópia.

Prioridade: Importante.

RNF-04

Nome: Mensagens inelegíveis entre módulos

Descrição: O sistema deverá trocar mensagens entre seus módulos de forma que seus conteúdos não sejam captados e interpretados.

Prioridade: Importante.

Desempenho**RNF-05**

Nome: Tempo de resposta

Descrição: O sistema deverá retornar as respostas de consultas em menos de 5 segundo.

Prioridade: Essencial.

RNF-06

Nome: Múltiplos clientes

Descrição: O servidor deverá suportar vários clientes conectados (várias urnas), respondendo as requisições dos mesmos.

Prioridade: Essencial.

RNF-07

Nome: Pequenas imagens

Descrição: O sistema deverá limitar o tamanho da foto de um candidato ou eleitor cadastrado, buscando economizar espaço no banco de dados e otimizar a resposta do mesmo ao carregá-las.

Prioridade: Essencial.

Segurança

RNF-08

Nome: Acesso ao sistema

Descrição: O acesso ao sistema só será possível por meio de autenticação de pessoas devidamente cadastradas.

Prioridade: Essencial.

RNF-09

Nome: Somente rede local

Descrição: O sistema deverá ser conectado em uma rede local feita exclusivamente para troca de mensagens entre seus módulos.

Prioridade: Importante.

RNF-10

Nome: Criptografia

Descrição: Os dados contidos nos sistema serão criptografados e liberados somente para pessoas autorizadas.

Prioridade: Essencial.

Distribuição

RNF-11

Nome: Cópias do sistema

Descrição: O sistema será disponibilizado exclusivamente para as eleições realizadas pelo Instituto Federal de Educação do Sertão Pernambucano.

Prioridade: Importante.

Hardware e *software*

RNF-12

Nome: Monitor sensível ao toque

Descrição: O sistema necessitará de ser executado em dois computadores (módulo servidor e cliente), sendo necessário um monitor sensível ao toque no PC que rodar o módulo cliente.

Prioridade: Importante.

RNF-13

Nome: Roteador

Descrição: Os computadores com os módulos do sistema deverão estar conectados através de um roteador (rede local) para devida conectividade entre os mesmos.

Prioridade: Importante.

RNF-14

Nome: Programação em JAVA

Descrição: O sistema deverá ser desenvolvido através desta linguagem por a mesma ser multi plataforma, se obtendo uma compatibilidade entre vários Sistemas Operacionais.

Prioridade: Desejável.

RNF-15

Nome: IDE *NetBeans*

Descrição: Será utilizado a IDE *NetBeans* para a codificação do sistema.

Prioridade: Desejável.

RNF-16

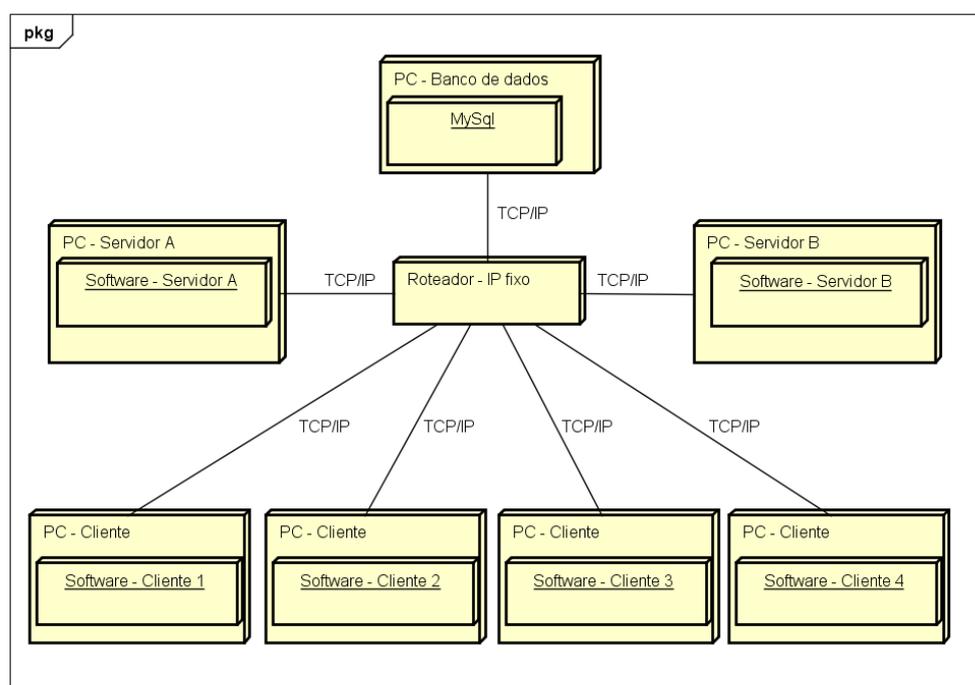
Nome: Banco de dados *MySQL*

Descrição: Por ser uma ferramenta muito eficiente e grátis, não gerando custo para o projeto.

Prioridade: Desejável.

Neste capítulo é detalhado a organização física do sistema, ou seja, a distribuição dos equipamentos envolvidos para a devida funcionalidade do mesmo. Para isso foi utilizado o diagrama de utilização, diagrama da linguagem UML que especifica de forma gráfica os equipamentos, como também a relação entre os mesmos, possibilitando um melhor entendimento do objetivo geral desejado.

Diagrama de utilização



powered by Astah

Detalhes do diagrama

O módulo servidor do sistema poderá rodar em quantos computadores forem necessários, compartilhando uma mesma base de dados, estrutura que possibilita a utilização de dois ou mais pontos de votação, sendo cada um deles operado por um responsável (mesário).

Cada ponto de votação poderá conter um PC Servidor com vários PCs clientes, sendo estes responsáveis por receber os dados inseridos pelo eleitor e

enviá-los para aquele, enquanto que o servidor, operado pelo mesário, fará as consultas e registros na base de dados.

A base de dados poderá está localizado em um computador distinto do módulo servidor como pertencer a este, dependendo da necessidade no momento da organização do evento, situação ligada a quantidade de pontos de votação necessários ou outros fatores.

O servidor de banco de dados poderá está dentro das dependências do local do evento (rede local) ou instalado em um servidor na Internet, possibilitando o registro de votação realizadas em vários institutos ao mesmo tempo.

Apêndice B: Projeto

Histórico das Revisões

Data	Versão	Descrição	Autor
27/11/2016	1.0	Criação do diagrama de casos de uso	Vanderlei Afonso
27/11/2016	1.0	Criação das telas gráficas	Vanderlei Afonso
27/11/2016	1.0	Detalhamento dos casos de uso	Vanderlei Afonso
28/11/2016	1.01	Atualização dos casos de uso	Vanderlei Afonso
01/12/2017	1.02	Atualização das telas gráficas	Vanderlei Afonso
01/08/2018	1.02	Atualização dos casos de uso	Vanderlei Afonso
10/08/2018	1.03	Atualização das telas gráficas	Vanderlei Afonso
17/08/2018	1.03	Atualização das telas gráficas	Vanderlei Afonso
17/08/2018	1.03	Atualização dos detalhes de casos de uso	Vanderlei Afonso
19/08/2018	1.03	Atualização dos casos de uso	Vanderlei Afonso
22/08/2018	1.03	Atualização dos casos de uso	Vanderlei Afonso

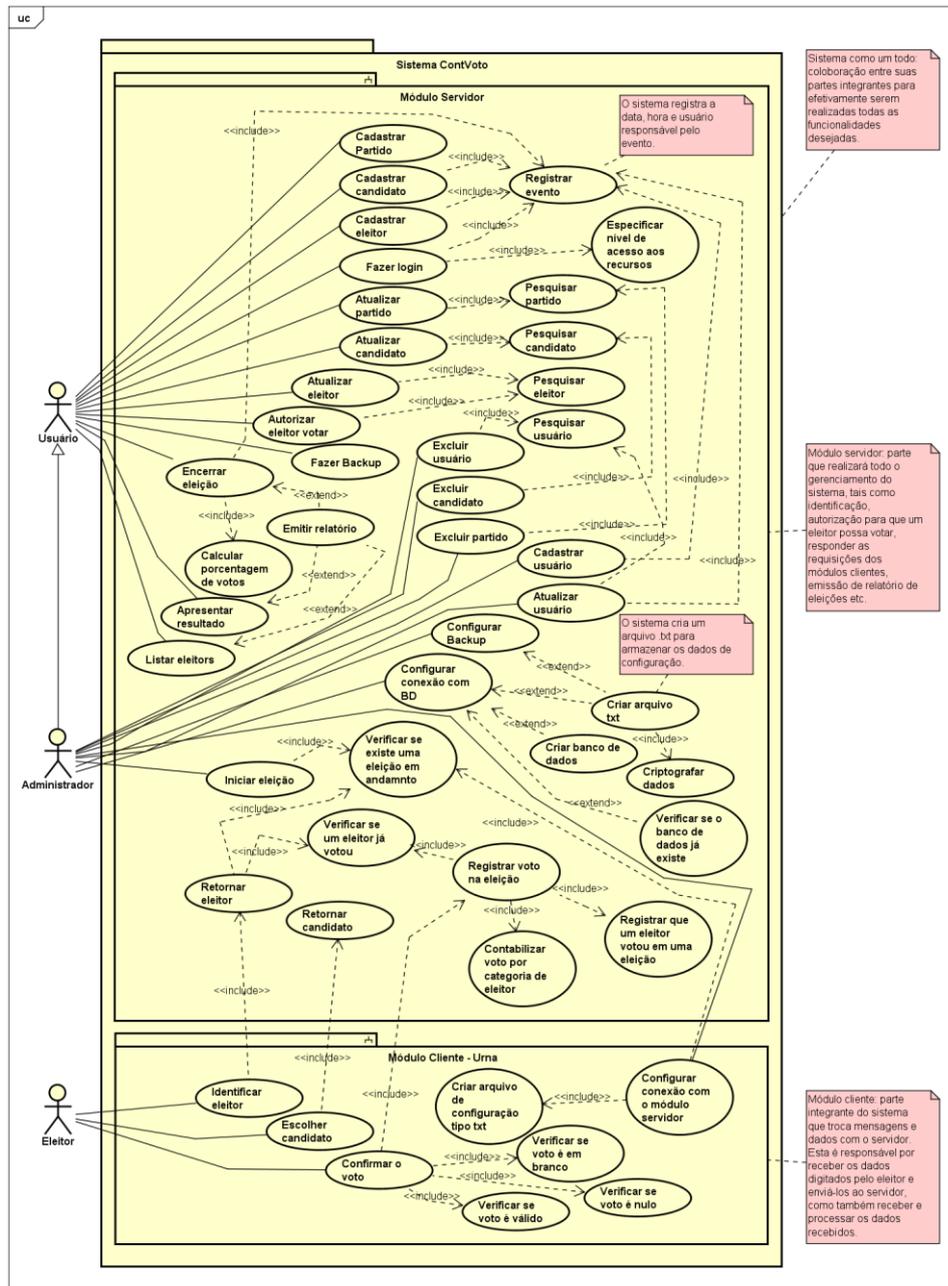
APRESENTAÇÃO

Este documento tem como objetivo, em primeiro plano, descrever de forma gráfica a relação de eventuais manipuladores do sistema, e posteriormente detalhar os passos percorridos e como essas ações são realizadas.

Este documento foi elaborado com base nos dados extraídos do documento de requisitos do sistema em questão, visto o mesmo propor um entendimento inicial das funcionalidades pretendidas.

DIAGRAMA DE CASOS DE USOS

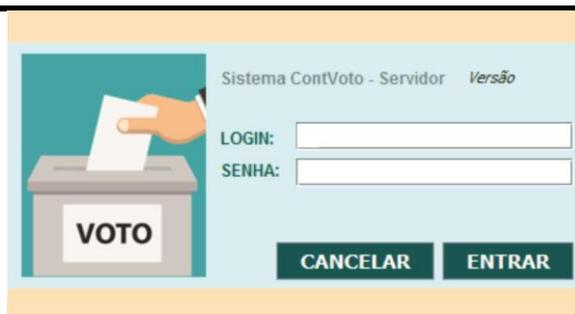
Este diagrama oferece um entendimento geral da relação dos manipuladores do sistema com as funcionalidades do mesmo, conforme levantamentos realizados e registrados no documento de requisitos.



INTERFACE GRÁFICA DO USUÁRIO (GUI)

Graphical User Interface (Interface Gráfica do Usuário, em português), é um conjunto de gráficos do sistema em que o usuário tem contato e os manipula, mantendo assim uma interação entre homem e máquina.

Login



Sistema ContVoto - Servidor *Versão*

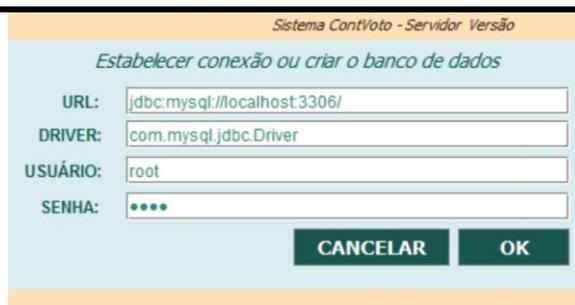
 VOTO

LOGIN:

SENHA:

CANCELAR **ENTRAR**

Administrar conexão do banco de dados



Sistema ContVoto - Servidor *Versão*

Estabelecer conexão ou criar o banco de dados

URL:

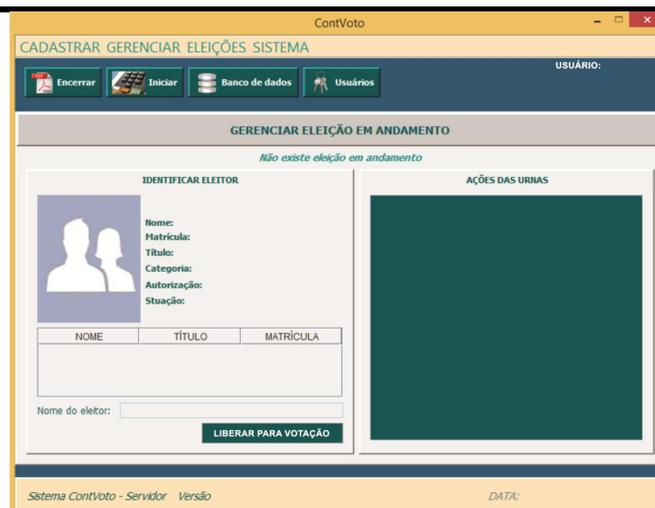
DRIVER:

USUÁRIO:

SENHA:

CANCELAR **OK**

Principal



ContVoto

CADASTRAR GERENCIAR ELEIÇÕES SISTEMA

Encerrar Iniciar Banco de dados Usuários

USUÁRIO:

GERENCIAR ELEIÇÃO EM ANDAMENTO

Não existe eleição em andamento

IDENTIFICAR ELEITOR

 Nome:
Matrícula:
Título:
Categoria:
Autorização:
Situação:

NOME	TÍTULO	MATRÍCULA
Nome do eleitor: <input type="text"/>		

LIBERAR PARA VOTAÇÃO

AÇÕES DAS URIAS

Sistema ContVoto - Servidor *Versão* DATA:

Administração de banco de dados (cópia)

The screenshot shows a window titled "GERENCIAMENTO DE BANCO DE DADOS" with a close button (X) in the top right corner. Below the title bar are three tabs: "Cópia de segurança" (selected), "Restauração", and "Configurar MySql". The main area contains a label "Local:" followed by a text input field and a browse button "...". At the bottom of the main area are two buttons: "VOLTAR" and "GERAR CÓPIA". The footer of the window reads "Sistema ContVoto - Servidor Versão".

Administração de BD (restauração)

The screenshot shows the same window as above, but with the "Restauração" tab selected. The main area contains a label "Local:" followed by a text input field and a browse button "...". At the bottom of the main area are two buttons: "VOLTAR" and "RESTAURAR". The footer of the window reads "Sistema ContVoto - Servidor Versão".

Administração de BD (configuração de backup)

The screenshot shows the same window as above, but with the "Configurar MySql" tab selected. The main area contains the text "ADICIONAR CAMINHO DO MYSQLDUMP" above a label "Local:" followed by a text input field and a browse button "...". At the bottom of the main area are two buttons: "VOLTAR" and "SALVAR". The footer of the window reads "Sistema ContVoto - Servidor Versão".

Cadastro de usuário

The screenshot shows a window titled "CADASTRO DE USUÁRIO" with a close button (X) in the top right corner. The main area contains several input fields: "*Nome:", "*CPF:", "RG:", "*Categoria:" (with a dropdown arrow), "*Login:", and "*Senha:". At the bottom of the main area are two buttons: "CANCELAR" and "SALVAR". The footer of the window reads "Sistema ContVoto - Servidor Versão".

Cadastro de partido

The screenshot shows a window titled "CADASTRO DE PARTIDO" with a close button (X) in the top right corner. The form contains three input fields: "Sigla:", "Nome:", and "Número:". Below the fields are two buttons: "CANCELAR" and "SALVAR". At the bottom of the window, the text "Sistema ContVoto - Servidor Versão" is displayed.

Cadastro de candidato

The screenshot shows a window titled "CADASTRO DE CANDIDATO" with a close button (X) in the top right corner. The form contains several input fields: "*Nome completo:", "*Nome na urna:", "*Partido:" (with a dropdown arrow), "*Cargo:" (with a dropdown arrow), and "*Número:". To the right of these fields is a "Foto" section with a large empty box and an "INSERIR" button below it. Below the main form area are two buttons: "CANCELAR" and "SALVAR". At the bottom of the window, the text "Sistema ContVoto - Servidor Versão" is displayed.

Cadastro de eleitor

The screenshot shows a window titled "CADASTRO DE ELEITOR" with a close button (X) in the top right corner. The form contains four input fields: "*Nome:", "Título de eleitor:", "*Matrícula:", and "*Categoria:" (with a dropdown arrow). To the right of these fields is a "Foto" section with a placeholder image of two people. Below the main form area are four buttons: "LIMPAR", "CANCELAR", "FOTO", and "CADASTRAR". At the bottom of the window, the text "Sistema ContVoto - Servidor Versão" is displayed.

Criar eleição

CRIAR UMA ELEIÇÃO ✖

Nome e início da eleição:

*Nome:

*Data de início: *Hora de início:

Tipo de cargo:

Diretor-Geral Reitor

Escolha de candidatos:

Feliciano/13
João/45

>>>
<<<

Sistema ContVoto - Servidor Versão

Gerenciamento de candidato

GERENCIAMENTO DE CANDIDATO ✖

*Nome completo:

*Nome de urna:

*Partido:

*Número:

*Cargo:

FOTO

NOME	NOME/URNA	NÚMERO	PARTIDO	CARGO

Nome para pesquisa:

Sistema ContVoto - Servidor Versão

Gerenciamento de eleições

GERENCIAMENTO DE ELEIÇÕES ✖

Resultados:

NOME	INÍCIO	ENCERRAMENTO

Quantidade:

Nome:

Sistema ContVoto - Servidor Versão

Gerenciamento de eleitor

GERENCIAMENTO DE ELEITOR ✖

***Nome:**

Título de eleitor:

***Matrícula:**

***Categoria:**

ATUALIZAR **CANCELAR** **SALVAR**

FOTO

ALTERAR

NOME	MATRÍCULA	CATEGORIA

Nome: **RELATÓRIO** **LISTAR** **EXCLUIR**

Nome sistemaVersão

Gerenciamento de partido

GERENCIAMENTO DE PARTIDO ✖

Sigla:

Nome:

Número:

ATUALIZAR **CANCELAR** **SALVAR**

NOME	SIGLA	NÚMERO

Nome: **LISTAR** **EXCLUIR**

Sistema ContVoto - ServidorVersão

Gerenciamento de usuário

GERENCIAMENTO DE USUÁRIO ✖

***Nome:**

***CPF:**

RG:

***Categoria:**

***Login:**

***Senha:**

ATUALIZAR **CANCELAR** **SALVAR**

NOME	CPF	CATEGORIA

Nome: **LISTAR** **EXCLUIR**

Sistema ContVoto - Servidor Versão

Configuração de conexão com o servidor

CONFIGURAÇÃO DE CONEXÃO COM O SERVIDOR

Porta:

Ip:

Conectar

Identificação de eleitor

IDENTIFICAÇÃO

INFORME O NÚMERO DE SUA MATRÍCULA OU TÍTULO DE ELEITOR

1	2	3
4	5	6
7	8	9
←	0	→

Votação para o cargo de Reitor

VOTAÇÃO PARA REITOR

NÚMERO: <input type="text"/>	
NOME:	
CARGO:	
PARTIDO:	

1	2	3
4	5	6
7	8	9
	0	
BRANCO	CORRIGIR	CONFIRMAR

Votação para o cargo de Diretor-Geral

VOTAÇÃO PARA DIRETOR-GERAL



NÚMERO:

NOME:
CARGO:
PARTIDO:

1	2	3
4	5	6
7	8	9
0		
BRANCO	CORRIGIR	CONFIRMAR

DETALHAMENTO DOS CASOS DE USO

Fazer *login*

Atores: Administrador e Usuário
Pré-condições: O módulo servidor do sistema encontra-se instalado na máquina, aberto na tela <i>login</i> e conectado ao Gerenciador de Banco de Dados.
Pós-condições: Sistema aberto e com todos os recursos liberados conforme o nível de acesso do usuário logado.
Requisitos correlacionados: RF001, RF002, RF010, RF014, RF033 e RF034.
Variações Tecnológicas: O sistema terá um usuário padrão com autorização de acesso a todos os recursos, podendo o mesmo ser modificado, não podendo o mesmo ficar sem nenhum desse tipo de usuário; O ator não poderá acessar a tela principal enquanto a tela de <i>login</i> estiver aberta.
Prioridade: Essencial.
Fluxo principal: <ol style="list-style-type: none"> 1. O ator aciona o ícone de atalho do sistema ou a opção "Trocar Usuário" no menu "Sistema". 2. O sistema apresenta a tela de <i>login</i>. 3. O ator preenche os campos e aciona o botão "ENTRAR" (Fe001) (Fe002) (Fe003) (Fe004) (Fe004) (Fe006). 4. O sistema abre a sua tela principal.
Fluxo de exceções: (Fe001) Campo login vazio. O sistema apresenta uma mensagem informando que o campo LOGIN precisa ser preenchido e volta para o fluxo principal no passo 3. (Fe002) Login muito curto. O sistema apresenta uma mensagem informando que o login informado é muito curto e volta para o fluxo principal no passo 3. (Fe003) Preencher o campo senha. O sistema apresenta uma mensagem informando que o campo senha precisa ser preenchido e volta para o fluxo principal no passo 3. (Fe004) Senha muito curta. O sistema apresenta uma mensagem informando que a senha inserida é muito curta e volta para o fluxo principal no passo 3. (Fe005) Login incorreto. O sistema apresenta uma mensagem informando que o login inserido é incorreto e volta para o fluxo principal no passo 3. (Fe006) Senha incorreta. O sistema apresenta uma mensagem informando que a senha inserida é incorreta e volta para o fluxo principal no passo 3.
GUI: "Login" e "Principal".

Cadastrar usuário

Atores: Administrador
Pré-condições: O sistema encontra-se instalado na máquina, funcionando e com um administrador logado.
Pós-condições: Usuário cadastrado.
Requisitos correlacionados: RF001, RF010, RF018, RF019, RF020 e RF024.
Variações Tecnológicas: Somente um usuário do tipo administrador poderá realizar

cadastro de usuários; O ator não poderá acessar a tela principal enquanto a tela de cadastro de usuário estiver aberta.

Prioridade: Importante.

Fluxo principal:

1. O ator aciona a opção "Usuário" no menu "Cadastro".
2. O sistema preenche o campo "Categoria" com as categorias existentes.
3. O sistema apresenta a tela de cadastro de usuário.
4. O ator preenche os campos e aciona o botão "Salvar" (Fe001) (Fe002) (Fe003) (Fe004) (Fe005) (Fe006) (Fe007) (Fe008).
5. O sistema apresenta uma mensagem dizendo que a operação foi realizada com sucesso.

Fluxo de exceções:

(Fe001) Campo nome vazio.

O sistema apresenta uma mensagem informando que o campo nome precisa ser preenchido e volta para o fluxo principal no passo 4.

(Fe002) Nome muito curto.

O sistema apresenta uma mensagem informando que o nome informado é muito curto, solicitando que seja inserido ao menos cinco caracteres e volta para o fluxo principal no passo 4.

(Fe002) Campo CPF vazio.

O sistema apresenta uma mensagem informando que o campo CPF precisa ser preenchido e volta para o fluxo principal no passo 4.

(Fe003) Selecionar uma categoria.

O sistema apresenta uma mensagem solicitando que seja selecionada uma categoria de usuário e volta para o fluxo principal no passo 4.

(Fe004) Campo login vazio.

O sistema apresenta uma mensagem solicitando que o campo login seja preenchido e volta para o fluxo principal no passo 4.

(Fe005) Login muito curto.

O sistema apresenta uma mensagem informando que o login é muito, solicitando que seja o mesmo contenha ao menos cinco caracteres e volta para o fluxo principal no passo 4.

(Fe006) Campo senha vazio.

O sistema apresenta uma mensagem solicitando que o campo senha seja preenchido e volta para o fluxo principal no passo 4.

(Fe007) Senha muito curta.

O sistema apresenta uma mensagem informado que a senha é muito curta, solicitando que seja inserido ao menos seis caracteres e volta para o fluxo principal no passo 4.

(Fe008) Usuário cadastrado.

O sistema apresenta uma mensagem informado que o processo não pode ser realizado porque o usuário já é cadastrado e volta para o fluxo principal no passo 4.

GUI: "Principal" e "Cadastro de Usuário".

Cadastrar partido

Atores: Administrador e usuário
Pré-condições: O sistema encontra-se instalado na máquina, funcionando e com um usuário ou administrador logado.
Pós-condições: Partido cadastrado.
Requisitos correlacionados: RF001, RF017, RF018, RF019, RF020, RF024 e RF031.
Variações Tecnológicas: O sistema deverá permitir o cadastro a quantidade de partidos desejada.

Prioridade: Importante.
Fluxo principal: <ol style="list-style-type: none"> 1. O ator aciona a opção "Partido" no menu "Cadastro". 2. O sistema apresenta a tela de cadastro. 3. O ator preenche os campos (Fe001). 4. O ator aciona o botão "Salvar" (Fe002) (Fe003) (Fe004). 5. O sistema apresenta uma mensagem dizendo que a operação foi realizada com sucesso.
Fluxo de exceções: <p>(Fe001) Número de dois dígitos. O sistema apresenta uma mensagem informando que o campo número poderá conter no máximo dois dígitos e volta para o fluxo principal no passo 3.</p> <p>(Fe002) Algum campo vazio. O sistema apresenta uma mensagem informando que os campos precisam ser preenchidos e volta para o fluxo principal no passo 3.</p> <p>(Fe003) Número errado. O sistema apresenta uma mensagem informando que o número inserido tem que conter no mínimo dois dígitos e volta para o fluxo principal no passo 3.</p> <p>(Fe004) Partido já cadastrado. O sistema apresenta uma mensagem informando que o processo não pode ser realizado porque o partido já encontra-se cadastrado e volta para o fluxo principal no passo 3.</p>
GUI: "Principal" e "Cadastro de Partido".

Cadastrar candidato

Atores: Administrador e usuário
Pré-condições: O sistema encontra-se instalado na máquina, funcionando e com um administrador ou usuário logado.
Pós-condições: Candidato cadastrado
Requisitos correlacionados: RF001, RF017, RF018, RF020, RF024, RF027, RF029, RF030 e RF032
Variações Tecnológicas: O sistema deverá deixar efetivar uma cadastro de usuário somente se ele pertencer a um determinado candidato.
Prioridade: Importante.
Fluxo principal: <ol style="list-style-type: none"> 1. O ator aciona a opção "Candidato" no menu "Cadastrar". 2. O sistema preenche o campo "Partido" com os partidos cadastrados. 3. O sistema preenche o campo "Categoria" com as categorias de candidatos. 4. O sistema apresenta a tela "Cadastro de candidato". 5. O ator preenche os campos (Fe001) (Fe002) (Fe003) (Fe004). 6. O ator aciona o botão salvar (Fe005) (Fe006) (Fe007) (Fe008) (Fe009) (Fe010) (Fe011). 7. O sistema apresenta uma mensagem dizendo que a operação foi realizada com sucesso.
Fluxo de exceções: <p>(Fe001) Quantidade máxima de caracteres no campo "nome completo". O sistema apresenta uma mensagem informando que o campo "nome" não poderá conter mais que 45 caracteres e volta para o fluxo principal no passo 5.</p> <p>(Fe002) Quantidade máxima de caracteres no campo "nome na urna". O sistema apresenta uma mensagem informando que o campo "nome na urna" não poderá conter mais que 45 caracteres e volta para o fluxo principal no passo 5.</p> <p>(Fe003) Quantidade máxima de caracteres no campo "número". O sistema apresenta uma mensagem informando que o campo "número" não</p>

<p>poderá ser composto por mais de 3 caracteres e volta para o fluxo principal no passo 5.</p> <p>(Fe004) Imagem muito grande. O sistema apresenta uma mensagem informando que a imagem selecionada supera o tamanho padrão aceito e volta para o fluxo principal no passo 5.</p> <p>(Fe005) Selecionar uma imagem. O sistema apresenta uma mensagem informando que uma imagem deverá ser selecionada e volta para o fluxo principal no passo 5.</p> <p>(Fe006) Campo "Nome completo" vazio. O sistema apresenta uma mensagem informando que o campo "nome completo" está vazio e volta para o fluxo principal no passo 5.</p> <p>(Fe007) Campo "Nome na urna" vazio. O sistema apresenta uma mensagem informando que o campo "nome na urna" está vazio e volta para o fluxo principal no passo 5.</p> <p>(Fe008) Campo "Partido" vazio. O sistema apresenta uma mensagem solicitando que seja selecionado um partido e volta para o fluxo principal no passo 5.</p> <p>(Fe009) Campo "Cargo" vazio. O sistema apresenta uma mensagem solicitando que seja selecionado um cargo e volta para o fluxo principal no passo 5.</p> <p>(Fe010) Número informado é incompatível com o cargo de Reitor. O sistema apresenta uma mensagem solicitando que o número do candidato deverá conter dois dígitos e volta para o fluxo principal no passo 5.</p> <p>(Fe011) Número informado é incompatível com o cargo de Diretor-Geral. O sistema apresenta uma mensagem solicitando que o número do candidato deverá conter três dígitos e volta para o fluxo principal no passo 5.</p>
<p>GUI: "Principal" e "Cadastro de Candidato".</p>

Cadastrar eleitor

<p>Atores: Administrador e usuário</p>
<p>Pré-condições: O sistema encontra-se instalado na máquina, funcionando e com um administrador ou usuário logado.</p>
<p>Pós-condições: Eleitor cadastrado</p>
<p>Requisitos correlacionados: RF001, RF017, RF018, RF020, RF024 e RF030.</p>
<p>Variações Tecnológicas: O sistema deverá exigir que um eleitor somente seja cadastrado se ele pertencer a uma das categorias de eleitores.</p>
<p>Prioridade: Importante.</p>
<p>Fluxo principal:</p> <ol style="list-style-type: none"> 1. O ator aciona a opção "Eleitor" no menu "Cadastrar". 2. O sistema apresenta a tela de cadastro. 3. O ator preenche os campos (Fe001) (Fe002). 4. O ator aciona o botão "Cadastrar" (Fe003) (Fe004) (Fe005) (Fe006). 5. O sistema apresenta uma mensagem dizendo que a operação foi realizada com sucesso.

<p>Fluxo de exceções:</p> <p>(Fe001) Quantidade máxima de caracteres no campo "nome". O sistema apresenta uma mensagem informando que o campo "nome" não poderá conter mais que 45 caracteres e volta para o fluxo principal no passo 3.</p> <p>(Fe002) Imagem muito grande. O sistema apresenta uma mensagem informando que a imagem selecionada supera o tamanho padrão aceito e volta para o fluxo principal no passo 3.</p> <p>(Fe003) Selecionar uma imagem. O sistema apresenta uma mensagem informando que uma imagem deverá ser selecionada e volta para o fluxo principal no passo 3.</p> <p>(Fe004) Campo "Nome" vazio. O sistema apresenta uma mensagem informando que o campo "nome" está vazio e volta para o fluxo principal no passo 3.</p> <p>(Fe005) Campo "Matrícula" vazio. O sistema apresenta uma mensagem informando que o campo "matrícula" está vazio e volta para o fluxo principal no passo 3.</p> <p>(Fe006) Campo "Categoria" vazio. O sistema apresenta uma mensagem informando que o campo "categoria" está vazio, solicitando que seja selecionada uma categoria e volta para o fluxo principal no passo 3.</p> <p>GUI: "Principal" e "Cadastro de eleitor".</p>

Atualizar usuário

Atores: Administrador e usuário
Pré-condições: O sistema encontra-se instalado na máquina, funcionando e com um administrador ou usuário logado.
Pós-condições: Usuário atualizado
Requisitos correlacionados: RF001, RF010, RF017, RF018, RF019, RF020 e RF024.
Variações Tecnológicas: O sistema deverá autorizar que um usuário com limitação de acesso atualize somente seus dados; O sistema deverá também possibilitar que o usuário escolha quais dados quer editar (todos, senha, login etc.); O sistema deverá verificar se existe uma eleição em andamento e em caso de positivo impedir a atualização.
Prioridade: Importante.
<p>Fluxo principal:</p> <ol style="list-style-type: none"> 1. O ator aciona a opção "Usuário" no menu "Gerenciar". 2. O sistema apresenta a tela de gerenciamento de usuários. 3. O ator pesquisa o usuário desejado (Fa001) (Fa002). 4. O sistema apresenta os resultados (Fe001). 5. O ator seleciona o usuário desejado. 6. O sistema preenche os dados do usuário selecionado nos campos de edição. 7. O ator aciona o botão atualizar (Fe002). 8. O sistema apresenta uma tela com as opções de dados que o usuário deseja editar. 9. O autor seleciona os tipos de dados que deseja atualizar e aciona o botão "OK" (Fe011). 10. O sistema desbloqueia os campos com os dados do usuário selecionado para edição. 11. O ator redige os campos desejados. 12. O ator aciona o botão salvar (Fe003) (Fe004) (Fe005) (Fe006) (Fe007) (Fe006) (Fe009) (Fe010). 13. O sistema apresenta uma mensagem dizendo que a operação foi realizada com sucesso.

Fluxo de exceções:**(Fe001) Sem resultados.**

O sistema não apresenta nenhum resultado porque o nome informado não foi encontrado e volta para o fluxo principal no passo 3.

(Fe002) Usuário sem permissão.

O sistema apresenta uma mensagem informando que o usuário não tem permissão para editar os dados e volta para o fluxo principal no passo 6.

(Fe003) Campo nome vazio.

O sistema apresenta uma mensagem informando que o campo nome precisa ser preenchido e volta para o fluxo principal no passo 10.

(Fe004) Nome muito curto.

O sistema apresenta uma mensagem informando que o nome informado é muito curto, solicitando que seja inserido ao menos cinco caracteres e volta para o fluxo principal no passo 10.

(Fe005) Campo CPF vazio.

O sistema apresenta uma mensagem informando que o campo CPF precisa ser preenchido e volta para o fluxo principal no passo 10.

(Fe006) Selecionar uma categoria.

O sistema apresenta uma mensagem solicitando que seja selecionada uma categoria de usuário e volta para o fluxo principal no passo 10.

(Fe007) Campo login vazio.

O sistema apresenta uma mensagem solicitando que o campo login seja preenchido e volta para o fluxo principal no passo 10.

(Fe008) Login muito curto.

O sistema apresenta uma mensagem informando que o login é muito, solicitando que seja o mesmo contenha ao menos cinco caracteres e volta para o fluxo principal no passo 10.

(Fe009) Campo senha vazio.

O sistema apresenta uma mensagem solicitando que o campo senha seja preenchido e volta para o fluxo principal no passo 8.

(Fe010) Senha muito curta.

O sistema apresenta uma mensagem informando que a senha é muito curta, solicitando que seja inserido ao menos seis caracteres e volta para o fluxo principal no passo 10.

(Fe011) Nenhum tipo de dado selecionado.

O sistema apresenta uma mensagem informando que o usuário precisa de selecionar ao menos um tipo de dado que irá ser editado e volta para o fluxo principal no passo 8.

Fluxo alternativo:**(Fa001) O ator aciona o botão listar todos.****(Fa002) O ator preenche o campo nome.**

GUI: "Principal" e "Gerenciamento de usuário".

Atualizar partido

Atores: Administrador e usuário

Pré-condições: O sistema encontra-se instalado na máquina, funcionando e com um administrador ou usuário logado.

Pós-condições: Partido atualizado.

Requisitos correlacionados: RF001, RF017, RF018, RF020, RF030, RF031, RF032 e RF034.

Variações Tecnológicas: Será permitida a atualização nas informações dos partidos quantas vezes forem necessárias; O sistema deverá verificar se existe uma eleição em andamento e caso positivo impedir a atualização dos dados do partido.

<p>Prioridade: Importante.</p> <p>Fluxo principal:</p> <ol style="list-style-type: none"> 1. O ator aciona a opção "Partido" no menu "Gerenciar". 2. O sistema apresenta a tela de gerenciamento. 3. O ator pesquisa o partido (Fa001) (Fa002). 4. O sistema apresenta os resultados (Fe001). 5. O ator seleciona o partido. 6. O sistema preenche os campos de edição bloqueados com os dados do partido selecionado. 7. O ator aciona o botão atualizar (Fe002). 8. O sistema desbloqueia os campos para edição. 9. O ator redige os campos desejados. 10. O ator aciona o botão salvar (Fe003) (Fe004) (Fe005) (Fe006). 11. O sistema apresenta uma mensagem dizendo que a operação foi realizada com sucesso.
<p>Fluxo de exceções:</p> <p>(Fe001) Nenhum registro encontrado. O sistema não apresenta resultados porque nada foi encontrado e volta para o fluxo principal no passo 3.</p> <p>(Fe002) Edição não permitida. O sistema apresenta uma mensagem informando que a edição não poderá ser realizada porque existe uma eleição em andamento e volta para o fluxo principal no passo 7.</p> <p>(Fe003) Número de dois dígitos. O sistema apresenta uma mensagem informando que o campo número poderá conter no máximo dois dígitos e volta para o fluxo principal no passo 9.</p> <p>(Fe004) Algum campo vazio. O sistema apresenta uma mensagem informando que os campos precisam ser preenchidos e volta para o fluxo principal no passo 9.</p> <p>(Fe005) Número errado. O sistema apresenta uma mensagem informando que o número inserido tem que conter dois dígitos e volta para o fluxo principal no passo 9.</p> <p>(Fe006) Partido já cadastrado. O sistema apresenta uma mensagem informando que o processo não pode ser realizado porque o partido já encontra-se cadastrado e volta para o fluxo principal no passo 9.</p>
<p>Fluxo alternativo:</p> <p>(Fa001) O ator aciona o botão listar.</p> <p>(Fa002) O ator preenche o campo nome.</p>
<p>GUI: "Principal" e "Gerenciamento de partido".</p>

Atualizar candidato

<p>Atores: Administrador e usuário</p>
<p>Pré-condições: O sistema encontra-se instalado na máquina, funcionando e com um administrador ou usuário logado.</p>
<p>Pós-condições: Candidato atualizado.</p>
<p>Requisitos correlacionados: RF001, RF017, RF018, RF020, RF024, RF027, RF029, RF30, RF032 e RF034.</p>
<p>Variações Tecnológicas: Será permitida a atualização nas informações dos administradores quantas vezes forem necessárias; O sistema deverá impedir a edição caso exista uma eleição em andamento.</p>
<p>Prioridade: Importante.</p>

Fluxo principal:

1. O ator aciona a opção "Candidato" no menu "Gerenciar".
2. O sistema apresenta a tela de gerenciamento de candidato.
3. O ator pesquisa o candidato (Fa001) (Fa002).
4. O sistema apresenta os resultados (Fe001).
5. O ator seleciona o candidato.
6. O sistema preenche os campos de edição bloqueados com os dados do candidato selecionado.
7. O ator aciona o botão atualizar (Fe002).
8. O sistema desbloqueia os campos para edição.
9. O ator redige os campos desejados (Fe003) (Fe004) (Fe005) (Fe006).
10. O ator aciona o botão salvar (Fe007) (Fe008) (Fe009) (Fe010) (Fe011).
11. O sistema apresenta uma mensagem dizendo que a operação foi realizada com sucesso.

Fluxo de exceções:**(Fe001) Sem resultado.**

O sistema não apresenta nenhum resultado porque não foram encontrados registros e volta para o fluxo principal no passo 3.

(Fe002) Atualização não permitida.

O sistema apresenta uma mensagem informando os dados não poderão ser atualizados porque existe uma eleição em andamento e volta para o fluxo principal no passo 6.

(Fe003) Quantidade máxima de caracteres no campo "nome completo".

O sistema apresenta uma mensagem informando que o campo "nome" não poderá conter mais que 45 caracteres e volta para o fluxo principal no passo 8.

(Fe004) Quantidade máxima de caracteres no campo "nome na urna".

O sistema apresenta uma mensagem informando que o campo "nome na urna" não poderá conter mais que 45 caracteres e volta para o fluxo principal no passo 8.

(Fe005) Quantidade máxima de caracteres no campo "número".

O sistema apresenta uma mensagem informando que o campo "número" não poderá ser composto por mais de 3 caracteres e volta para o fluxo principal no passo 8.

(Fe006) Imagem muito grande.

O sistema apresenta uma mensagem informando que a imagem selecionada supera o tamanho padrão aceito e volta para o fluxo principal no passo 8.

(Fe007) Selecionar uma imagem.

O sistema apresenta uma mensagem informando que uma imagem deverá ser selecionada e volta para o fluxo principal no passo 9.

(Fe008) Campo "Nome completo" vazio.

O sistema apresenta uma mensagem informando que o campo "nome completo" está vazio e volta para o fluxo principal no passo 5.

(Fe009) Campo "Nome na urna" vazio.

O sistema apresenta uma mensagem informando que o campo "nome na urna" está vazio e volta para o fluxo principal no passo 9.

(Fe010) Campo "Partido" vazio.

O sistema apresenta uma mensagem solicitando que seja selecionado um partido e volta para o fluxo principal no passo 9.

(Fe011) Campo "Cargo" vazio.

O sistema apresenta uma mensagem solicitando que seja selecionado um cargo e volta para o fluxo principal no passo 9.

(Fe012) Número informado é incompatível com o cargo de Reitor.

O sistema apresenta uma mensagem solicitando que o número do candidato deverá conter dois dígitos e volta para o fluxo principal no passo 9.

(Fe013) Número informado é incompatível com o cargo de Diretor-Geral.

O sistema apresenta uma mensagem solicitando que o número do candidato deverá conter três dígitos e volta para o fluxo principal no passo 9.
Fluxo alternativo: (Fa001) O ator aciona o botão listar todos. (Fa002) O ator preenche o campo nome.
GUI: "Principal" e "Gerenciamento de candidato".

Atualizar eleitor

Atores: Administrador e usuário
Pré-condições: O sistema encontra-se instalado na máquina, funcionando e com um administrador ou usuário logado.
Pós-condições: Eleitor atualizado
Requisitos correlacionados: RF001, RF017, RF018, RF019, RF020, RF024 e RF034.
Variações Tecnológicas: Será permitida a atualização nas informações dos eleitores quantas vezes forem necessárias; O sistema deverá verificar um tamanho máximo permitido da imagem a ser inserida no sistema e, caso a mesma extrapole esse limite a atualização deverá ser abortada; O sistema deve impedir que uma atualização seja realizada caso existe uma eleição em andamento.
Prioridade: Importante.
Fluxo principal: 1. O ator aciona a opção "Eleitor" no menu "Gerenciar". 2. O sistema apresenta a tela de gerenciamento. 3. O ator pesquisa o eleitor (Fa001) (Fa002). 4. O sistema apresenta os resultados (Fe001). 5. O ator seleciona o eleitor. 6. O sistema preenche os campos bloqueados com as informações do eleitor selecionado 7. O ator aciona o botão atualizar (Fe002). 8. O sistema desbloqueia os campos com os dados do eleitor selecionado. 9. O ator redige os campos desejados (Fe003) (Fe004). 10. O ator aciona o botão salvar (Fe005) (Fe006) (Fe007) (Fe008). 11. O sistema apresenta uma mensagem dizendo que a operação foi realizada com sucesso.
Fluxo de exceções: (Fe001) Sem resultados. O sistema não apresenta nenhum usuário porque não foram encontrados registros e volta para o fluxo principal no passo 3. (Fe002) Atualização não permitida. O sistema apresenta uma mensagem informando que existe uma eleição em andamento e volta para o fluxo principal no passo 6. (Fe003) Quantidade máxima de caracteres no campo "nome". O sistema apresenta uma mensagem informando que o campo "nome" não poderá conter mais que 45 caracteres e volta para o fluxo principal no passo 8. (Fe004) Imagem muito grande. O sistema apresenta uma mensagem informando que a imagem selecionada supera o tamanho padrão aceito e volta para o fluxo principal no passo 8. (Fe005) Selecionar uma imagem. O sistema apresenta uma mensagem informando que uma imagem deverá ser selecionada e volta para o fluxo principal no passo 9. (Fe006) Campo "Nome" vazio. O sistema apresenta uma mensagem informando que o campo "nome" está vazio e volta para o fluxo principal no passo 9.

<p>(Fe007) Campo "Matrícula" vazio. O sistema apresenta uma mensagem informando que o campo "matrícula" está vazio e volta para o fluxo principal no passo 9.</p> <p>(Fe008) Campo "Categoria" vazio. O sistema apresenta uma mensagem informando que o campo "categoria" está vazio, solicitando que seja selecionada uma categoria e volta para o fluxo principal no passo 9.</p>
<p>Fluxo alternativo: (Fa001) O ator aciona o botão listar. (Fa002) O ator preenche o campo nome.</p>
<p>GUI: "Principal" e "Gerenciamento de eleitor".</p>

Excluir usuário

<p>Atores: Administrador.</p>
<p>Pré-condições: O sistema encontra-se instalado na máquina, funcionando e com um administrador logado.</p>
<p>Pós-condições: Usuário excluído.</p>
<p>Requisitos correlacionados: RF001 e RF024.</p>
<p>Variações Tecnológicas: O sistema deverá informar as consequências no banco de dados na exclusão de um usuário, como também o motivo pelo qual o mesmo não poderá ser excluído.</p>
<p>Prioridade: Importante.</p>
<p>Fluxo principal:</p> <ol style="list-style-type: none"> 1. O ator aciona a opção "Usuário" no menu "Gerenciar". 2. O sistema apresenta a tela de gerenciamento de usuários. 3. O ator pesquisa o usuário desejado (Fa001) (Fa002). 4. O sistema apresenta os resultados (Fe001). 5. O ator seleciona o usuário desejado. 6. O sistema preenche os dados do usuário selecionado nos campos bloqueados de edição. 7. O ator aciona o botão "Excluir" (Fe002) (Fe003) (Fe004). 8. O sistema apresenta uma mensagem dizendo que a operação foi realizada com sucesso.
<p>Fluxo de exceções:</p> <p>(Fe001) Sem resultados. O sistema não apresenta nenhum resultado porque o nome informado não foi encontrado e volta para o fluxo principal no passo 3.</p> <p>(Fe002) Histórico de usuário. O sistema apresenta uma mensagem informando que o usuário não pode ser excluído porque o mesmo faz parte de uma relação de vários atos registrados no banco de dados e volta para o fluxo principal no passo 5.</p> <p>(Fe003) Usuário logado. O sistema apresenta uma mensagem que o usuário não pode ser excluído porque encontra-se logado e volta para o fluxo principal no passo 5.</p> <p>(Fe004) Único administrador. O sistema apresenta uma mensagem informando que o usuário não pode ser excluído porque é o único administrador cadastrado e volta para o fluxo principal no passo 5.</p>
<p>Fluxo alternativo: (Fa001) O ator aciona o botão listar todos. (Fa002) O ator preenche o campo nome.</p>
<p>GUI: "Principal" e "Gerenciamento de usuário".</p>

Excluir partido

Atores: Administrador
Pré-condições: O sistema encontra-se instalado na máquina, funcionando e com um administrador logado.
Pós-condições: Partido excluído.
Requisitos correlacionados: RF001 e RF024.
Variações Tecnológicas: O sistema deverá informar as consequências no banco de dados na exclusão de um partido, como também o motivo pelo qual o mesmo não poderá ser excluído.
Prioridade: Importante.
Fluxo principal: <ol style="list-style-type: none"> 1. O ator aciona a opção "Usuário" no menu "Gerenciar". 2. O sistema apresenta a tela de gerenciamento de usuários. 3. O ator pesquisa o usuário desejado (Fa001) (Fa002). 4. O sistema apresenta os resultados (Fe001). 5. O ator seleciona o usuário desejado. 6. O sistema preenche os dados do usuário selecionado nos campos bloqueados de edição. 7. O ator aciona o botão "Excluir" (Fe002). 8. O sistema apresenta uma mensagem dizendo que a operação foi realizada com sucesso.
Fluxo de exceções: (Fe001) Sem resultados. O sistema não apresenta nenhum resultado porque o nome informado não foi encontrado e volta para o fluxo principal no passo 2. (Fe002) Histórico do partido. O sistema apresenta uma mensagem informando que o partido não pode ser excluído porque o mesmo faz parte de uma relação de vários atos registrados no banco de dados e volta para o fluxo principal no passo 5.
Fluxo alternativo: (Fa001) O ator aciona o botão listar. (Fa002) O ator preenche o campo nome.
GUI: "Principal" e "Gerenciamento de partido".

Excluir candidato

Atores: Administrador
Pré-condições: O sistema encontra-se instalado na máquina, funcionando e com um administrador logado.
Pós-condições: Candidato excluído.
Requisitos correlacionados: RF001 e RF024.
Variações Tecnológicas: O sistema deverá informar as consequências no banco de dados na exclusão de um candidato, como também o motivo pelo qual o mesmo não poderá ser excluído.
Prioridade: Importante.
Fluxo principal: <ol style="list-style-type: none"> 1. O ator aciona a opção "Candidato" no menu "Gerenciar". 2. O sistema apresenta a tela de gerenciamento de candidato. 3. O ator pesquisa o candidato (Fa001) (Fa002). 4. O sistema apresenta os resultados (Fe001). 5. O ator seleciona o candidato.

<p>6. O sistema preenche os campos de edição bloqueados com os dados do candidato selecionado.</p> <p>7. O ator aciona o botão "Excluir" (Fe002).</p> <p>8. O sistema apresenta uma mensagem dizendo que a operação foi realizada com sucesso.</p>
<p>Fluxo de exceções:</p> <p>(Fe001) Sem resultado. O sistema não apresenta nenhum resultado porque não foram encontrados registros e volta para o fluxo principal no passo 3.</p> <p>(Fe002) Exclusão não permitida. O sistema apresenta uma mensagem informando que a exclusão não poderá ser realizada e volta para o fluxo principal no passo 6.</p>
<p>Fluxo alternativo:</p> <p>(Fa001) O ator aciona o botão listar todos.</p> <p>(Fa002) O ator preenche o campo nome.</p>
<p>GUI: "Principal" e "Gerenciamento de candidato".</p>

Excluir eleitor

<p>Atores: Administrador</p>
<p>Pré-condições: O sistema encontra-se instalado na máquina, funcionando e com um administrador logado.</p>
<p>Pós-condições: Eleitor excluído</p>
<p>Requisitos correlacionados: RF001 e RF024.</p>
<p>Variações Tecnológicas: O sistema deverá informar as consequências no banco de dados na exclusão de um eleitor, como também o motivo pelo qual o mesmo não poderá ser excluído.</p>
<p>Prioridade: Importante.</p>
<p>Fluxo principal:</p> <ol style="list-style-type: none"> 1. O ator aciona a opção "Eleitor" no menu "Gerenciar". 2. O sistema apresenta a tela de gerenciamento. 3. O ator pesquisa o eleitor (Fa001) (Fa002). 4. O sistema apresenta os resultados (Fe001). 5. O ator seleciona o eleitor. 6. O sistema preenche os campos bloqueados com as informações do eleitor selecionado 7. O ator aciona o botão "Excluir" (Fe002). 8. O sistema apresenta uma mensagem dizendo que a operação foi realizada com sucesso.
<p>Fluxo de exceções:</p> <p>(Fe001) Sem resultado. O sistema não apresenta nenhum resultado porque não foram encontrados registros e volta para o fluxo principal no passo 3.</p> <p>(Fe002) Exclusão não permitida. O sistema apresenta uma mensagem informando que a exclusão não poderá ser realizada e volta para o fluxo principal no passo 6.</p>
<p>Fluxo alternativo:</p> <p>(Fa001) O ator aciona o botão listar.</p> <p>(Fa002) O ator preenche o campo nome.</p>
<p>GUI: "Principal" e "Gerenciamento de eleitor".</p>

Configurar conexão com BD

Atores: Administrador
Pré-condições: O sistema encontra-se instalado na máquina.
Pós-condições: Banco de dados conectado
Requisitos correlacionados: RF001, RF002, RF003, RF004, RF005 e RF008.
Variações Tecnológicas: O sistema deverá informar quando não for encontrado um Gerenciador de Banco de Dados, como também se o mesmo possui seu banco de dados, criando-o automaticamente caso necessário, sem a necessidade de intervenção do usuário para tal.
Prioridade: Essencial.
Fluxo principal: 1. O ator aciona o ícone de atalho do módulo servidor na área de trabalho. 2. O sistema apresenta uma mensagem informando que o banco de dados não está conectado. 3. O ator aciona o botão "OK". 4. O sistema apresenta a tela de configuração do banco de dados. 5. O ator preenche os campos aciona o botão "OK" (Fe001). 7. O sistema confirma a conexão (Fa001) (Fa002).
Fluxo de exceções: (Fe001) Conexão não realizada. O sistema apresenta uma mensagem informando que não foi conectado ao banco de dados e volta para o fluxo principal no passo 4.
Fluxo alternativo: (Fa001) O sistema confirma que o banco de dados já existe. (Fa002) O sistema confirma que o banco de dados foi criado.
GUI: "Administrar conexão do banco de dados".

Configurar Backup

Atores: Administrador
Pré-condições: O sistema encontra-se instalado na máquina, com um administrador logado e MySqlDump não configurado.
Pós-condições: MySqlDump configurado.
Requisitos correlacionados: RF001, RF003 e RF008.
Variações Tecnológicas: Os serviços de cópia e restauração do Gerenciador de Banco de Dados MySql utiliza uma aplicação para que esses serviços sejam executados de forma externa ao gerenciador, ou seja, sem está logado no mesmo. Essa aplicação encontra-se na pasta onde são armazenados os arquivos do gerenciador, sendo necessário que o caminho da mesma seja indicado para que o MySql consiga efetivar os serviços de backup.
Prioridade: Importante.
Fluxo principal: 1. O ator aciona o botão "Banco de dados" na tela principal. 2. O sistema apresenta a tela de gerenciamento de banco de dados na aba "Configurar MySql". 3. O ator pesquisa o caminho da aplicação e aciona o botão "Salvar" (Fe001) (Fe002). 4. O sistema apresenta uma mensagem dizendo que a operação foi realizada com sucesso.
Fluxo de exceções: (Fe001) Caminho vazio. O sistema apresenta uma mensagem o campo do caminho deve ser preenchido e volta para o fluxo principal no passo 2.

(Fe002) MySql não configurado. O sistema apresenta uma mensagem informando que a configuração não foi realizada e volta para o fluxo principal no passo 2.
GUI: "Principal" e "Gerenciamento de banco de dados".

Iniciar eleição

Atores: Administrador
Pré-condições: O sistema encontra-se instalado na máquina, funcionando e com um administrador logado.
Pós-condições: Eleição iniciada e em andamento.
Requisitos correlacionados: RF001, RF010, RF017, RF018, RF019, RF023, RF028 e RF034.
Variações Tecnológicas: O sistema deverá verificar se já existe uma eleição em andamento e em caso positivo impedir a abertura de outra.
Prioridade: Importante.
Fluxo principal: 1. O ator aciona o botão "Iniciar" ou a opção "Criar" no menu "Eleições". 2. O sistema apresenta a tela de "Criar uma eleição". 3. O ator preenche os campos (Fa001) (Fa002). 4. O ator aciona o botão "Criar" (Fe002) (Fe003) (Fe004) (Fe005) (Fe006). 5. O sistema apresenta uma mensagem dizendo que a operação foi realizada com sucesso.
Fluxo de exceções: (Fe001) Participação não autorizada. O sistema apresenta uma mensagem informando que o candidato não poderá participar da eleição porque existe concorrentes do seu partido participando do certame e volta para o fluxo principal no passo 2. (Fe002) Candidato incompatível com o cargo escolhido. O sistema apresenta uma mensagem informando que o candidato selecionado faz parte de uma categoria diferente da escolhida e volta para o fluxo principal no passo 2. (Fe003) Preencher o campo nome. O sistema apresenta uma mensagem informando que o campo nome precisa ser preenchido e volta para o fluxo principal no passo 2. (Fe004) Data inferior à atual. O sistema apresenta uma mensagem informando a data de início não pode ser inferior à atual e volta para o fluxo principal no passo 2. (Fe005) Hora inferior à atual. O sistema apresenta uma mensagem informando a hora de início não pode ser inferior à atual e volta para o fluxo principal no passo 2. (Fe006) Selecionar candidatos para todas as categorias. O sistema apresenta uma mensagem informando não foram selecionados candidatos para todas as categorias desejadas e volta para o fluxo principal no passo 2.
Fluxo alternativo: (Fa001) Uma única categoria (Fe001). (Fa002) Mais de uma categoria. Escolha de qual cargo um candidato irá concorrer (Fe001) (Fe002).
GUI: "Principal" e "Criar uma eleição".

Autorizar eleitor votar

Atores: Administrador e usuário
Pré-condições: O sistema encontra-se instalado na máquina, funcionando e com um

administrador ou usuário logado com uma eleição em andamento.
Pós-condições: Eleitor autorizado a votar
Requisitos correlacionados: RF001, RF011, RF017, RF025 e RF034.
Variações Tecnológicas: O sistema deverá informar, ao pesquisar um eleitor, se o mesmo está autorizado a votar na eleição em andamento e se o mesmo já votou.
Prioridade: Importante.
Fluxo principal: 1. O ator preenche o campo "nome do eleitor". 2. O sistema apresenta os resultados na tabela (Fe001). 3. O ator seleciona um eleitor. 4. O sistema apresenta os campos com os dados do eleitor selecionado (Fe002). 5. O ator aciona o botão "Autorizar para votação". 6. O sistema apresenta uma mensagem dizendo que a operação foi realizada com sucesso.
Fluxo de exceções: (Fe001) Sem resultado. O sistema não apresenta nenhum resultado porque não foram encontrados registros e volta para o fluxo principal no passo 1. (Fe002) Eleitor já autorizado a votar. O sistema bloqueia o botão "Liberar para votação" e volta para o fluxo principal no passo 3.
GUI: "Principal".

Autoidentificação de eleitor

Atores: Eleitor
Pré-condições: O sistema (cliente) encontra-se instalado na máquina, funcionando, conectado ao módulo servidor e com a tela de identificação aberta.
Pós-condições: Eleitor identificado.
Requisitos correlacionados: RF001, RF007, RF011, RF034, RF036, RF037 e RF038.
Variações Tecnológicas: Ao identificar o usuário o sistema deverá verificar se o mesmo já votou na eleição em andamento e impedi-lo de votar.
Prioridade: Importante.
Fluxo principal: 1. O ator aciona os botões do teclado. 2. O sistema preenche o campo de identificação com os dados do teclado. 3. O ator aciona o botão "->" (Fe001) (Fe002) (Fe003) (Fe004) (Fe005). 4. O sistema apresenta a tela de votação.
Fluxo de exceções: (Fe001) Campo "identificação" vazio. O sistema apresenta uma mensagem informando que o campo de identificação precisa ser preenchido e volta para o fluxo principal no passo 1. (Fe002) Sem eleição em andamento. O sistema apresenta uma mensagem informando que não existe eleição em andamento e volta para o fluxo principal no passo 1. (Fe003) Eleitor não encontrado. O sistema apresenta uma mensagem informando que o eleitor não foi encontrado e volta para o fluxo principal no passo 1. (Fe004) Eleitor não autorizado a votar. O sistema apresenta uma mensagem informando que o eleitor não está autorizado a votar na eleição em andamento e volta para o fluxo principal no passo 1. (Fe005) Eleitor já votou.

O sistema apresenta uma mensagem informando que o eleitor já votou na eleição em andamento e volta para o fluxo principal no passo 1.
GUI: "Identificação".

Escolher candidato

Atores: Eleitor
Pré-condições: O módulo cliente do sistema encontra-se instalado na máquina, funcionando, com um eleitor identificado, conectado ao servidor e com a tela de votação aberta.
Pós-condições: Candidato selecionado e voto computado para o mesmo.
Requisitos correlacionados: RF001, RF007, RF013, RF021, RF022, RF026, RF036, RF038, RF053, RF054, RF055 e RF056.
Variações Tecnológicas: O sistema deverá identificar se a eleição em andamento tem mais de uma categoria de candidatos, situação que o mesmo deverá disponibilizar inicialmente a tela na qual registra o voto para o cargo Reitor; Caso a eleição seja somente para uma única categoria o sistema deverá identificar qual delas e disponibilizar a tela correta para o eleitor votar; Caso a eleição seja para mais de uma categoria e o eleitor só votou em uma delas, o sistema deverá disponibilizar somente a tela de votação correspondente a categoria que ele ainda não votou.
Prioridade: Essencial.
Fluxo principal: 1. O ator escolhe sua opção de voto (Fa001) (Fa002) (Fa002). 2. O ator aciona o botão "Confirmar" (Fe001). 3. O sistema apresenta uma mensagem dizendo que o voto foi computado com sucesso.
Fluxo de exceções: (Fe001) Voto não computado. O sistema apresenta uma mensagem avisando que o voto não foi computado porque o eleitor já votou, fecha a tela de votação e volta para tela de identificação.
Fluxo alternativo: (Fa001) Voto nulo. O ator não digita nenhum número. O ator digita um número que não corresponde a nenhum candidato. (Fa002) Voto em branco. O ator aciona a tecla "Branco". (Fa002) Voto válido. O ator digita um número e o sistema apresenta o candidato.
GUI: "Identificação de eleitor", "Votação para Reitor" e "Votação para Diretor-Geral".

Encerrar votação

Atores: Administrador
Pré-condições: O sistema encontra-se instalado na máquina, funcionando, com um administrador logado e uma eleição em andamento.
Pós-condições: Eleição encerrada e resultado apresentado.
Requisitos correlacionados: RF001, RF010, RF034, RF057, RF058 e RF060.
Variações Tecnológicas: Somente um usuário com nível de administrador poderá encerrar uma votação.
Prioridade: Importante.

<p>Fluxo principal:</p> <ol style="list-style-type: none"> 1. O ator aciona o botão "Encerrar" na tela principal. 2. O sistema pergunta se realmente o ator quer encerrar a votação (Fe001). 3. O sistema encerra a eleição e pergunta se o ator quer emitir o relatório da eleição (Fa001) (Fa002). 4. Operação realizada com sucesso.
<p>Fluxo de exceções:</p> <p>(Fe001) Ator não quer encerrar a votação. O ator aciona a opção "Não" ou "Cancelar" e o sistema volta para o fluxo principal no passo 1.</p>
<p>Fluxo alternativo:</p> <p>(Fa001) Sem relatório da votação. O ator aciona o botão "Não" e o sistema somente encerra a votação sem apresentar o resultado.</p> <p>(Fa002) Com relatório. O ator aciona o botão "Sim" e o sistema apresenta o resultado detalhado.</p>
<p>GUI: "Principal".</p>

Configurar conexão com o servidor

<p>Atores: Administrador</p>
<p>Pré-condições: O sistema (módulo cliente) encontra-se instalado na máquina e o módulo servidor funcionando no mesmo ou outro computador, neste caso estando os dois conectados no mesmo roteador.</p>
<p>Pós-condições: Servidor conectado.</p>
<p>Requisitos correlacionados: RF003, RF006, RF007, RF008, RF018 e RF034.</p>
<p>Variações Tecnológicas: Os dados da conexão deverão ser armazenados de forma permanente em um arquivo para sempre serem acessados quando necessário.</p>
<p>Prioridade: Essencial.</p>
<p>Fluxo principal:</p> <ol style="list-style-type: none"> 1. O ator aciona o atalho do sistema. 2. O sistema constata que não está conectado ao servidor e abre a tela de configuração. 3. O ator preenche os campos e aciona o botão "Conectar". 4. O sistema apresenta uma mensagem dizendo que o módulo foi conectado com sucesso.
<p>Fluxo de exceções:</p> <p>(Fe001) Campo "Porta" vazio. O sistema apresenta uma mensagem dizendo que o campo "Porta" precisa ser preenchido e volta para o fluxo principal no passo 3.</p> <p>(Fe002) Campo "Ip" vazio. O sistema apresenta uma mensagem dizendo que o campo "Ip" precisa ser preenchido e volta para o fluxo principal no passo 3.</p> <p>(Fe003) Conexão não realizada. O sistema apresenta uma mensagem dizendo que o conexão não foi realizada e volta para o fluxo principal no passo 3.</p>
<p>GUI: "Configuração de conexão com o servidor".</p>

Apresentar resultado

<p>Atores: Administrador e usuário.</p>
<p>Pré-condições: O sistema encontra-se instalado na máquina, funcionando e com um administrador ou usuário logado.</p>
<p>Pós-condições: Resultado emitido.</p>

Requisitos correlacionados: RF001 e RF017.
Variações Tecnológicas: O sistema possibilitará que seja escolhida uma eleição realizada para a emissão de seu resultado detalhado; O sistema deverá impedir que uma eleição em andamento tenha o seu resultado emitido.
Prioridade: Importante.
Fluxo principal: 1. O ator aciona a opção "Gerenciar" no menu "Eleições". 2. O sistema apresenta a tela de gerenciamento de eleições. 3. O ator pesquisa a eleição (Fa001) (Fa002). 4. O sistema apresenta os resultados (Fe001). 5. O ator seleciona a eleição. 6. O ator aciona o botão "Relatório" (Fe002). 7. O sistema apresenta o resultado.
Fluxo de exceções: (Fe001) Sem resultado. O sistema não apresenta nenhum resultado porque não foram encontrados registros e volta para o fluxo principal no passo 2. (Fe002) Sem eleição selecionada. O sistema apresenta uma mensagem informando que nenhuma eleição foi selecionada e volta para o fluxo principal no passo 4.
Fluxo alternativo: (Fa001) O ator aciona o botão listar. (Fa002) O ator preenche o campo nome.
GUI: "Principal" e "Gerenciamento de eleições".

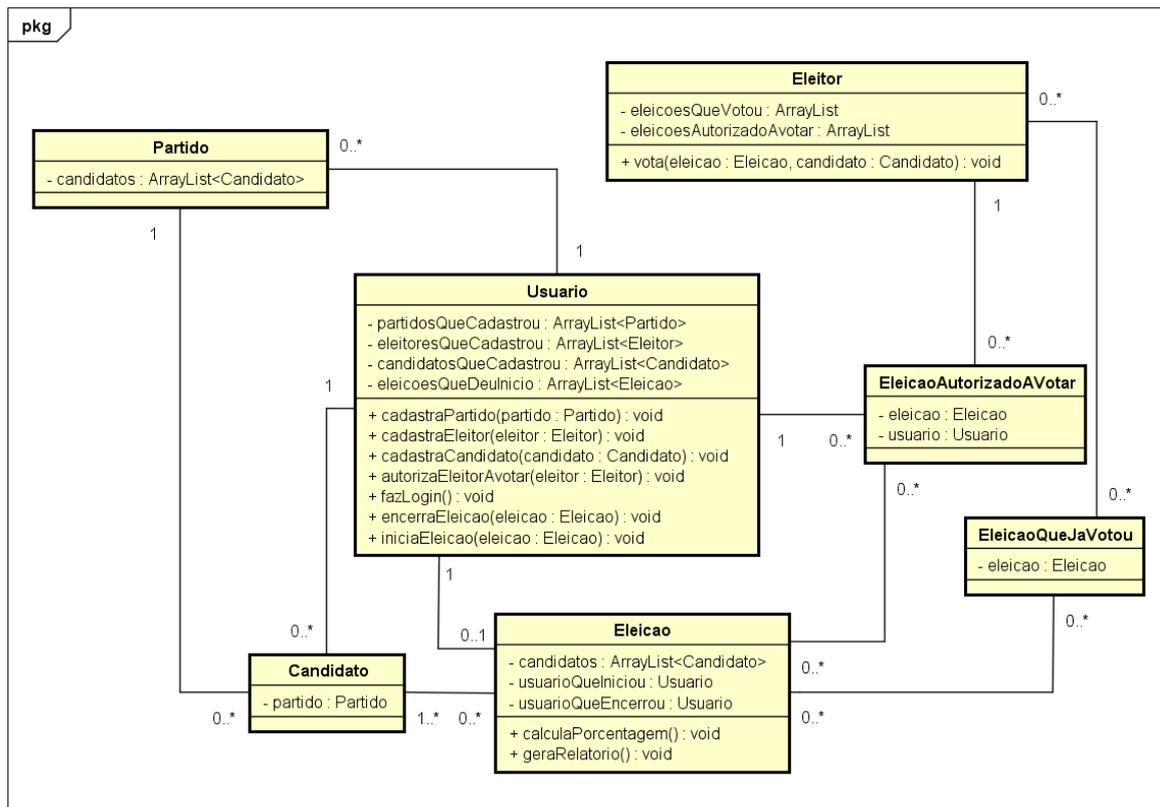
Listar eleitores

Atores: Administrador e usuário.
Pré-condições: O sistema encontra-se instalado na máquina, funcionando e com um administrador ou usuário logado.
Pós-condições: Lista de eleitores emitida.
Requisitos correlacionados: RF001 e RF017.
Variações Tecnológicas: O sistema deverá possibilitar a emissão de uma lista com os eleitores aptos a votar, separando-os por categoria.
Prioridade: Importante.
Fluxo principal: 1. O ator aciona a opção "Eleitor" no menu "Gerenciar". 2. O sistema apresenta a tela de gerenciamento. 3. O ator aciona o botão "Listar". 4. O sistema apresenta os resultados (Fe001). 5. O ator aciona o botão "Relatório" (Fe002). 6. O sistema apresenta uma mensagem dizendo que a operação foi realizada com sucesso.
Fluxo de exceções: (Fe001) Sem resultado. O sistema não apresenta nenhum resultado porque não foram encontrados registros e volta para o fluxo principal no passo 2. (Fe002) Sem eleitores. O sistema apresenta uma mensagem informando que a lista não poderá ser emitida porque não tem eleitores pesquisados e volta para o fluxo principal no passo 2.
GUI: "Principal" e "Gerenciamento de eleitor".

Fazer Backup

Atores: Administrador e usuário.
Pré-condições: O sistema encontra-se instalado na máquina, funcionando, com um administrador ou usuário logado e o MySQLDump configurado.
Pós-condições: Cópia de segurança realizada com sucesso.
Requisitos correlacionados: RF001.
Variações Tecnológicas: O sistema deverá acessar os recursos do MySQL para criar cópias de segurança do banco de dados; Também, com esses mesmos recursos, restaurar o banco de dados a partir de uma cópia de segurança.
Prioridade: Importante.
Fluxo principal: <ol style="list-style-type: none"> 1. O ator aciona o botão "Banco de dados" na tela principal. 2. O sistema apresenta a tela de gerenciamento de banco de dados na aba "Cópia de Segurança". 3. O ator aciona o botão "..." e escolhe a pasta onde será salvo o arquivo. 4. O sistema preenche o campo "Local" com o caminho da pasta. 5. O ator aciona o botão "Gerar cópia" (Fe001) (Fe002). 6. O sistema apresenta uma mensagem dizendo que a operação foi realizada com sucesso.
Fluxo de exceções: <p>(Fe001) Campo "Local" vazio. O sistema apresenta uma mensagem solicitando que o ator selecione uma pasta e volta para o fluxo principal no passo 3.</p> <p>(Fe002) Operação fracassada. O sistema não apresenta uma mensagem informando que a operação não foi realizada e volta para o fluxo principal no passo 3.</p>
GUI: "Principal" e "Gerenciamento de banco de dados".

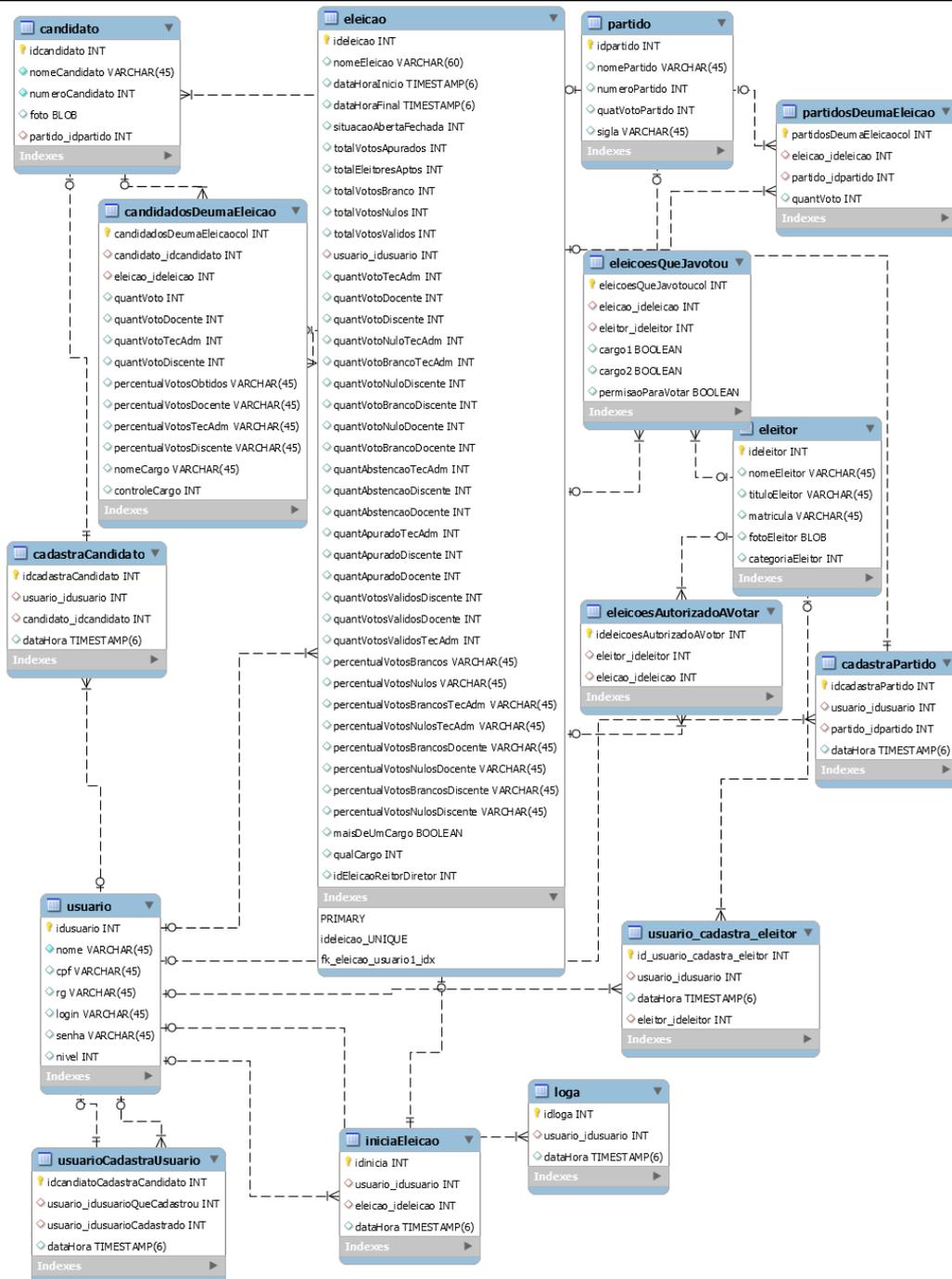
Diagrama de classes



PERSISTÊNCIA DE DADOS

Definição das tabelas com seus atributos e relacionamentos na camada de persistência, especificando assim o banco de dados do sistema.

Modelo Relacional

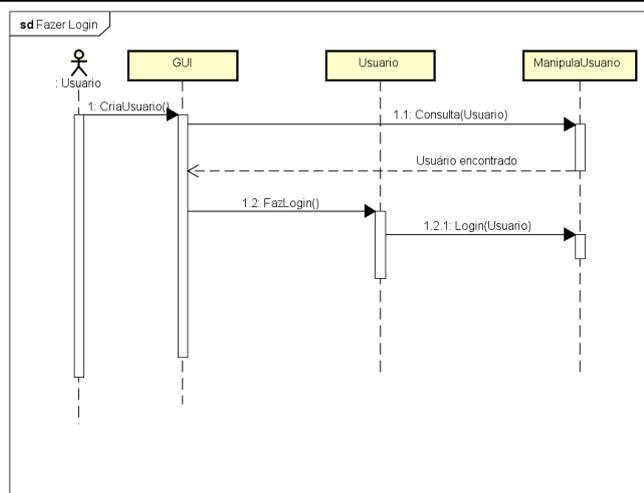


MODELAGEM DINÂMICA

Esta modelagem será dividida em duas partes, sendo que a primeira, através do diagrama de sequência, será detalhado iteração dos objetos de diversas classes em determinadas funcionalidades, não todas, pois várias delas é uma repetição da outras, podendo elas ficarem para serem resolvidas na programação. Enquanto que a segunda parte será especificado, através do diagrama de máquina de estados, o comportamento das telas de interação com o usuário.

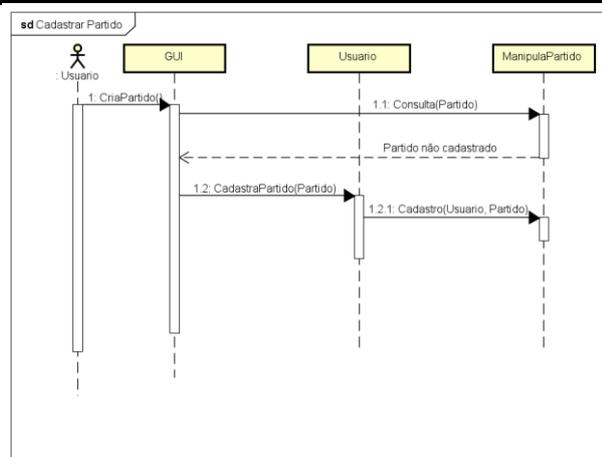
Diagrama de sequência

Fazer Login



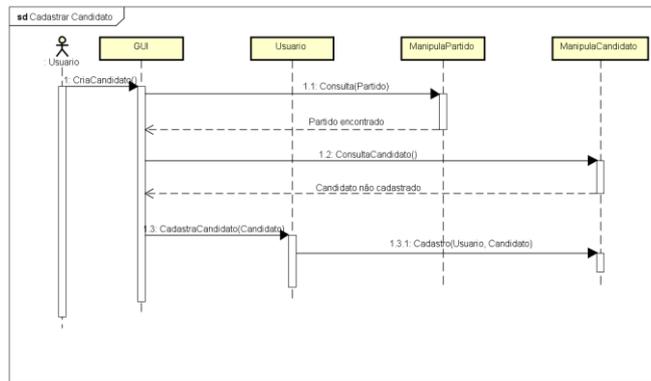
powered by Astah

Cadastrar partido



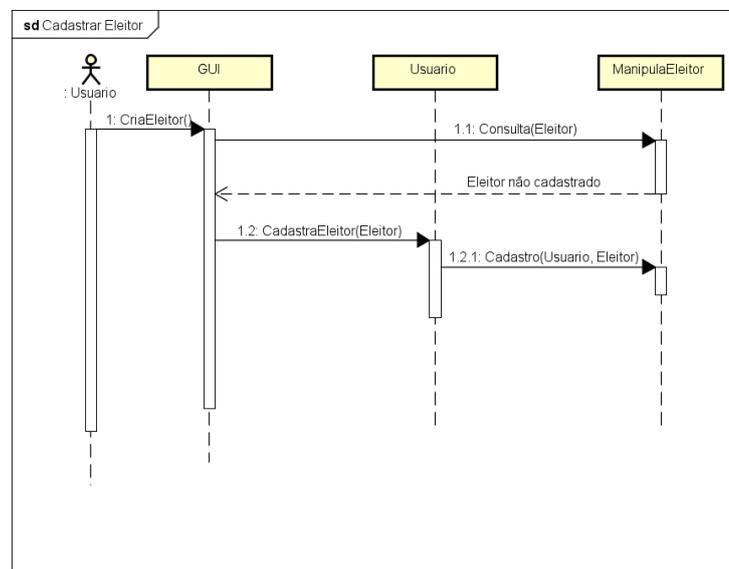
powered by Astah

Cadastrar candidato



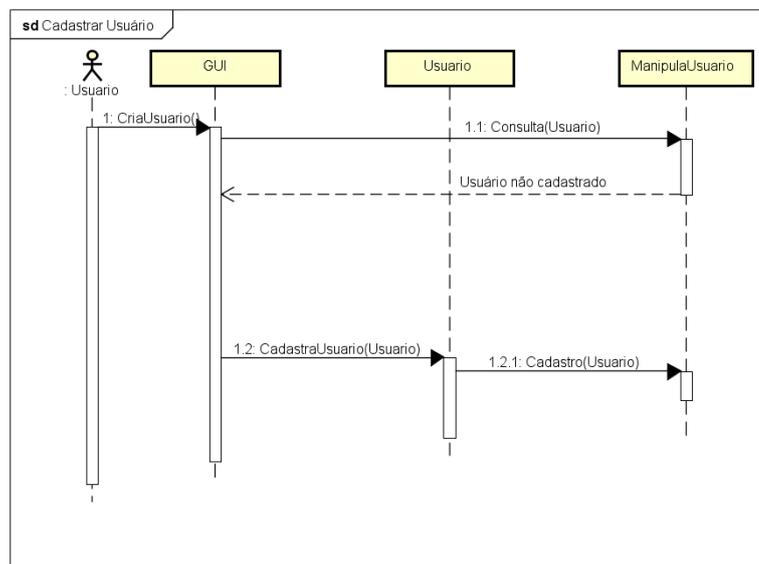
powered by Astah

Cadastrar eleitor



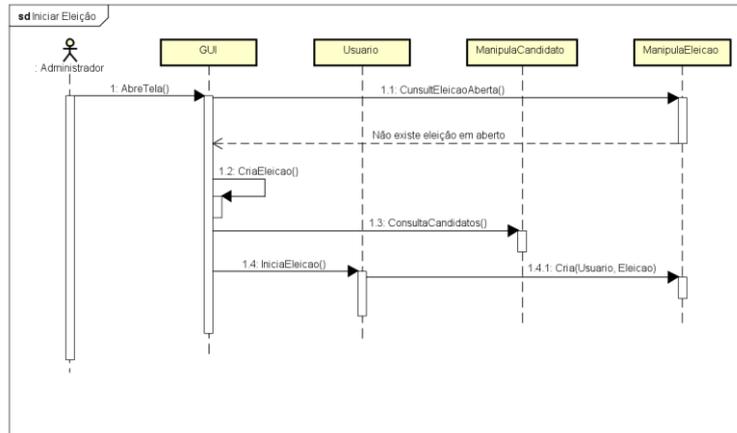
powered by Astah

Cadastrar usuário



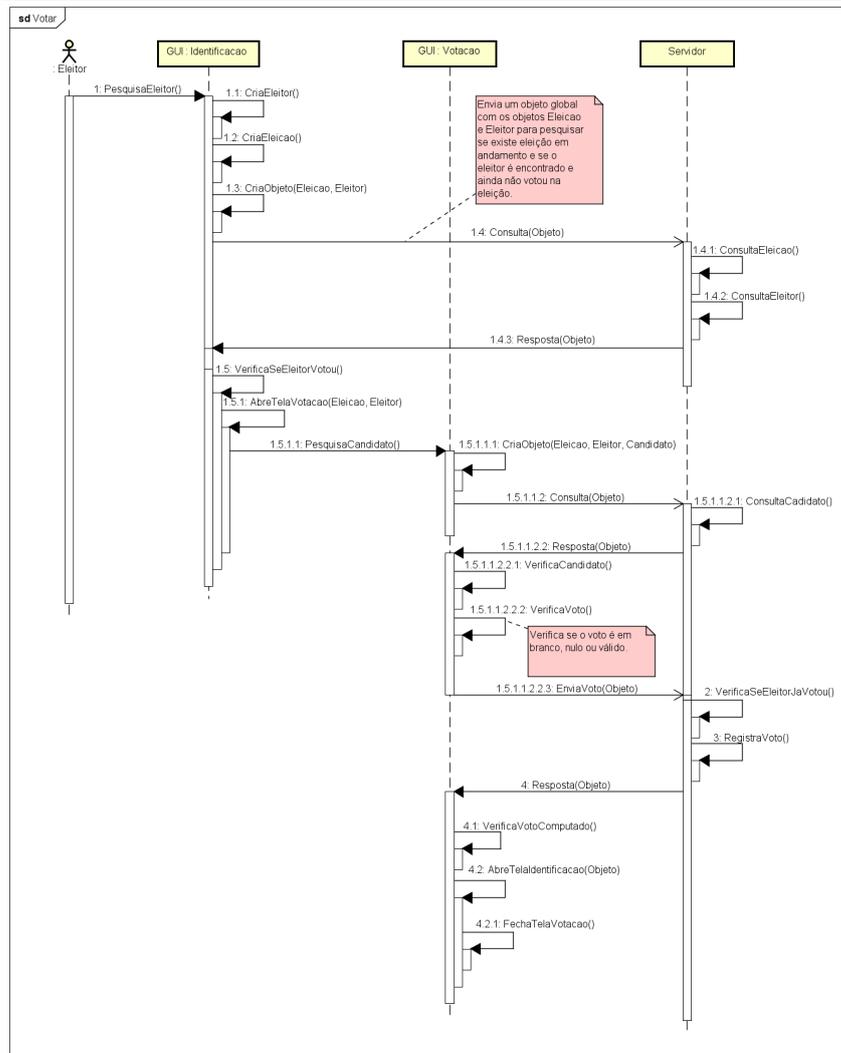
powered by Astah

Iniciar eleição



powered by Astah

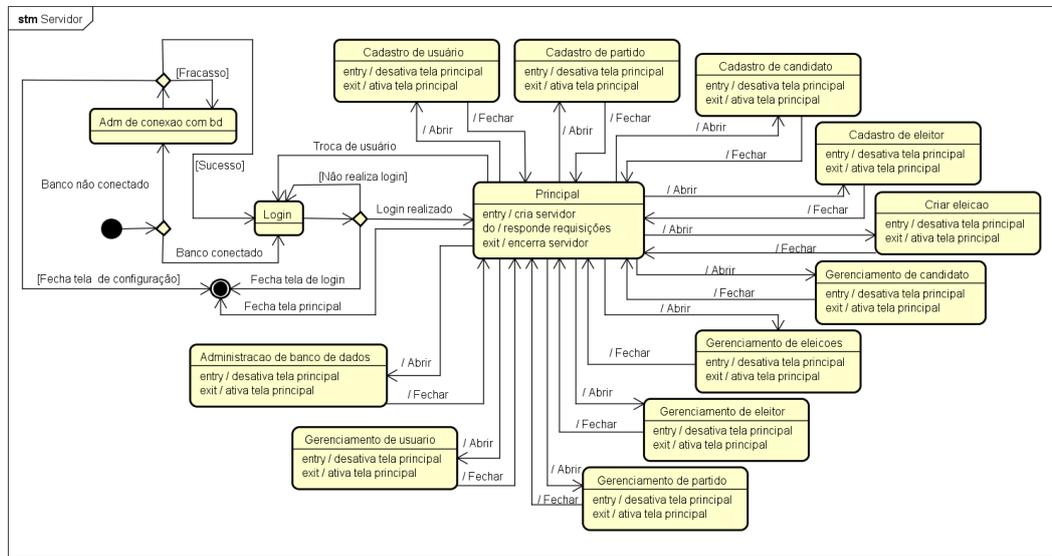
Votar



powered by Astah

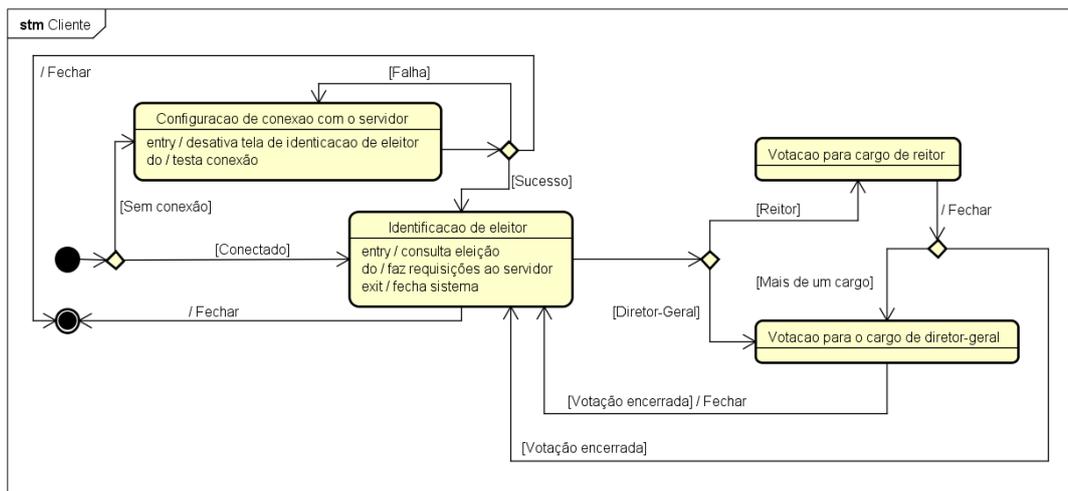
Diagrama de máquina de estado

Estado de interfaces gráficas (Servidor)



powered by Astah

Estado de interfaces gráficas (Cliente)



powered by Astah