

UNIVERSIDADE FEDERAL DO PARANÁ

ALISSON FRANCISCO KRUL
BRUNO HENRIQUE PEREIRA BRAGA
CARLOS AUGUSTO LIMA GRISPAN
LUCAS ERNESTO KINDINGER

URNA ELETRÔNICA

CURITIBA
JUNHO DE 2015

ALISSON KRUL
BRUNO HENRIQUE
CARLOS AUGUSTO
LUCAS ERNESTO KINDINGER

URNA ELETRÔNICA

Trabalho apresentado como requisito parcial
para aprovação na disciplina de Implementação de
Aplicação para Computador do curso de
Tecnologia em Análise e Desenvolvimento de
Sistemas da Universidade Federal do Paraná.

Orientador: Prof. Dr. Luiz Antônio Pereira Neves.

CURITIBA
JUNHO DE 2015

LISTA DE TERMOS, ABREVIATURAS E SIGLAS

Stakeholders – Partes interessadas ao projeto.

DBA – Administrador de Banco de Dados.

Popup – Janela de diálogo sobreposta à janela principal.

TADS – Tecnologia em Análise e Desenvolvimento de Sistemas.

LISTA DE ILUSTRAÇÕES

FIGURA 1 – LOGIN DO SISTEMA.....	8
FIGURA 2 – RECUPERAÇÃO DE SENHA.....	8
FIGURA 3 – CADASTRO DE USUÁRIO.....	9
FIGURA 4 – TELA INICIAL DO ADMINISTRADOR.....	10
FIGURA 5 – GERENCIAMENTO DE ELEITORES.....	10
FIGURA 6 – GERENCIAMENTO DE PARTIDOS.....	11
FIGURA 7 – CADASTRO DE PARTIDO.....	12
FIGURA 8 – GERENCIAMENTO DE ELEIÇÕES.....	13
FIGURA 9 – PERÍODO DAS ELEIÇÕES.....	14
FIGURA 10 – TIPO DE ELEIÇÃO.....	14
FIGURA 11 – VAGAS DA ELEIÇÃO MUNICIPAL.....	15
FIGURA 12 – VAGAS ELEIÇÃO ESTADUAL E FEDERAL.....	15
FIGURA 13 – GERENCIAMENTO DE CANDIDATOS DA ELEIÇÃO.....	16
FIGURA 14 – CADASTRO DE CANDIDATOS.....	16
FIGURA 15 – RELATÓRIO DE ELEITORES.....	17
FIGURA 16 – RELATÓRIO DO RESULTADO DA ELEIÇÃO.....	18
FIGURA 17 – URNA (LIMPA).....	19
FIGURA 18 – URNA (PREENCHIDA).....	19
FIGURA 19 – AGRADECIMENTO.....	20
FIGURA 20 – COMPROVANTE DE VOTAÇÃO.....	20
FIGURA 21 – FLUXO DE TELAS DO ELEITOR.....	21
FIGURA 22 – FLUXO DE TELAS DO ADMINISTRADOR.....	22
FIGURA 23 – MODELO CONCEITUAL.....	24
FIGURA 24 – MODELO LÓGICO.....	25
FIGURA 25 – DICIONÁRIO DE DADOS (PARTE 1 DE 2).....	27
FIGURA 26 – DICIONÁRIO DE DADOS (PARTE 2 DE 2).....	28
FIGURA 27 – ESQUEMA USUÁRIO.....	29
FIGURA 28 – ESQUEMA CEP.....	30
FIGURA 29 – ESQUEMA TICKET.....	30
FIGURA 30 – ESQUEMA CANDIDATO.....	31
FIGURA 31 – ESQUEMA DE ELEIÇÃO.....	32
FIGURA 32 – ESQUEMA TIPO.....	33

FIGURA 33 – ESQUEMA PARTIDO.....	33
FIGURA 34 – ESQUEMA VOTO.....	34
FIGURA 35 – ESQUEMA VAGA.....	35
FIGURA 36 – ESQUEMA ENDEREÇOS.....	35
FIGURA 37 – DIAGRAMA DE CONTEXTO.....	38
FIGURA 38 – NÍVEL ZERO.....	40
FIGURA 39 – GERENCIAR CANDIDATOS.....	41
FIGURA 40 – GERENCIAR PARTIDOS.....	42
FIGURA 41 – GERENCIAR USUÁRIOS.....	43
FIGURA 42 – GERENCIAR ELEIÇÕES.....	44
FIGURA 43 – COMPUTAR VOTO.....	45
FIGURA 44 – RECUPERAR SENHA.....	46
FIGURA 45 – VERIFICAR LOGIN.....	47
FIGURA 46 – SELEÇÃO DE USUÁRIO.....	53
FIGURA 47 – TELA DE LOGIN.....	53
FIGURA 48 – MENU INICIAL DO ADMINISTRADOR.....	54
FIGURA 49 – PESQUISA DE PARTIDOS.....	54
FIGURA 50 – PESQUISA DE CANDIDATO.....	55
FIGURA 51 – PESQUISA DE ELEITORES.....	55
FIGURA 52 – AGENDAMENTO DA VOTAÇÃO.....	56
FIGURA 53 – EMISSÃO DE RELATÓRIOS.....	56
FIGURA 54 - TABELA DE ERRO.....	57

SUMÁRIO

1	INTRODUÇÃO	6
2	PROTOTIPAÇÃO DE TELAS	7
2.1	PROTÓTIPOS.....	7
2.2	FLUXO DE TELAS.....	21
3	MODELAGEM DE DADOS	23
3.1	MODELO ENTIDADE RELACIONAMENTO	23
3.2	MODELO RELACIONAL	23
3.3	DICIONÁRIO DE DADOS	26
4	SCRIPT DE BANCO DE DADOS.....	29
4.1	TABELA USUARIO	29
4.2	TABELA CEP	30
4.3	TABELA TICKET.....	30
4.4	TABELA CANDIDATO	31
4.5	TABELA ELEICAO	32
4.6	TABELA TIPO	33
4.7	TABELA PARTIDO.....	33
4.8	TABELA VOTO	34
4.9	TABELA VAGA	35
4.10	TABELA ENDERECO	35
4.11	TABELA ERRO	36
4.12	ALTERAÇÕES EM TABELAS.....	37
5	DIAGRAMA DE FLUXO DE DADOS	38
5.1	DIAGRAMA DE CONTEXTO	38
5.2	DFD - NÍVEL ZERO	39
5.3	DFD - NÍVEL UM.....	41
5.3.1	GERENCIAR CANDIDATOS.....	41
5.3.2	GERENCIAR PARTIDOS	42

5.3.3	GERENCIAR USUÁRIOS.....	43
5.3.4	GERENCIAR ELEIÇÃO	44
5.3.5	COMPUTAR VOTO	45
5.3.6	RECUPERAR SENHA.....	46
5.3.7	VERIFICAR LOGIN	47
6	FONTES	48
7	CONCLUSÃO.....	51
8	APÊNDICE I – PROTÓTIPOS DE TELAS NÃO UTILIZADOS	53
9	APÊNDICE II – TABELA DE ERROS DO SISTEMA.....	57

1 INTRODUÇÃO

No âmbito acadêmico, exercitar o conhecimento através da implementação de projetos é, além de essencial, requisito indispensável para formar profissionais com visão sistêmica e capacidade de realizar conexão entre diferentes disciplinas. Com este objetivo dá-se a disciplina de Implementação de Aplicação para Computadores, a qual propôs este projeto de criação de uma aplicação web de uma urna eleitoral.

Apresentados os requisitos, cabe aos autores deste trabalho realizar as etapas subsequentes do desenvolvimento de software: revisar e realinhar os requisitos, desenvolver a prototipação de telas, modelar os dados, criar o script de dados, planejar como se dará o fluxo dos dados através das funções do sistema e, por fim, implementar o sistema.

Tal trabalho tem como objetivo a integração dos conhecimentos adquiridos até este momento no curso, possibilitando aos discentes e futuros Analistas e Desenvolvedores de Sistemas também expandir seus conhecimentos, buscando novas ferramentas e tecnologias que visam facilitar o desenvolvimento e melhorar a qualidade do produto final.

Ao final deste, esperam os autores terem adquirido uma visão completa das etapas de desenvolvimento de software, resolvendo assim o problema gerado a partir da segmentação do processo de desenvolvimento em diferentes disciplinas, assim como ter desenvolvido um produto que atenda à criteriosa avaliação da banca examinadora que espelha o exigente mercado que nos espera.

2 PROTOTIPAÇÃO DE TELAS

Parte importante do processo de desenvolvimento de software, a prototipação de telas é útil e necessária como parte do levantamento de requisitos pois a utilização desta ferramenta visual ajuda os analistas e os stakeholders a chegarem a um consenso de forma mais rápida e clara. Sua utilidade é também notada no processo de desenvolvimento do produto, pois auxilia o analista a ter uma visão sistêmica do software desejado.

Assim sendo, iniciamos este projeto de desenvolvimento de uma urna eletrônica pela prototipação de telas a qual apresentaremos a seguir. Separamos os protótipos de telas por usuário, ou seja, administrador e eleitor e por último apresentaremos um fluxo de telas para demonstrar como é feita a navegação pelo sistema.

Para a prototipação utilizamos computadores portáteis com sistemas operacionais Windows 7 e Windows 8 com processadores Intel i5 de segunda e terceira gerações. Tais computadores possuem entre quatro e seis gigabytes de memória principal. O software utilizado foi o Balsamic Mockups.

2.1 PROTÓTIPOS

Para iniciar representamos a tela de acesso ao sistema (FIGURA 1). Nesta tela os usuários, administrador ou eleitor, digitam os dados necessários para acesso ao sistema, previamente cadastrados. Tratando-se de eleitor os dados são CPF e senha. Já para o administrador é solicitado uma sequência de caracteres alfanuméricos iniciados pelo caractere especial “#”, que deve ser digitado no campo “CPF” e senha.

Nesta tela também há dois enlaces: “Esqueci minha senha” (FIGURA 2) e “Primeiro acesso” (FIGURA 3). O primeiro enlace abre um *popup* para a tela de recuperação de senha, enquanto o segundo abre um *popup* para a tela de cadastro de eleitor. O cadastro do administrador é inserido diretamente pelo DBA no banco de dados do sistema.

Administrador - Pesquisa

VOLTAR SAIR

VOTE BEM

CPF

Senha

[Esqueci minha senha](#)
[Primeiro acesso?](#)

LOGIN

FIGURA 1 – LOGIN DO SISTEMA.
FONTE: os Autores (2015).

A Web Page

Recuperação de senha

CPF

Enviar código para o e-mail

Insira código enviado por email: Código

Confirmar código

FIGURA 2 – RECUPERAÇÃO DE SENHA.
FONTE: os Autores (2015).

The image shows a web browser window with a title bar 'A Web Page' and a URL bar containing 'http://'. Inside the browser, there is a form titled 'Cadastro\Edição de Usuário' with a close button (X) in the top right corner. The form fields are as follows:

- Nome: [Text input]
- Titulo de Eleitor: [Text input]
- Zona: [Text input]
- Seção: [Text input]
- CPF: [Text input]
- Data de nascimento: [Date input with calendar icon]
- CEP: [Text input]
- Endereço: [Text input]
- Nº: [Text input]
- Complemento: [Text input]
- Bairro: [Text input]
- Cidade: [Text input]
- Estado: [Text input]
- Email: [Text input]
- Senha: [Text input]
- Confirmar Senha: [Text input]
- Enviar: [Blue button]

FIGURA 3 – CADASTRO DE USUÁRIO.
FONTE: os Autores (2015).

Na sequência, apresentamos os protótipos das telas do administrador, iniciando pelo menu inicial do administrador (FIGURA 4). Nesta tela são apresentadas as opções “Gerenciar Eleitores”, “Gerenciar Partidos” e “Gerenciar Eleições” que redirecionam o usuário para as respectivas telas.

Na tela de gerenciamento de eleitores (FIGURA 5), é possível realizar busca por eleitor através de parâmetros definidos pelo administrador do sistema. Esta busca retorna uma tabela com os dados dos eleitores. Clicando em alguma das linhas desta tabela, abre um *popup* (FIGURA 3) onde é possível visualizar ou excluir o cadastro do eleitor.

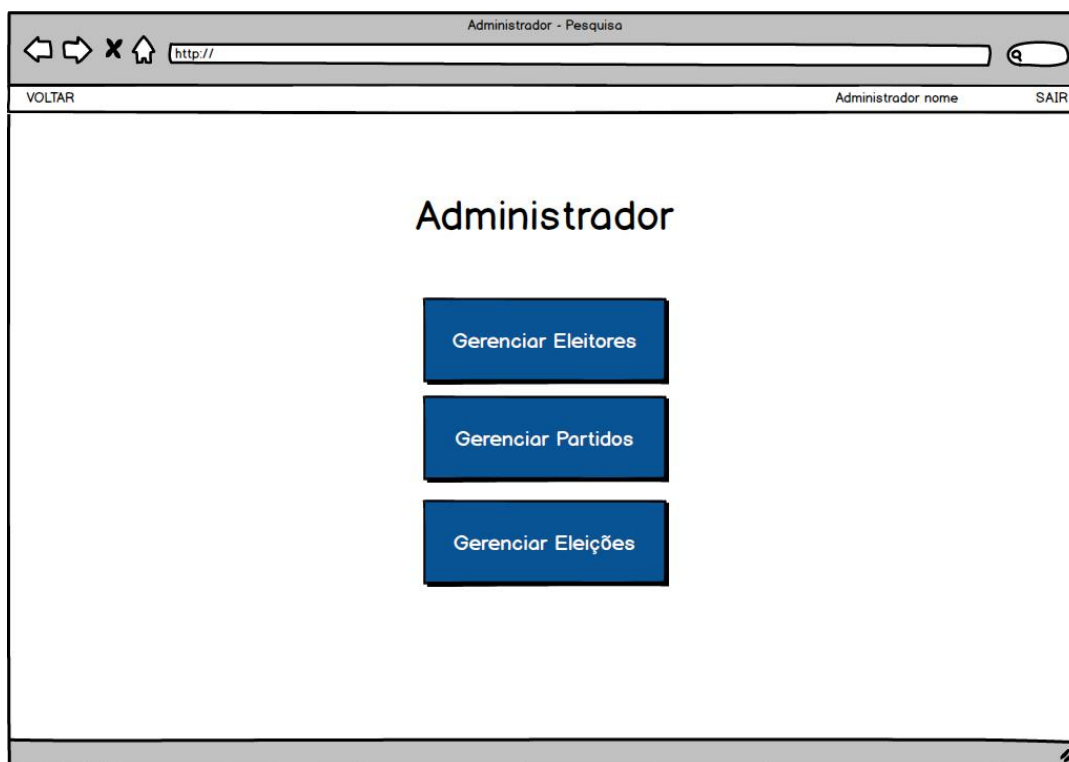


FIGURA 4 – TELA INICIAL DO ADMINISTRADOR.
FONTE: os Autores (2015).

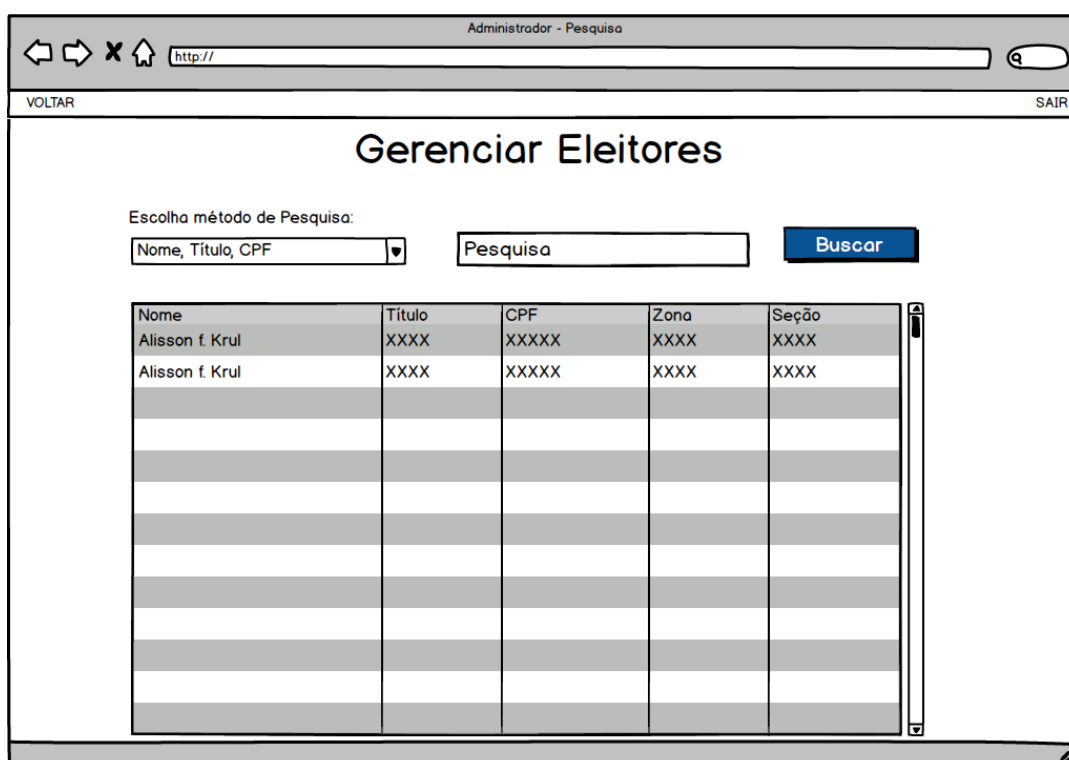


FIGURA 5 – GERENCIAMENTO DE ELEITORES.
FONTE: os Autores (2015).

Já na tela de gerenciamento de partidos (FIGURA 6), é possível realizar busca por partido através de parâmetros definidos pelo administrador do sistema. Esta busca retorna uma tabela com os dados dos partidos. Clicando em alguma das linhas dessa tabela, abre um *popup* (FIGURA 7) onde é possível visualizar ou excluir o cadastro do partido. Também há um botão “Novo Partido” que abre um *popup* (FIGURA 7) para a tela de cadastro de partido.

Administrador - Pesquisa

VOLTAR SAIR

Gerenciar Partidos

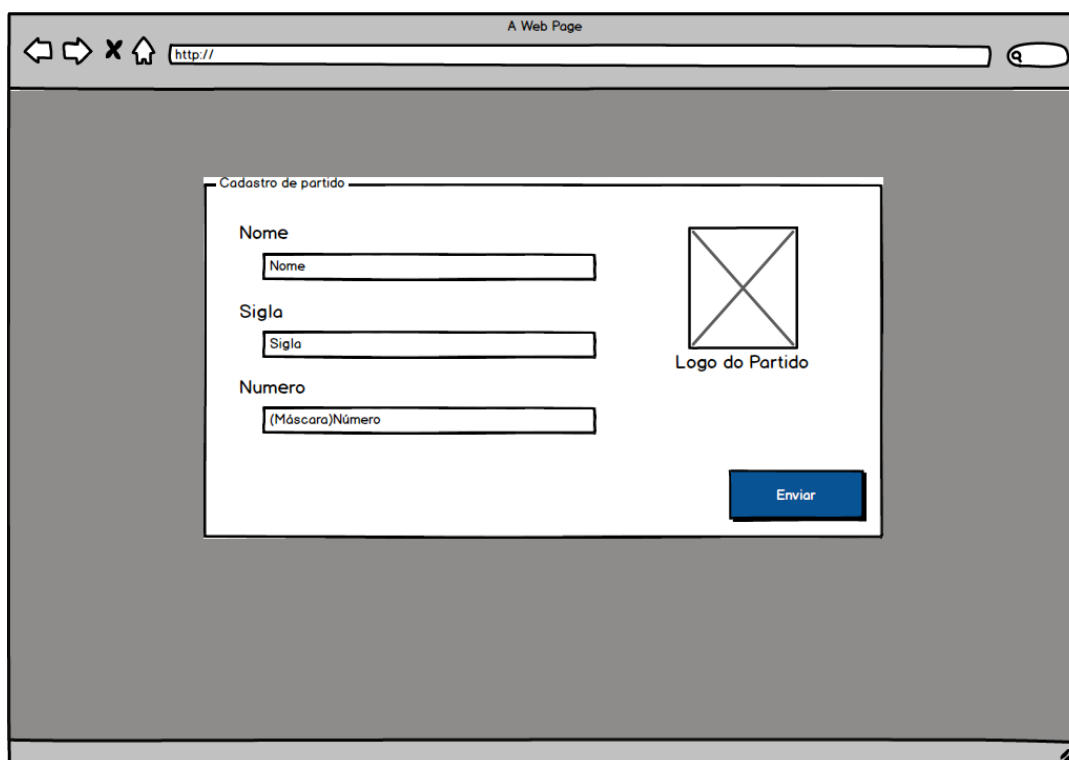
Escolha método de Pesquisa:

Sigla ou número Pesquisa

Buscar Adicionar Partido

Sigla	Número	Nome
P1	99	Partido1
P2	87	Partido2

FIGURA 6 – GERENCIAMENTO DE PARTIDOS.
FONTE: os Autores (2015).



The image shows a web browser window titled 'A Web Page' with a search bar containing 'http://'. The main content area displays a form titled 'Cadastro de partido'. The form contains three input fields: 'Nome' (with a placeholder 'Nome'), 'Sigla' (with a placeholder 'Sigla'), and 'Número' (with a placeholder '(Máscara)Número'). To the right of these fields is a placeholder for a logo, represented by a square with an 'X' inside, labeled 'Logo do Partido'. At the bottom right of the form is a blue button labeled 'Enviar'.

FIGURA 7 – CADASTRO DE PARTIDO.
FONTE: os Autores (2015).

A tela gerenciamento de eleições (FIGURA 8) permite realizar busca por eleição através de parâmetros definidos pelo administrador do sistema. Essa busca retorna uma tabela com os dados das eleições. Clicando em alguma das linhas dessa tabela, abre um *popup* onde pode: visualizar, alterar ou excluir a eleição, caso o estado da mesma seja “agendada” (FIGURA 9) ou visualizar relatórios caso o estado seja “finalizada” (FIGURA 15 e 16). Se o estado for “iniciada”, nenhuma ação é disponibilizada. Também há um botão “Nova Eleição” que abre uma sequência de *popups* administrador do sistema para a tela de cadastro de eleição.

Administrador - Pesquisa

VOLTAR SAIR

Gerenciar Eleições

Escolha método de Pesquisa:

Código, tipo, data, status Pesquisa Buscar Adicionar Eleição

cod	tipo	Data	Início	Fim
XX	XXXXXX	XX/XX/XXXX	XX:XX	XX:XX
XX	XXXXXX	XX/XX/XXXX	XX:XX	XX:XX
XX	XXXXXX	XX/XX/XXXX	XX:XX	XX:XX
XX	XXXXXX	XX/XX/XXXX	XX:XX	XX:XX

FIGURA 8 – GERENCIAMENTO DE ELEIÇÕES.
FONTE: os Autores (2015).

Os protótipos das telas “Período” (FIGURA 9), “Tipo” (FIGURA 10), “Vagas Eleição Municipal” (FIGURA 11), “Vagas Eleição Estadual e Federal”, (FIGURA 12) e “Candidatos” (FIGURA 13) demonstram as telas que possibilitam a entrada de dados referentes ao cadastro de nova eleição. A navegação entre essas telas se dá através de enlaces “Próximo” e “Voltar”. Na primeira o administrador do sistema determina a data e o horário de início e finalização das votações. Após, escolhe entre eleição municipal ou estadual e federal. Tal escolha determina os tipos de cargos que serão votados na eleição assim como a tela que o sistema deve seguir (FIGURA 11 ou 12), na qual o número de vagas para cada cargo é definido. Na tela “Candidatos” é possível realizar busca por candidato através de parâmetros definidos pelo administrador do sistema. Esta busca retorna uma tabela com os dados dos candidatos. Clicando em uma linha desta tabela o administrador abre um *popup* (FIGURA 14) onde é possível visualizar ou excluir o cadastro do candidato. Também terá um botão “Novo Candidato” que abre um *popup* (FIGURA 14) para a tela de cadastro de candidato.

A screenshot of a web browser window titled "A Web Page". The address bar shows "http://". The main content area is gray. In the center, there is a white modal window titled "Eleição - Período" with a close button (X) in the top right corner. The modal contains the following fields:

- Data: A date input field with slashes (//) and a calendar icon.
- Hora Início: A time input field showing "00:00".
- Hora Finalização: A time input field showing "00:00".

At the bottom of the modal, there are two buttons: "Voltar" (light gray) and "Próximo" (dark blue).

FIGURA 9 – PERÍODO DAS ELEIÇÕES.
FONTE: os Autores (2015).

A screenshot of a web browser window titled "A Web Page". The address bar shows "http://". The main content area is gray. In the center, there is a white modal window titled "Eleição - Tipo" with a close button (X) in the top right corner. The modal contains the following fields:

- Tipo: A dropdown menu with the text "Escolha um Tipo" and a downward arrow. The dropdown is open, showing two options: "Federal e Estadual" and "Municipal".

At the bottom of the modal, there are two buttons: "Voltar" (light gray) and "Próximo" (dark blue).

FIGURA 10 – TIPO DE ELEIÇÃO.
FONTE: os Autores (2015).

A Web Page

http://

Eleição - Vagas para Eleição Municipal

Prefeito: ☐ 1 Vaga

Vereadores: Vagas

Voltar Próxímo

FIGURA 11 – VAGAS DA ELEIÇÃO MUNICIPAL.
FONTE: os Autores (2015).

A Web Page

http://

Eleição - Vagas para Eleição Estadual e Federal

Presidente: ☐ 1 Vaga

Governador: ☐ 1 Vaga

Deputados Federais: Vagas

Deputados Estaduais: Vagas

Senadores: Vagas

Voltar Agendar eleição

FIGURA 12 – VAGAS ELEIÇÃO ESTADUAL E FEDERAL.
FONTE: os Autores (2015).

The screenshot shows a web browser window titled 'Administrador - Pesquisa'. The address bar shows 'http://'. The page has a navigation bar with 'VOLTAR' on the left and 'SAIR' on the right. The main heading is 'Gerenciar Candidatos'. Below the heading, there is a section for search options: 'Escolha método de Pesquisa:' followed by a dropdown menu showing 'Nome, Número, Partido' and a search input field labeled 'Pesquisa'. To the right of these are two buttons: 'Buscar' and 'Adicionar Candidato'. Below the search section is a table with four columns: 'Cargo', 'Número', 'Partido', and 'Nome'. The table contains five rows of data, with the first row having a grey background. The data rows are as follows:

Cargo	Número	Partido	Nome
Prefeito	99	Partido1	Nome_nome
Deputado	99123	Partido1	Nome_nome
Vereador	97122	Partido2	Nome_nome
Vereador	99545	Partido1	Nome_nome

FIGURA 13 – GERENCIAMENTO DE CANDIDATOS DA ELEIÇÃO.
FONTE: os Autores (2015).

The screenshot shows a web browser window titled 'A Web Page'. The address bar shows 'http://'. The page has a navigation bar with 'VOLTAR' on the left and 'SAIR' on the right. The main heading is 'Cadastro de candidato'. Below the heading, there is a form with four input fields: 'Cargo' (a dropdown menu), 'Nome' (a text input), 'Partido' (a dropdown menu), and 'Numero' (a text input with a placeholder '(Máscara)Número'). To the right of the form is a placeholder for a photo, represented by a square with an 'X' inside, labeled 'Foto 3x4'. Below the photo placeholder is a blue button labeled 'Enviar'.

FIGURA 14 – CADASTRO DE CANDIDATOS.
FONTE: os Autores (2015).

No protótipo da tela “Relatório de Eleitores” (FIGURA 15) uma lista é mostrada com dados de todos os eleitores que votaram. Já na tela “Relatório de Resultados” (FIGURA 16) é apresentado o número total de votos, os votos válidos, brancos e nulos e uma lista com todos os dados compilados sobre os candidatos participantes de determinada eleição e a respectiva quantidade de votos recebidos.

Nome	Título	CPF	Zona	Sessão	Data Eleição	Hora Voto
Alisson f. Krul	XXXX	XXXXXX	xXXX	XXXXXX	XX/XX/XXX	XX:XX
Alisson f. Krul	XXXX	XXXXXX	xXXX	XXXXXX	XX/XX/XXX	XX:XX

FIGURA 15 – RELATÓRIO DE ELEITORES.
FONTE: os Autores (2015).

Administrador - Pesquisa

VOLTAR SAIR

Relatório - Resultado

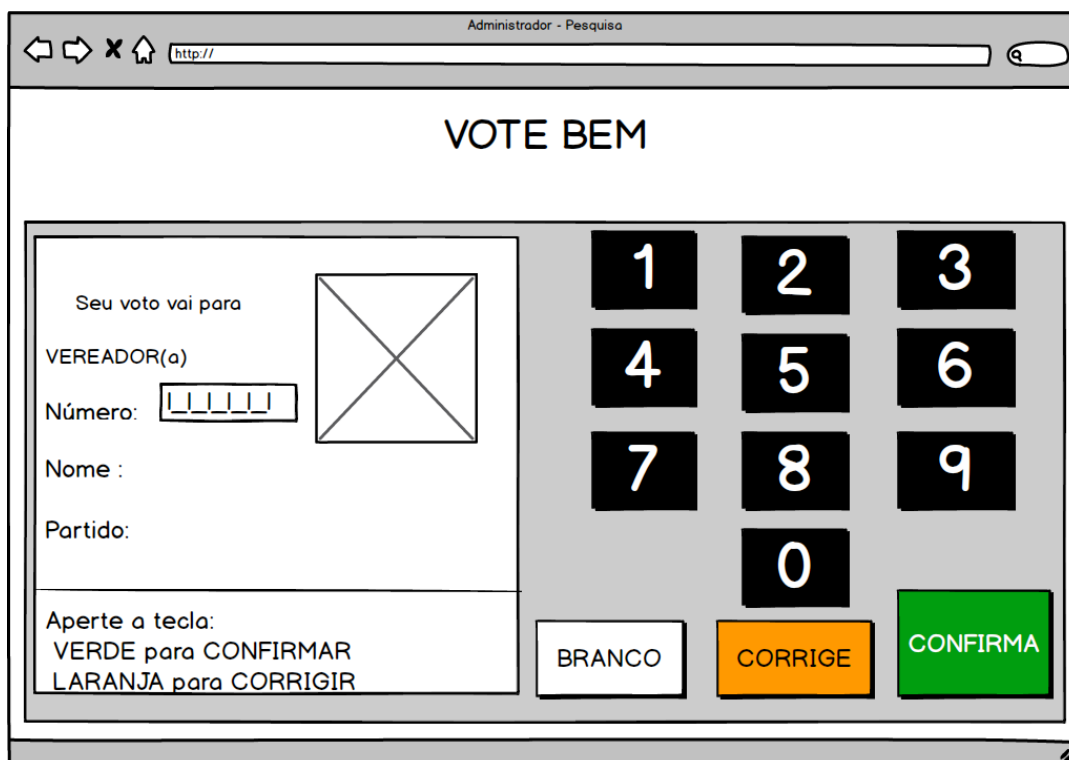
Cargo	Número	Partido	Nome	Status	Quantidade de Votos
Prefeito	99	Partido1	Nome_nome	Eleito	10023
Prefeito	88	Partido1	Nome_nome	Não Eleito	10022
Vereador	97122	Partido2	Nome_nome	Eleito	1382
Vereador	99545	Partido1	Nome_nome	Eleito	1120

FIGURA 16 – RELATÓRIO DO RESULTADO DA ELEIÇÃO.
 FONTE: os Autores (2015).

Realizada a prototipação de telas do administrador, apresentamos a seguir os protótipos das telas referentes ao acesso e utilização do sistema por parte dos eleitores.

Depois de realizado o acesso ao sistema, é apresentado ao eleitor a tela “Urna Vazia” (FIGURA 17), e conforme o eleitor digita o número seu candidato a urna toma outra aparência (FIGURA 18). Esta sequência de telas é repetida conforme o número de cargos a serem votados.

Concluída a votação é exibida a tela representada pelo protótipo denominado “Agradecimento” (FIGURA 19) que terá um botão “Imprimir Comprovante” que, quando clicado, gera um PDF (FIGURA 20), encerrando a navegação do eleitor pelo sistema e das telas abordadas por este capítulo.



Administrador - Pesquisa

http://

VOTE BEM

Seu voto vai para

VEREADOR(a)

Número:

Nome :

Partido:

Aperte a tecla:
VERDE para CONFIRMAR
LARANJA para CORRIGIR

1 2 3

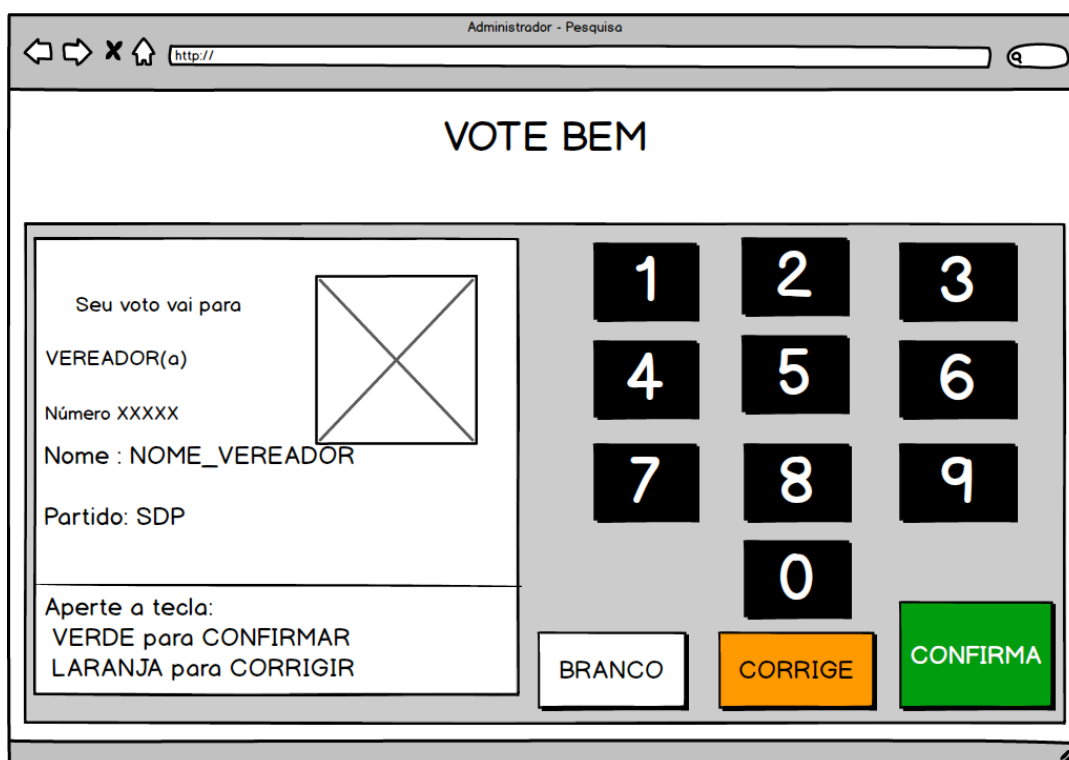
4 5 6

7 8 9

0

BRANCO CORRIGE CONFIRMA

FIGURA 17 – URNA (LIMPA).
FONTE: os Autores (2015).



Administrador - Pesquisa

http://

VOTE BEM

Seu voto vai para

VEREADOR(a)

Número XXXXX

Nome : NOME_VEREADOR

Partido: SDP

Aperte a tecla:
VERDE para CONFIRMAR
LARANJA para CORRIGIR

1 2 3

4 5 6

7 8 9

0

BRANCO CORRIGE CONFIRMA

FIGURA 18 – URNA (PREENCHIDA).
FONTE: os Autores (2015).

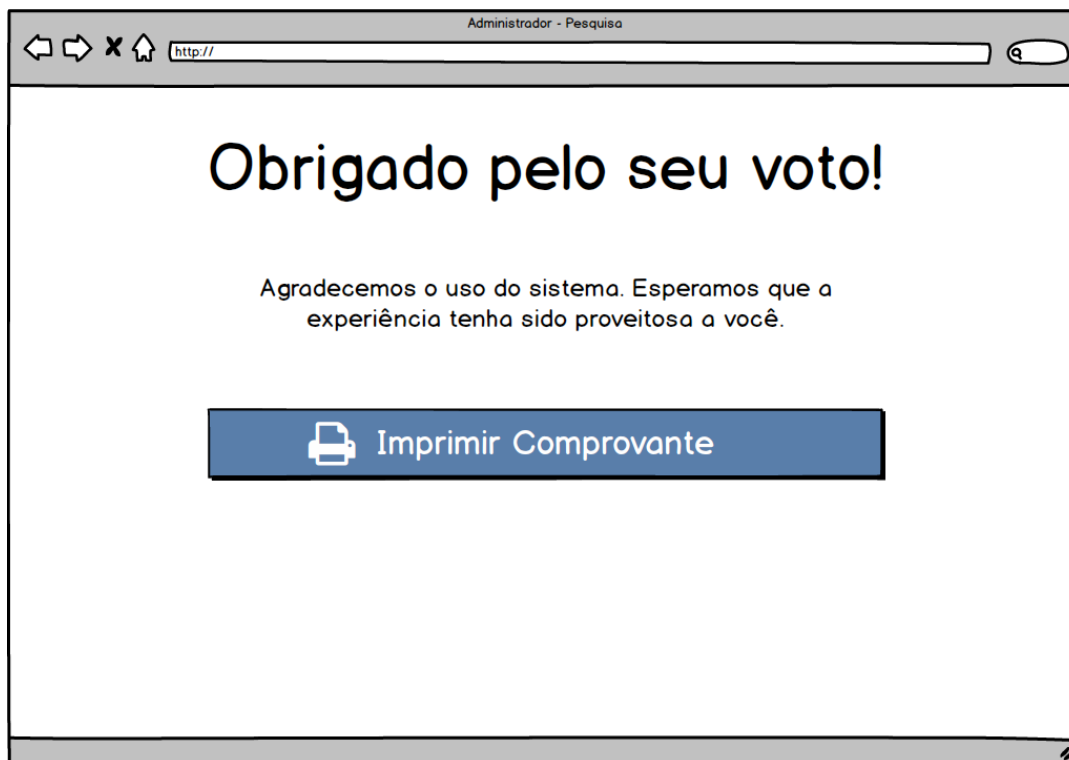


FIGURA 19 – AGRADECIMENTO.
FONTE: os Autores (2015).

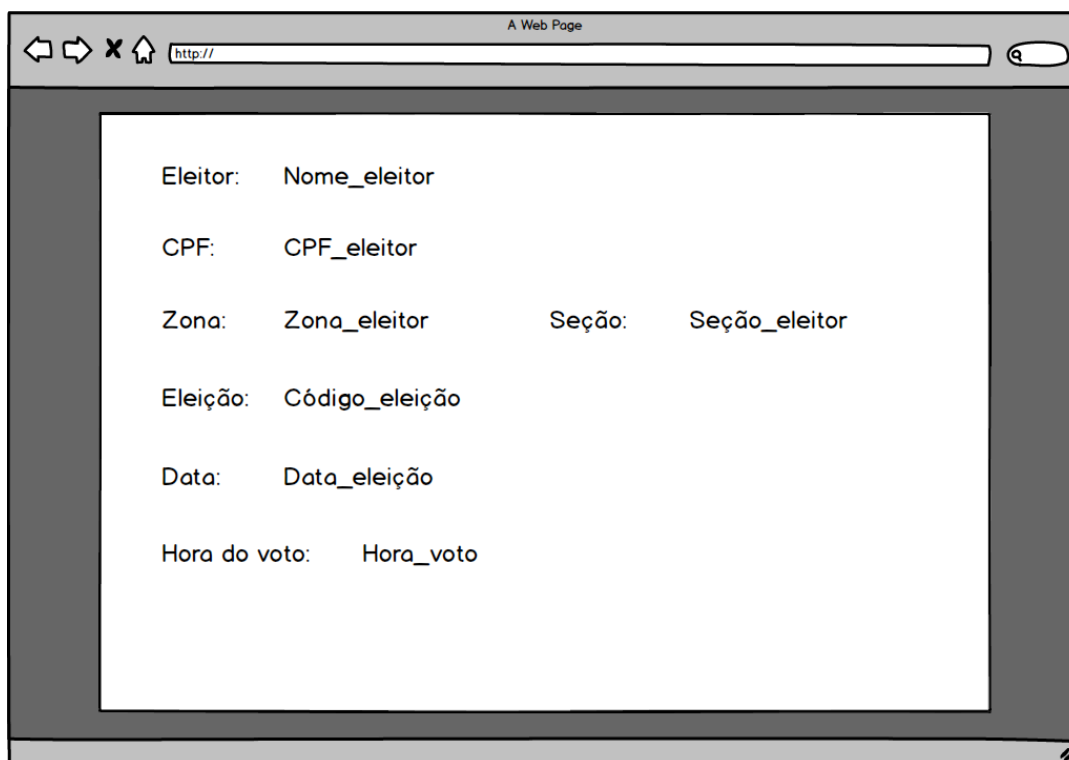


FIGURA 20 – COMPROVANTE DE VOTAÇÃO.
FONTE: os Autores (2015).

2.2 FLUXO DE TELAS

Para facilitar o entendimento de como se dá a navegação dentro do sistema, apresentamos abaixo o fluxo de telas, divididos entre “Fluxo de Telas do Eleitor” (FIGURA 21) e “Fluxo de Telas do Administrador” (FIGURA 22).

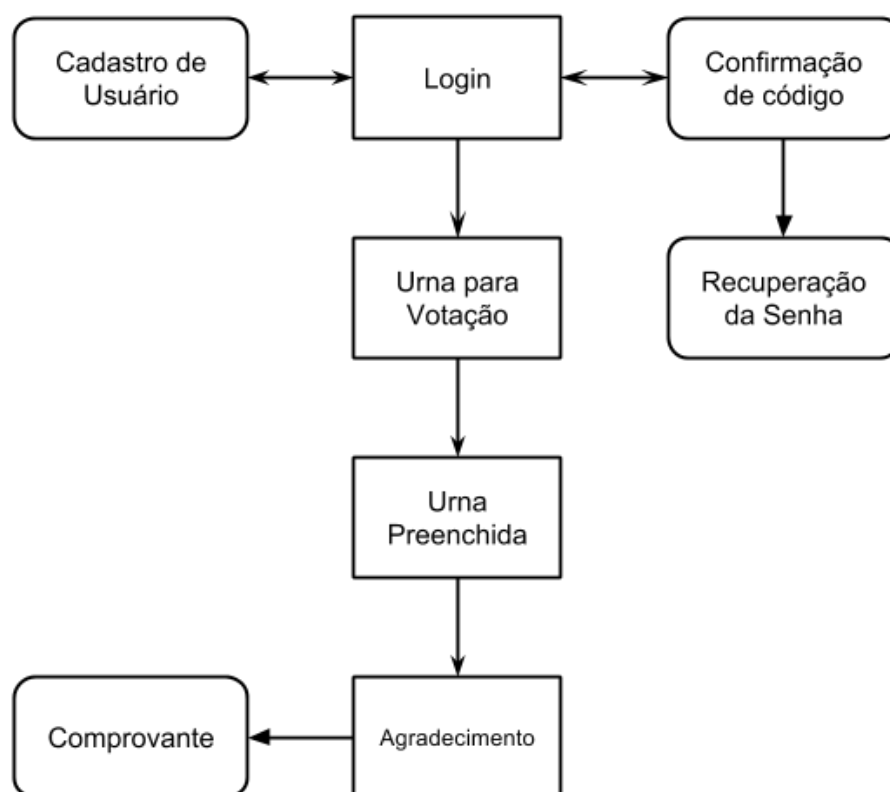


FIGURA 21 – FLUXO DE TELAS DO ELEITOR.
FONTE: os Autores (2015).

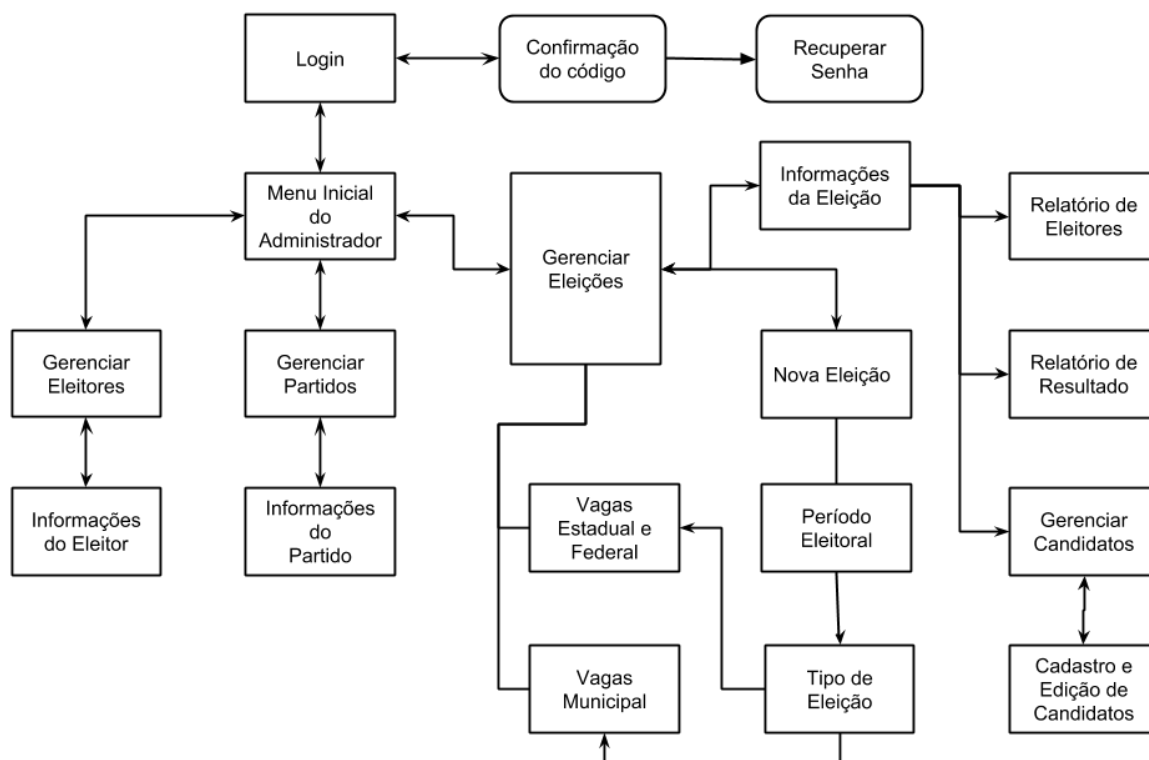


FIGURA 22 – FLUXO DE TELAS DO ADMINISTRADOR.
FONTE: os Autores (2015).

Como se espera notar com a observação dos protótipos das telas apresentadas, o sistema tem como base a facilidade de utilização, fazendo com que a experiência do usuário, administrador ou eleitor, seja intuitiva, simples e amigável.

Além disso, a prototipação do sistema auxiliou na compreensão do mesmo, explicitando aos analistas as características e funcionalidades que o sistema deve ter, permitindo assim melhor organização e planejamento do processo de implementação.

Assim como em outras partes do projeto, a prototipação sofreu alterações desde a primeira versão até a aqui apresentada, como é possível observar analisando as telas mostradas no APÊNDICE I – PROTÓTIPOS DE TELAS NÃO UTILIZADOS que, se comparadas com as atuais, diferem em conteúdo e qualidade, evidenciando a evolução do projeto.

3 MODELAGEM DE DADOS

Próximo passo do processo de desenvolvimento, é através da modelagem dos dados que se compreende quais dados são necessários para o bom funcionamento do sistema, assim como quais dados são gerados a partir do mesmo.

A análise dos dados e o estudo de sua forma de armazenamento são fundamentais para que um sistema seja confiável e eficiente, evitando redundâncias e lentidão com buscas em bancos de dados mal projetados.

Com tal objetivo iniciamos, com o auxílio do software BrModelo, a criação do Modelo Entidade e Relacionamento, gerando a partir dele o Modelo Relacional, os quais apresentamos a seguir. Este software também foi utilizado para auxiliar na criação do *script* de dados, que trataremos no capítulo seguinte.

3.1 MODELO ENTIDADE RELACIONAMENTO

Este modelo (FIGURA 23) serve como representação conceitual do modelo de dados a ser implementado. É através desta abstração que se identifica, de forma simplificada e de fácil compreensão, como os dados manipulados pelo sistema estão relacionados entre si e qual é a forma mais otimizada de armazená-los.

3.2 MODELO RELACIONAL

Gerado a partir do modelo entidade relacionamento, o modelo relacional (FIGURA 24) caracteriza-se por ser uma abstração mais próxima do modelo físico implementado, representando os esquemas gerados a partir das entidades e seus atributos, chaves primárias e estrangeiras, relacionamentos e suas cardinalidades. É a partir deste modelo que é criado o *script* de dados, como veremos no próximo capítulo.

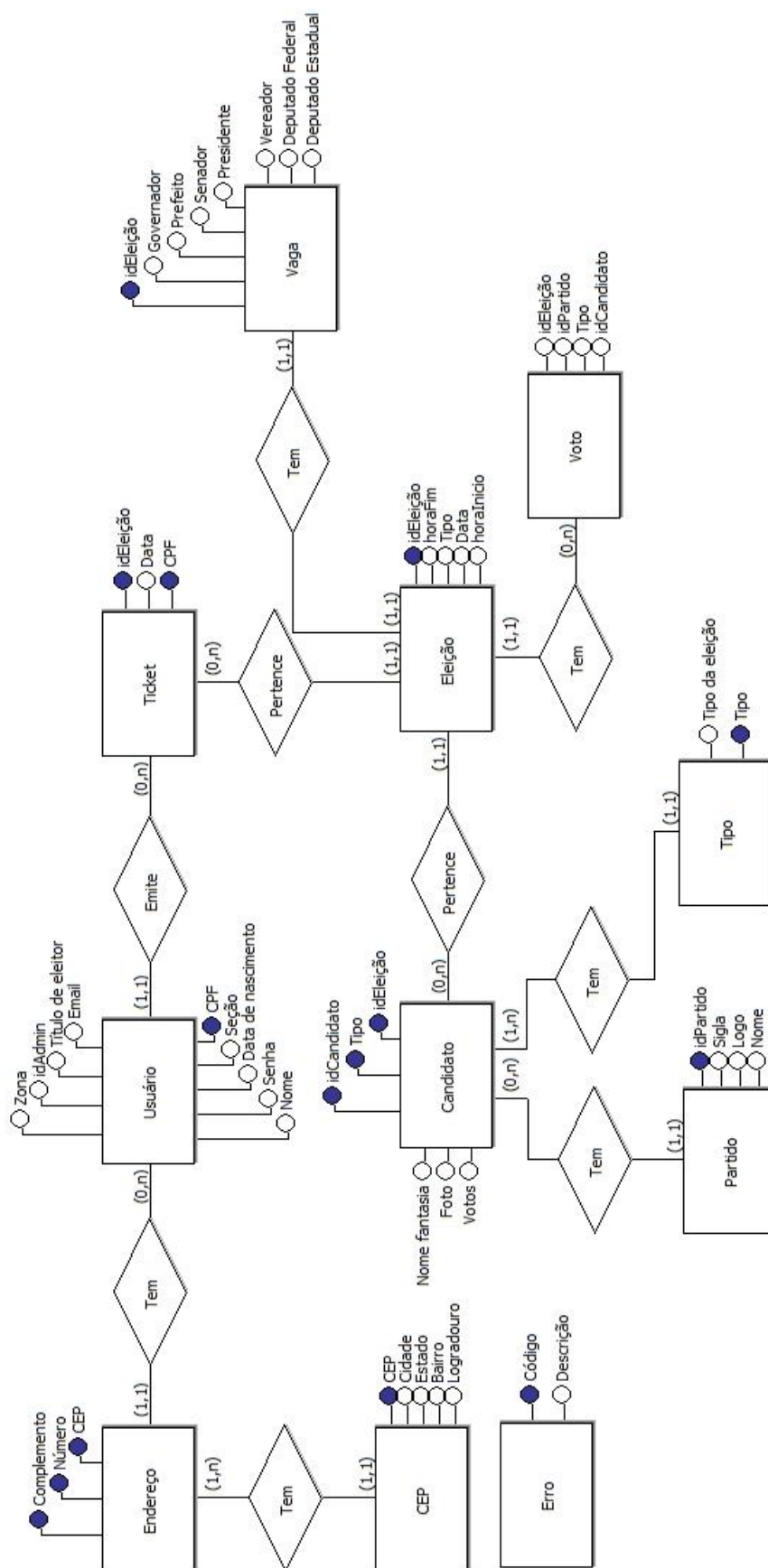


FIGURA 23 – MODELO CONCEITUAL.
FONTE: os Autores (2015).

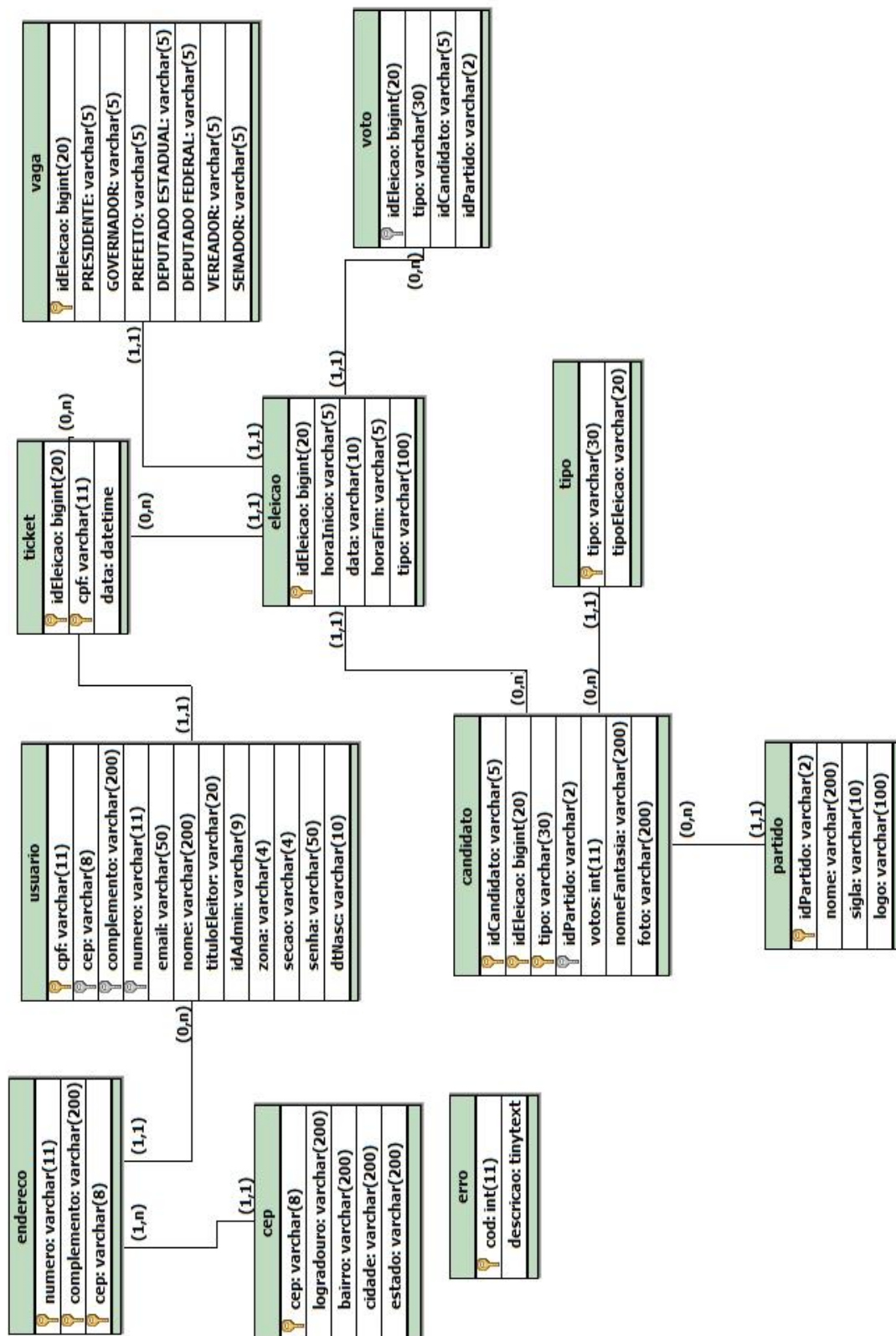


FIGURA 24 – MODELO LÓGICO.
FONTE: os Autores (2015).

Analisando os modelos apresentados, verificamos que temos um banco de dados composto por duas diferentes estruturas, uma de armazenamento dos dados do usuário e de logs do acesso ao sistema por este e outra de armazenamento dos dados do processo eleitoral como um todo.

Na primeira estrutura, a entidade principal, “USUARIOS”, relaciona-se com outras duas entidades: “ENDERECOS”, entidade criada como forma de evitar redundância de dados visto que um mesmo endereço pode abrigar mais de um usuário e “TICKET”, que armazena unicamente a data e o horário em que o usuário realizou acesso ao sistema. Com a mesma justificativa de “ENDERECOS”, criou-se a entidade “CEP”, evitando assim a repetição desnecessária de um conjunto de dados em comum.

Na segunda estrutura, mais complexa, foram criadas as entidades: “PARTIDOS” e “CANDIDATOS” para identificar os candidatos aos cargos políticos em disputa e suas respectivas afiliações, assim como a entidade “TIPOS” para determinar a qual cargo cada um concorrerá.

A entidade “ELEICOES” foi criada para armazenar dados pertinentes a cada processo eleitoral possibilitando a geração de relatórios específicos pelo administrador do sistema e o armazenamento dos dados ao longo do tempo.

Por último, a entidade “VOTOS”, contrariando a premissa de evitar redundância, visando segurança eleitoral, foi criada com o único objetivo de existir um meio de conferir se o número total de votos computados é compatível com a somatória do número de votos computados a cada candidato, mais votos brancos e nulos.

3.3 DICIONÁRIO DE DADOS

Tendo em vista que um sistema possui vasta gama de dados que são persistidos no banco de dados, faz-se necessário o uso do dicionário de dados (FIGURA 25). Este possui a função de armazenar os metadados dos esquemas, ou seja, guardar as informações referentes aos campos em que os dados são armazenados, como o nome, a regra de restrição, seu tipo e tamanho, facilitando assim o entendimento do sistema.

USUARIO				
Campo	Restrição	Tipo	Tamanho	Descrição
cpf	PK	VARCHAR	11	CPF do usuário
cep	FK	VARCHAR	8	CEP do usuário
numero	FK	VARCHAR	11	Nº residencia do usuário
complemento	FK	VARCHAR	200	Complemento de endereço do usuário
email		VARCHAR	50	Email do usuário
nome		VARCHAR	200	Nome do usuário
tituloEleitor		VARCHAR	20	Título do usuário
zona		VARCHAR	4	Zona eleitoral
secao		VARCHAR	4	Seção eleitoral
senha		VARCHAR	50	Senha de acesso
dtNasc		VARCHAR	10	dd/mm/aaaa
idAdmin		VARCHAR	9	Identificação do administrador do sistema
CANDIDATO				
Campo	Restrição	Tipo	Tamanho	Descrição
idCandidato	PK	VARCHAR	5	Identificação do candidato
idEleicao	PK+FK	BIGINT	20	Identificação da eleição
tipo	PK+FK	VARCHAR	30	Identificação do cargo do candidato
idPartido	FK	VARCHAR	2	Identificação do partido do candidato
nomeFantasia		VARCHAR	200	Nome do candidato na urna
votos		INTEGER	11	Quantidade de votos
foto		VARCHAR	200	Endereço do arquivo que contém a foto do candidato
ELEICAO				
Campo	Restrição	Tipo	Tamanho	Descrição
idEleicao	PK	BIGINT	20	Código de identificação da eleição
horaInicio		VARCHAR	5	Hora de início da eleição
horaFim		VARCHAR	5	Hora de finalização da eleição
data		VARCHAR	10	Data da eleição
tipo		VARCHAR	100	Identifica se a eleição é municipal, estadual ou federal
PARTIDO				
Campo	Restrição	Tipo	Tamanho	Descrição
idPartido	PK	VARCHAR	2	Identificação do partido
nome		VARCHAR	200	Nome do partido
sigla		VARCHAR	10	Sigla do partido
logo		VARCHAR	100	Identificação do endereço que contém o logo do partido
CEP				
Campo	Restrição	Tipo	Tamanho	Descrição
cep	PK	VARCHAR	8	CEP do usuário
logradouro		VARCHAR	200	Logradouro do usuário
bairro		VARCHAR	200	Bairro do usuário
cidade		VARCHAR	200	Cidade do usuário
estado		VARCHAR	200	Estado do usuário

FIGURA 25 – DICIONÁRIO DE DADOS (PARTE 1 DE 2).

FONTE: os Autores (2015).

ENDERECO				
Campo	Restrição	Tipo	Tamanho	Descrição
numero	PK	VARCHAR	11	Número da casa do usuário
complemento	PK	VARCHAR	200	Complemento de endereço do usuário
cep	PK+FK	VARCHAR	8	CEP do usuário
VAGA				
Campo	Restrição	Tipo	Tamanho	Descrição
idEleicao	PK+FK	BIGINT	20	Código para identificação da eleição
PRESIDENTE		VARCHAR	5	Qtde Vagas p/ Presidente
GOVERNADOR		VARCHAR	5	Qtde Vagas p/ Governador
PREFEITO		VARCHAR	5	Qtde Vagas p/ Prefeito
DEPUTADO ESTADUAL		VARCHAR	5	Qtde Vagas p/ Deputado Estadual
DEPUTADO FEDERAL		VARCHAR	5	Qtde Vagas p/ Deputado Federal
VEREADOR		VARCHAR	5	Qtde Vagas p/ Vereador
SENADOR		VARCHAR	5	Qtde Vagas p/ Senador
TIPO				
Campo	Restrição	Tipo	Tamanho	Descrição
tipo	PK	VARCHAR	30	Cargo exercido
tipoEleicao		VARCHAR	20	Indica se o cargo é de uma Eleição Federal ou Municipal
TICKET				
Campo	Restrição	Tipo	Tamanho	Descrição
idEleicao	PK+FK	BIGINT	20	Indica de qual eleição o ticket pertence
cpf	PK+FK	VARCHAR	11	CPF do usuario
data		DATETIME		Data e hora da votação
VOTO				
Campo	Restrição	Tipo	Tamanho	Descrição
idEleicao	FK	BIGINT	20	Eleição a qual o será computado o voto
tipo		VARCHAR	30	Cargo que foi votado
idPartido		VARCHAR	2	Indica o partido foi votado
idCandidato		VARCHAR	5	Código para identificação do candidato
ERRO				
Campo	Restrição	Tipo	Tamanho	Descrição
cod	PK	INTEGER	11	Define qual o código do erro a ser retornado
descricao		TINYTEXT		É a descrição do erro em si.

FIGURA 26 – DICIONÁRIO DE DADOS (PARTE 2 DE 2).

FONTE: os Autores (2015).

4 SCRIPT DE BANCO DE DADOS

Gerado a partir do modelo de dados apresentado no último capítulo, este *script* foi criado na Linguagem de Consulta Estruturada, mais conhecida como SQL e tem como objetivo a criação do banco de dados utilizado por esta aplicação.

Foram tomados os devidos cuidados na criação de chaves primárias e estrangeiras a fim de evitar redundâncias e garantir a integridade referencial.

Apresentamos a seguir os *scripts* e uma representação visual do das tabelas por eles geradas.

4.1 TABELA USUARIO

Esta tabela contém os dados de todos os usuários do sistema, seja eleitor ou administrador do sistema.

```
CREATE TABLE IF NOT EXISTS `usuarios` (
  `cpf` varchar(11) NOT NULL DEFAULT '',
  `numero` varchar(11) DEFAULT NULL,
  `email` varchar(50) DEFAULT NULL,
  `cep` varchar(8) DEFAULT NULL,
  `complemento` varchar(200) NOT NULL DEFAULT '',
  `nome` varchar(200) DEFAULT NULL,
  `tituloEleitor` varchar(20) DEFAULT NULL,
  `idAdmin` varchar(10) DEFAULT NULL,
  `zona` varchar(4) DEFAULT NULL,
  `secao` varchar(4) DEFAULT NULL,
  `senha` varchar(50) DEFAULT NULL,
  `dtNasc` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`cpf`),
  UNIQUE (`tituloEleitor`),
  KEY `fkUsuariosEnderecos` (`numero`,`cep`,`complemento`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

USUARIO												
cpf	cep	numero	complemento	email	nome	tituloEleitor	idAdmin	zona	secao	senha	dtNasc	idAdmin

FIGURA 27 – ESQUEMA USUÁRIO.
FONTE: os Autores (2015).

4.2 TABELA CEP

Esta tabela contém os dados referentes ao CEP dos endereços de usuários e foi criada pois inúmeros endereços possuem o mesmo CEP e, assim sendo, seria redundante repetir esta informação em cada instância de um endereço.

```
CREATE TABLE IF NOT EXISTS `cep` (
  `cep` varchar(8) NOT NULL DEFAULT '0',
  `logradouro` varchar(200) DEFAULT NULL,
  `bairro` varchar(200) DEFAULT NULL,
  `cidade` varchar(200) DEFAULT NULL,
  `estado` varchar(200) DEFAULT NULL,
  PRIMARY KEY (`cep`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

CEP				
cep	logradouro	bairro	cidade	estado

FIGURA 28 – ESQUEMA CEP.
FONTE: os Autores (2015).

4.3 TABELA TICKET

A tabela TICKET serve ao único propósito de armazenar os dados de log de acesso do usuário ao sistema.

```
CREATE TABLE IF NOT EXISTS `ticket` (
  `idEleicao` bigint(20) unsigned DEFAULT NULL,
  `cpf` varchar(11) NOT NULL DEFAULT '',
  `data` datetime DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`cpf`,`idEleicao`),
  KEY `fkTicketUsuarios` (`cpf`),
  KEY `fkTicketEleicao` (`idEleicao`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

TICKET		
idEleicao	cpf	dataHora

FIGURA 29 – ESQUEMA TICKET.
FONTE: os Autores (2015).

4.4 TABELA CANDIDATO

Esta tabela contém os dados dos candidatos de cada eleição. É importante destacar este vínculo entre candidato e eleição, pois uma mesma pessoa que se candidata em duas diferentes eleições, é instanciada duas vezes na tabela CANDIDATO.

Isto serve para que seja possível recuperar a lista de candidatos de qualquer eleição anterior assim como os resultados obtidos naquele momento.

```
CREATE TABLE IF NOT EXISTS `candidatos` (
  `idCandidato` `varchar`(5) NOT NULL DEFAULT '0',
  `idEleicao` `bigint`(20) unsigned NOT NULL,
  `tipo` `varchar`(30) DEFAULT NULL,
  `idPartido` `varchar`(2) DEFAULT NULL,
  `votos` `int`(11) NOT NULL DEFAULT '0',
  `nomeFantasia` `varchar`(200) DEFAULT NULL,
  `foto` `varchar`(200) DEFAULT NULL,
  PRIMARY KEY (`idCandidato`,`idEleicao`,`tipo`),
  KEY `fkCandidatosTipos` (`tipo`),
  KEY `fkCandidatosPartidos` (`idPartido`),
  KEY `fkCandidatosEleicoes` (`idEleicao`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

CANDIDATO						
idCandidato	idEleicao	tipo	idPartido	votos	nomeFantasia	foto

FIGURA 30 – ESQUEMA CANDIDATO.
FONTE: os Autores (2015).

4.5 TABELA ELEICAO

Nesta tabela guardam-se os dados das eleições geradas e gerenciadas pelo sistema.

```
CREATE TABLE IF NOT EXISTS `eleicoes` (
  `idEleicao` `bigint`(20) unsigned NOT NULL AUTO_INCREMENT,
  `horaInicio` `varchar`(5) DEFAULT NULL,
  `data` `varchar`(10) DEFAULT NULL,
  `horaFim` `varchar`(5) DEFAULT NULL,
  `tipo` `varchar`(100) DEFAULT NULL,
  PRIMARY KEY (`idEleicao`),
  UNIQUE KEY `data` (`data`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=0 ;
```

ELEICAO					
idEleicao	tipo	horaInicio	horaFim	data	tipo

FIGURA 31 – ESQUEMA DE ELEIÇÃO.
FONTE: os Autores (2015).

4.6 TABELA TIPO

Esta é a tabela que guarda os dados dos cargos a serem disputados em uma eleição.

```
CREATE TABLE IF NOT EXISTS `tipos` (
  `tipoEleicao` varchar(30) DEFAULT NULL,
  `tipo` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`tipo`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

TIPO	
tipo	tipoEleicao

FIGURA 32 – ESQUEMA TIPO.
FONTE: os Autores (2015).

4.7 TABELA PARTIDO

Esta tabela armazena os dados dos partidos eleitorais.

```
CREATE TABLE IF NOT EXISTS `partidos` (
  `idPartido` varchar(2) NOT NULL DEFAULT '0',
  `nome` varchar(200) DEFAULT NULL,
  `sigla` varchar(10) DEFAULT NULL,
  `logo` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`idPartido`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

PARTIDO			
idPartido	nome	sigla	logo

FIGURA 33 – ESQUEMA PARTIDO.
FONTE: os Autores (2015).

4.8 TABELA VOTO

Esta tabela, fugindo à regra de evitar-se redundância, foi criada apenas como forma alternativa de contagem e conferência de votos. Seu propósito é segurança de informação.

```
CREATE TABLE IF NOT EXISTS `votos` (
  `idEleicao` bigint(20) unsigned DEFAULT NULL,
  `idCandidato` varchar(5) DEFAULT NULL,
  `tipo` varchar(30) DEFAULT NULL,
  `idPartido` varchar(2) DEFAULT NULL,
  KEY `fkVotosCandidatos` (`idCandidato`),
  KEY `fkVotosEleicoes` (`idEleicao`),
  KEY `fkVotosTipos` (`tipo`),
  KEY `fkVotosPartidos` (`idPartido`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

VOTO			
idEleicao	tipo	idPartido	idCandidato

FIGURA 34 – ESQUEMA VOTO.
FONTE: os Autores (2015).

4.9 TABELA VAGA

Os números de vagas de cada tipo em cada eleição determinados pelo administrador do sistema serão armazenados nesta tabela.

```
CREATE TABLE IF NOT EXISTS `vagas` (
  `idEleicao` bigint(20) unsigned DEFAULT NULL,
  `PREFEITO` varchar(5) NOT NULL DEFAULT '0',
  `GOVERNADOR` varchar(5) NOT NULL DEFAULT '0',
  `PRESIDENTE` varchar(5) NOT NULL DEFAULT '0',
  `SENADOR` varchar(5) NOT NULL DEFAULT '0',
  `DEPUTADO ESTADUAL` varchar(5) NOT NULL DEFAULT '0',
  `DEPUTADO FEDERAL` varchar(5) NOT NULL DEFAULT '0',
  `VEREADOR` varchar(5) NOT NULL DEFAULT '0',
  PRIMARY KEY (`idEleicao`),
  KEY `fkVagasEleicoes` (`idEleicao`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

VAGA							
idEleicao	PRESIDENTE	GOVERNADOR	PREFEITO	DEPUTADO ESTADUAL	DEPUTADO FEDERAL	VEREADOR	SENADOR

FIGURA 35 – ESQUEMA VAGA.
FONTE: os Autores (2015).

4.10 TABELA ENDERECO

Esta tabela armazena os dados do endereço de cada usuário do sistema.

```
CREATE TABLE IF NOT EXISTS `enderecos` (
  `numero` varchar(11) NOT NULL DEFAULT '0',
  `complemento` varchar(200) NOT NULL DEFAULT '',
  `cep` varchar(8) NOT NULL DEFAULT '0',
  PRIMARY KEY (`numero`,`cep`,`complemento`),
  KEY `fkEnderecosCep` (`cep`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

ENDERECO		
numero	complemento	cep

FIGURA 36 – ESQUEMA ENDEREÇOS.
FONTE: os Autores (2015).

4.11 TABELA ERRO

A tabela de erros tem como propósito armazenar o código de erro e a descrição do mesmo no sistema. Como pode ser observado na FIGURA 54 no capítulo APÊNDICE II – TABELA DE ERROS DO SISTEMA;

```
CREATE TABLE IF NOT EXISTS `erros` (  
  `cod` int(11) NOT NULL DEFAULT '0',  
  `descricao` tinytext,  
  PRIMARY KEY (`cod`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

4.12 ALTERAÇÕES EM TABELAS

Por fim, realizamos ao final do *script* de inicialização do banco de dados do sistema alterações nas tabelas USUARIOS e CANDIDATOS, adicionando as chaves estrangeiras que visam manter a integridade referencial do banco, como mencionamos no início deste capítulo.

O primeiro comando apresentado no trecho de código abaixo estabelece relação entre a tabela USUARIOS e a tabela ENDERECOS.

Os comandos seguintes estabelecem relação entre a tabela CANDIDATO e as tabelas TIPOS, PARTIDOS e ELEICOES, respectivamente.

```
ALTER TABLE `candidatos`
  ADD CONSTRAINT `fkCandidatosEleicoes` FOREIGN KEY (`idEleicao`)
    REFERENCES `eleicoes` (`idEleicao`),
  ADD CONSTRAINT `fkCandidatosTipos` FOREIGN KEY (`tipo`)
    REFERENCES `tipos` (`tipo`),
  ADD CONSTRAINT `fkCandidatosPartidos` FOREIGN KEY (`idPartido`)
    REFERENCES `partidos` (`idPartido`);

ALTER TABLE `enderecos`
  ADD CONSTRAINT `fkEnderecosCep` FOREIGN KEY (`cep`)
    REFERENCES `cep` (`cep`);

ALTER TABLE `ticket`
  ADD CONSTRAINT `fkTicketUsuarios` FOREIGN KEY (`cpf`)
    REFERENCES `usuarios` (`cpf`),
  ADD CONSTRAINT `fkTicketEleicoes` FOREIGN KEY (`idEleicao`)
    REFERENCES `eleicoes` (`idEleicao`);

ALTER TABLE `usuarios`
  ADD CONSTRAINT `fkUsuariosEnderecos` FOREIGN KEY (`numero`, `cep`,
`complemento`)
    REFERENCES `enderecos` (`numero`, `cep`, `complemento`);

ALTER TABLE `vagas`
  ADD CONSTRAINT `fkVagasEleicoes` FOREIGN KEY (`idEleicao`)
    REFERENCES `eleicoes` (`idEleicao`);

ALTER TABLE `votos`
  ADD CONSTRAINT `fkVotosTipos` FOREIGN KEY (`tipo`)
    REFERENCES `tipos` (`tipo`),
  ADD CONSTRAINT `fkVotosEleicoes` FOREIGN KEY (`idEleicao`)
    REFERENCES `eleicoes` (`idEleicao`);
```

5 DIAGRAMA DE FLUXO DE DADOS

Com o modelo de dados representado e o banco de dados construído, faz-se necessário demonstrar através do Diagrama de Fluxo de Dados de que maneira estes dados trafegarão pelo sistema.

Iniciamos pelo diagrama de contexto para conhecer as entidades externas que interagem com o sistema, seguindo com DFDs de níveis 0, e 1 de acordo com a necessidade de aprofundamento em determinadas funcionalidades.

5.1 DIAGRAMA DE CONTEXTO

Neste diagrama (FIGURA 37), identificamos as seguintes entidades externas ao sistema: *web service*, eleitor e o administrador. O eleitor fornecerá os dados da votação e receberá um ticket de comprovação. Já o administrador, fornecerá os dados para que a eleição seja realizada, e receberá do sistema relatórios. O *web service* fornecerá os dados referente ao CEP para o sistema.

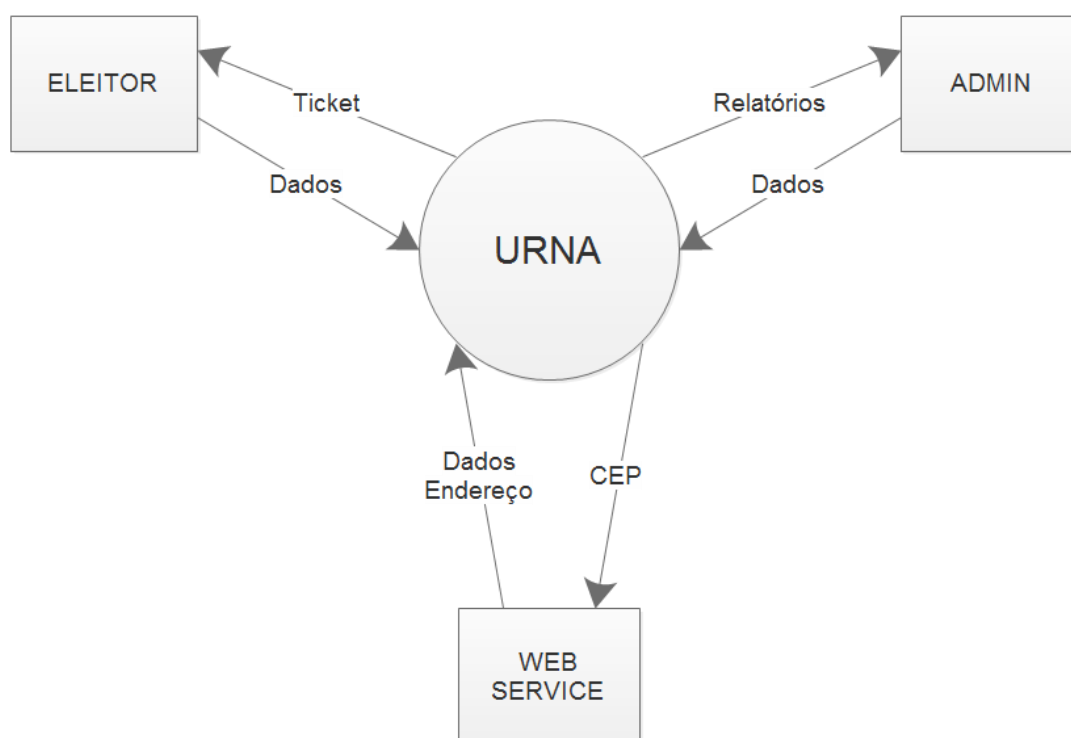


FIGURA 37 – DIAGRAMA DE CONTEXTO.
FONTE: os Autores (2015).

5.2 DFD - NÍVEL ZERO

Organizamos nosso sistema em dez grandes funções, representadas pela FIGURA 38. São elas: login, recuperar senha, computar voto, gerenciar ticket, gerenciar usuários, gerenciar partidos, gerenciar candidatos, gerenciar eleições, gerenciar erro e gerenciar relatório.

5.3 DFD - NÍVEL UM

Tendo o DFD acima representado o sistema em sua superficialidade, apresentamos agora o DFD de nível um com o objetivo de iniciar o aprofundamento no tema.

5.3.1 GERENCIAR CANDIDATOS

A função “gerenciar candidatos” foi dividida em “cadastrar candidato”, função responsável por inserir um novo candidato no banco de dados, “procurar candidatos”, função que lista na tela todos os candidatos que atendem ao padrão de busca escolhida pelo usuário do sistema e “deletar candidato”, função que exclui o registro selecionado do banco de dados. As funções “gerenciar eleições”, “gerenciar erros” e “gerenciar login” são funções do nível zero que foram reproduzidas neste nível.

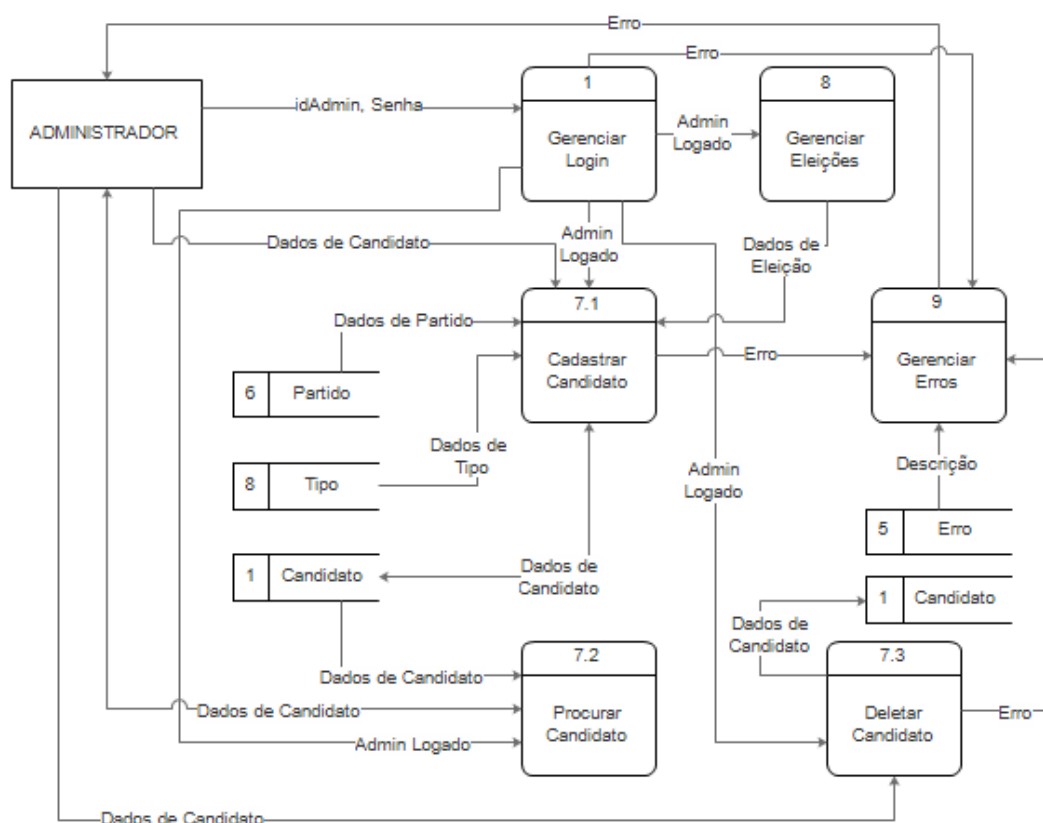


FIGURA 39 – GERENCIAR CANDIDATOS.
FONTE: os Autores (2015).

5.3.2 GERENCIAR PARTIDOS

A função “gerenciar partidos” foi dividida em “cadastrar partido”, função responsável por inserir um novo partido no banco de dados, “buscar partidos”, função que lista na tela todos os partidos que atendem ao padrão de busca escolhida pelo usuário do sistema, “deletar partido”, função que exclui o registro selecionado do banco de dados.

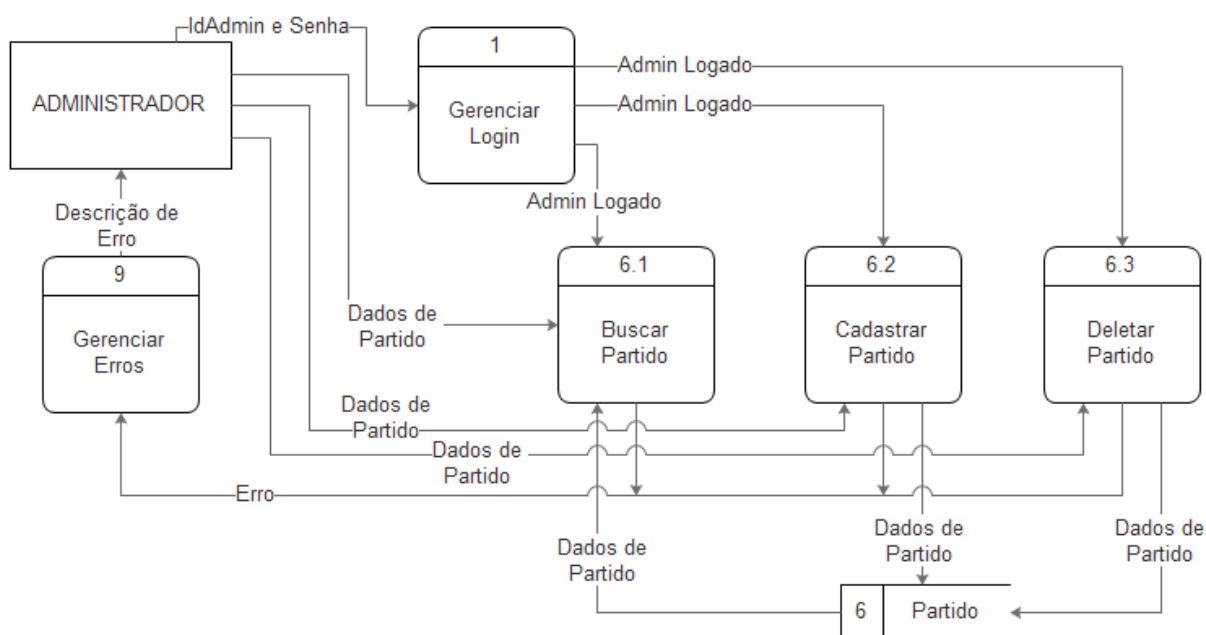


FIGURA 40 – GERENCIAR PARTIDOS.
FONTE: os Autores (2015).

5.3.4 GERENCIAR ELEIÇÃO

A função “gerenciar eleições” foi dividida em “cadastrar eleição”, função responsável por inserir uma nova eleição no banco de dados, “buscar eleições”, função que lista na tela todas as eleições que atendem ao padrão de busca escolhida pelo usuário do sistema, “excluir eleição”, função que exclui o registro selecionado do banco de dados e “alterar de eleição”, responsável por alterar dados que necessitem correção.

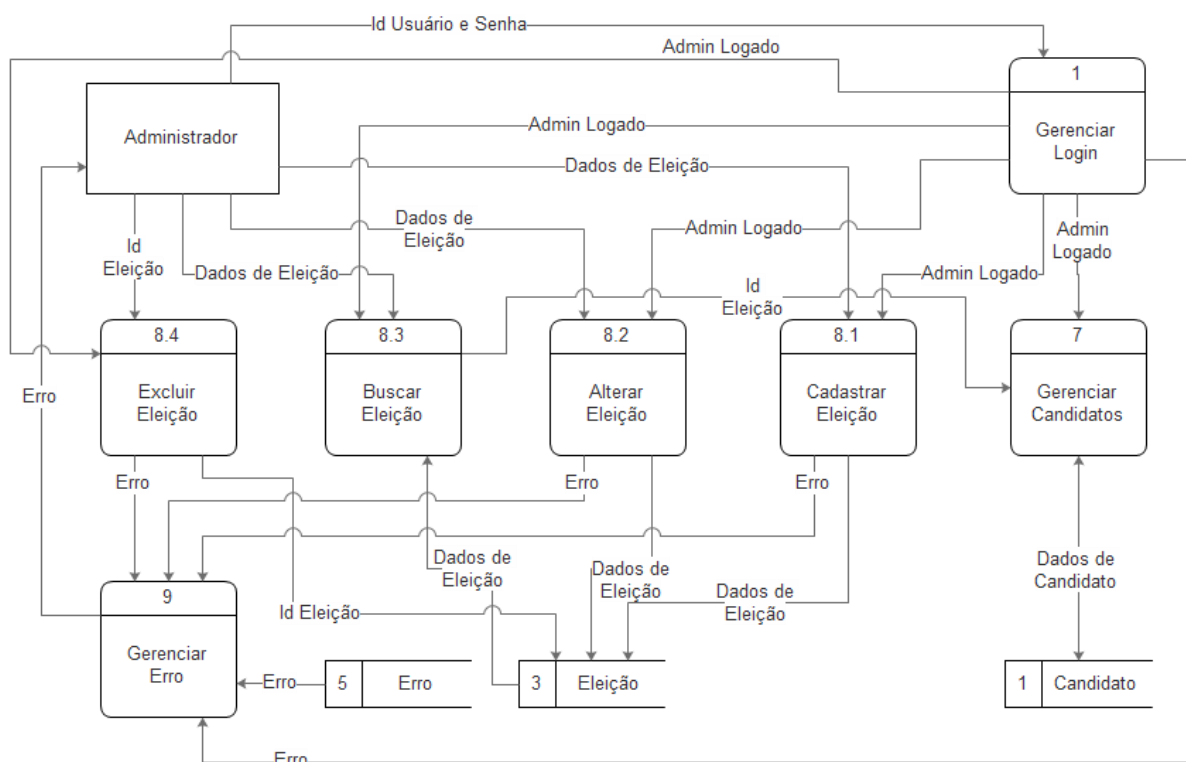


FIGURA 42 – GERENCIAR ELEIÇÕES.
FONTE: os Autores (2015).

5.3.5 COMPUTAR VOTO

A função “computar voto” foi dividida em “computar voto”, função responsável por inserir um voto no banco de dados, “buscar partido”, função que busca um partido no banco de dados e “busca candidato”, função que busca um candidato no banco de dados.

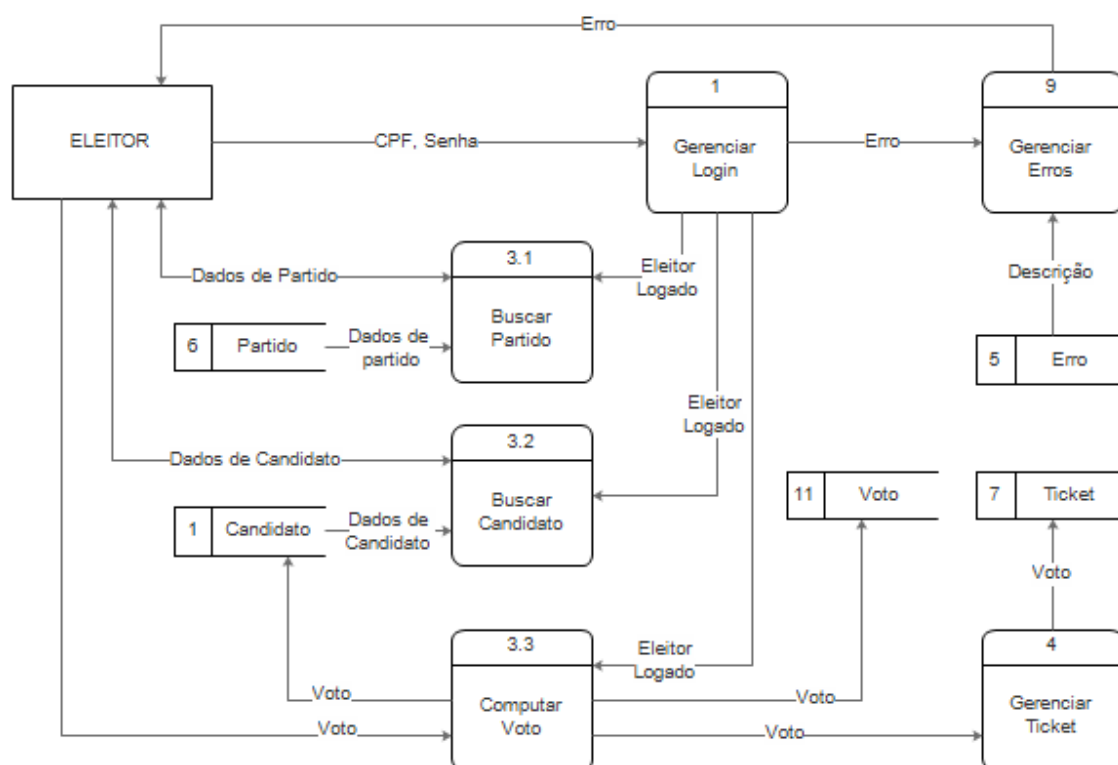


FIGURA 43 – COMPUTAR VOTO.
FONTE: os Autores (2015).

5.3.6 RECUPERAR SENHA

Esta função foi quebrada em outras três: gerar código, gerenciar código e verificar erro. Caso o usuário que está tentando recuperar a senha exista no banco de dados é gerado um código e este é enviado para o endereço eletrônico do usuário que deverá inseri-lo no sistema. Caso contrário, uma mensagem de erro será emitida.

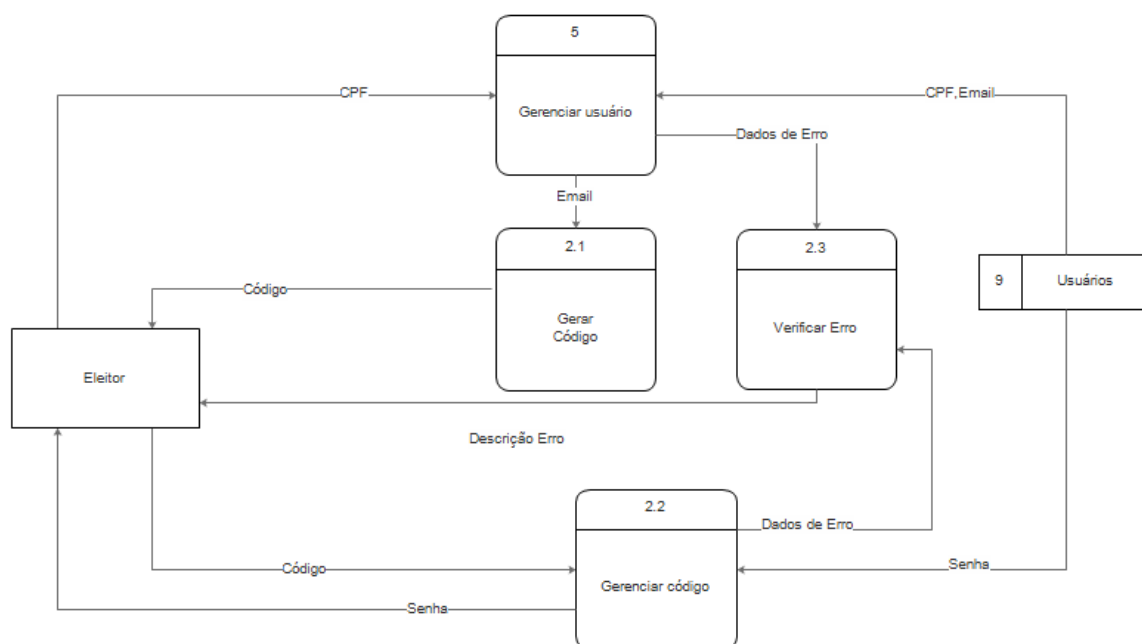


FIGURA 44 – RECUPERAR SENHA.
FONTE: os Autores (2015).

5.3.7 VERIFICAR LOGIN

A verificação de *login* possui as funções “validar usuário”, responsável por validar os dados de CPF e senha, “verificar usuário”, que verifica se os dados correspondem a um usuário existente no banco de dados, “verificar eleição”, que verifica se alguma eleição está com o *status* de “em andamento”, “verificar ticket”, responsável por verificar se o usuário em questão já votou na atual eleição e “controlador *login*”, que realiza o gerenciamento e a sequência de execução das funções de “verificar login”.

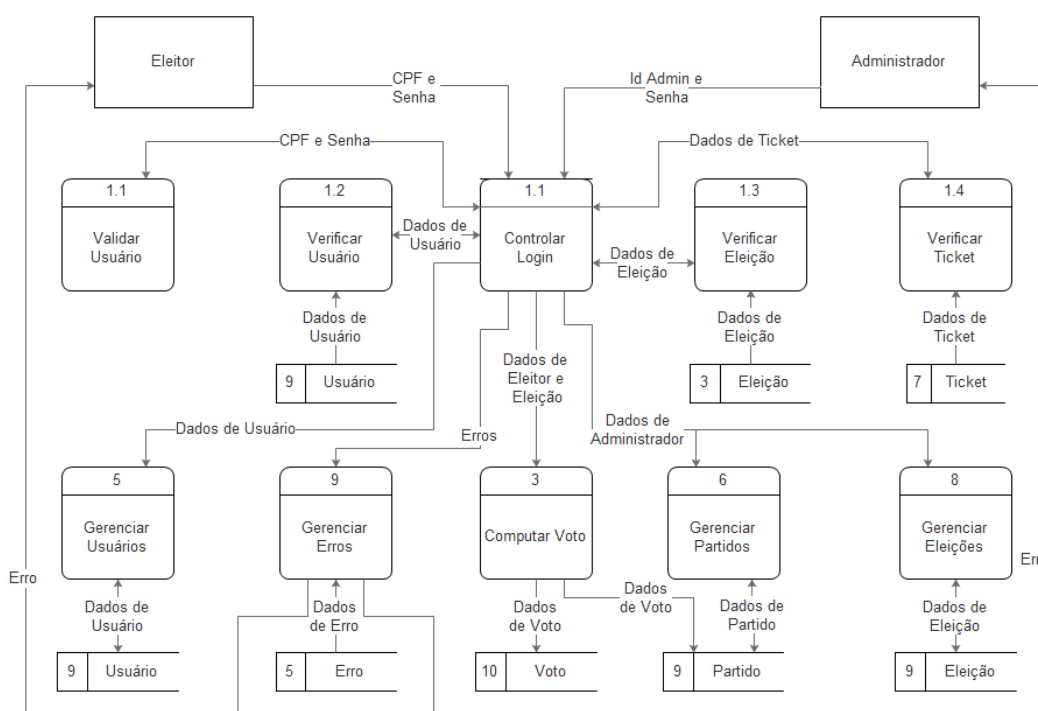


FIGURA 45 – VERIFICAR LOGIN.
FONTE: os Autores (2015).

Com os diagramas apresentados é possível ter ampla noção sobre as funcionalidades e tráfego de dados do sistema, facilitando assim a compreensão do mesmo e de sua implementação.

6 FONTES

O desenvolvimento de tal projeto não seria possível sem a utilização de diversas ferramentas que possibilitam e facilitam a programação em alto nível e disto trata-se este capítulo. Nosso objetivo aqui é explicitar os motivos pelos quais optamos por cada uma destas ferramentas e como elas nos auxiliaram a alcançar nossos objetivos.

Seguindo a cronologia do desenvolvimento deste trabalho, começamos pelos softwares da Microsoft, Word e Excel. O primeiro, utilizamos para a confecção deste texto, enquanto o segundo utilizamos em diversos momentos para nos auxiliar na organização do tempo, através de um cronograma, e do trabalho, com tabelas com divisão de tarefas, assim como na confecção de tabelas para ilustrar o trabalho. A escolha por estes programas se deu pela grande familiaridade que todos os integrantes do grupo possuem com o pacote *Office*.

Para a prototipação de telas vista no capítulo 2 utilizamos o “Balsamiq Mockups”. Esta poderosa ferramenta de simples utilização possibilitou a rápida visualização das telas, permitindo assim correções e alinhamentos que seriam inviáveis se discutidos apenas em abstrações.

Outro software essencial para este projeto, o BrModelo, foi utilizado pelos autores deste trabalho para modelagem de dados, tanto do Modelo Entidade Relacionamento quanto do Modelo Relacional. Este software nos foi apresentado pelo Professor Doutor João Eugênio Marynowski, responsável pela disciplina homônima, como uma boa alternativa para a construção dos modelos.

Gerado num primeiro momento a partir também do BrModelo, o script de dados foi extensivamente testado e otimizado a partir de sua implementação através do sistema gerenciador de banco de dados MySQL, o qual escolhemos por fazer parte do ambiente de desenvolvimento web WAMP, acrônimo de Windows, Apache, MySQL e PHP, que nos forneceu de maneira simples, toda a base para o desenvolvimento deste projeto. Este ambiente é bastante intuitivo e fácil de utilizar, ideal para graduandos.

Para a criação do Diagrama de Fluxo de Dados, utilizamos o software Edraw Max 7.9. Entre suas bibliotecas, há uma que possui todos os elementos

necessários para a representação gráfica desta etapa. Seus comandos são fáceis de aprender e permitem que se organize o DFD de maneira bastante eficiente.

Partindo então para a implementação do projeto, nossa equipe foi dividida em duas, os desenvolvedores frontend e os desenvolvedores backend. Os primeiros optaram pela IDE Code, da Microsoft. Uma das grandes vantagens desta opção é a integração com a plataforma de controle de versões GitHub, utilizada de maneira contínua pelas equipes de desenvolvimento.

A interface visual foi desenvolvida em HTML 5 e CSS 3, utilizando-se largamente o framework Bootstrap. A interatividade do sistema foi possível através da utilização da popular biblioteca Java Script, JQuery. Com o auxílio desta biblioteca foi feita toda a validação dos campos de inserção de dados e através de Ajax, a comunicação assíncrona com o backend. Os desenvolvedores frontend também utilizaram a biblioteca “Masked Input Plugin” para a criação de máscaras para impedir que os campos de inserção de dados fossem preenchidos de maneira incorreta pelos usuários do sistema. Já para a inclusão de arquivos de áudio no sistema, foi utilizada a biblioteca “Ion.Sound” em sua versão 2.0.1, por sua rica documentação e fácil compreensão.

Destacamos aqui a utilização do Webservice “República Virtual” onde buscamos informações de endereço a partir da inserção de um número de CEP no campo correspondente, trazendo conveniência a nossos usuários e confiabilidade nas informações fornecidas.

Por sua vez, os desenvolvedores backend optaram pela IDE Sublime Text 3 pela versatilidade da plataforma, assim como pela usabilidade e infinidade de atalhos que facilitam a programação encontradas neste software para codificar o sistema na linguagem de programação PHP, que se encontra em sua quinta versão.

Para o envio do código para recuperação da senha do usuário através de correio eletrônico utilizamos a biblioteca PHPMailer, atualmente na versão 5.2.4. Em pesquisa realizada de maneira rápida e informal na internet, principalmente em fóruns de desenvolvedores como o “Stack Overflow”, constatamos que é uma das bibliotecas de envio de correio eletrônico autenticado mais utilizadas e atendeu às nossas necessidades.

Já para a geração de relatórios em PDF, escolhemos a biblioteca FPDF pela familiaridade com a mesma. Sua utilização é simples e há documentação em português.

Além do GitHub, mencionado anteriormente como ferramenta escolhida para controle de versões, o Google Drive, ferramenta de armazenamento na nuvem, foi largamente utilizado por sua capacidade de compartilhamento de arquivos e edição simultânea de documentos.

Para o desenvolvimento deste projeto foram utilizados computadores com as seguintes configurações:

1. Dell Vostro 3550, Windows 7 Professional – Service Pack 1 – 64 bits, processador Intel i5-2450M 2.5 GHz, 8 GB de memória RAM e 685 GB de disco rígido.
2. Asus K45A, Windows 7 Home Premium – Service Pack 1 – 64 Bits, processador Intel Core i5-3210M 2.5 GHz, 6 GB de memória RAM e 1 TB de disco rígido.
3. Acer aspire M, Windows 8.1 Single Language - 64 Bits, processador Intel Core i3-3227U 1.9 GHz, 4 GB de memória RAM, 500 GB de disco rígido, e 20 GB de SSD.
4. Acer aspire E1-572-6638, Windows 8.1 Single Language – 64 Bits, processador Intel Core i5-4200U 1.6GHz, 4 GB de memória RAM e 500 GB de disco rígido.

Não é difícil chegar à conclusão de que o vasto conhecimento de plataformas, frameworks e ferramentas de desenvolvimento é essencial a todos os desenvolvedores, pois sua utilização diminui o tempo necessário para a conclusão dos projetos e possibilita o acesso a uma ampla gama de possibilidades que demandariam extremo esforço para se obter por conta própria.

7 CONCLUSÃO

Todas as etapas deste projeto foram concluídas dentro do prazo e de maneira satisfatória, assim como o produto final atende a todos os requisitos de sistema propostos. Não obstante, ainda há espaço para melhorias no sistema e é possível no futuro adaptá-lo a diferentes necessidades, como por exemplo, eleições acadêmicas ou estudantis, facilitando o processo eleitoral e aumentando a participação. Ideias estas que passam a integrar o rol de possibilidades para o trabalho de conclusão de curso desta equipe.

Entretanto, não só o objetivo de desenvolver uma urna eletrônica foi alcançado como também os foram os objetivos de aprendizado. Através de muita pesquisa, tentativas, correções de erros, reestruturações na organização do projeto e consultas a inúmeros professores, acreditamos ter ido além do conhecimento exigido como base para esta disciplina e este projeto.

Seria leviano afirmar que nossa principal conquista tenha sido o sistema funcional e esteticamente bem elaborado que desenvolvemos. Acreditamos sim, que o conhecimento adquirido na utilização do padrão de arquitetura MVC (Modelo-Visão-Controlador), na utilização de inúmeras bibliotecas, na integração do sistema com diferentes tecnologias como envio de correio eletrônico, geração de arquivos no formato pdf ou a comunicação com um Webservice é o que nos trará benefícios para a continuidade do curso e posteriormente no mercado de trabalho.

A experiência de gerenciamento de projetos e de definição de prazos foi igualmente enriquecedora, mostrando-nos que muitas vezes temos que nos adaptar a imprevistos e buscar soluções alternativas àquelas que em primeiro momento parecem mais acertadas.

Colocamos em prática conceitos aprendidos em distintas disciplinas e assim integramos e consolidamos o conhecimento adquirido até o momento no curso, desde as vantagens da modularização de um algoritmo, da utilização de uma metodologia estruturada para modelagem de sistemas, como a implementação de um banco de dados baseado em um modelo previamente construído.

Tendo atendido a todos os pontos propostos no início do processo de desenvolvimento deste projeto, concluimos que este trabalho obteve sucesso ao alcançar de maneira bastante satisfatória seus objetivos, superando as expectativas de seus autores.

8 APÊNDICE I – PROTÓTIPOS DE TELAS NÃO UTILIZADOS

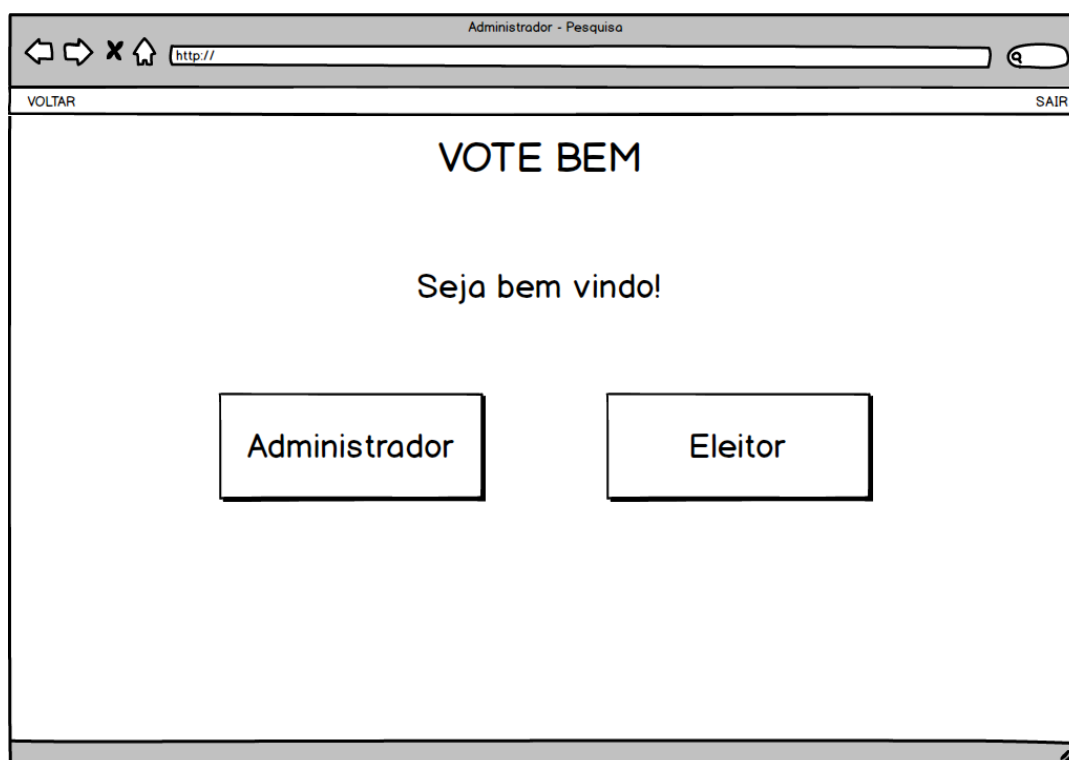


FIGURA 46 – SELEÇÃO DE USUÁRIO.
FONTE: os Autores (2015).

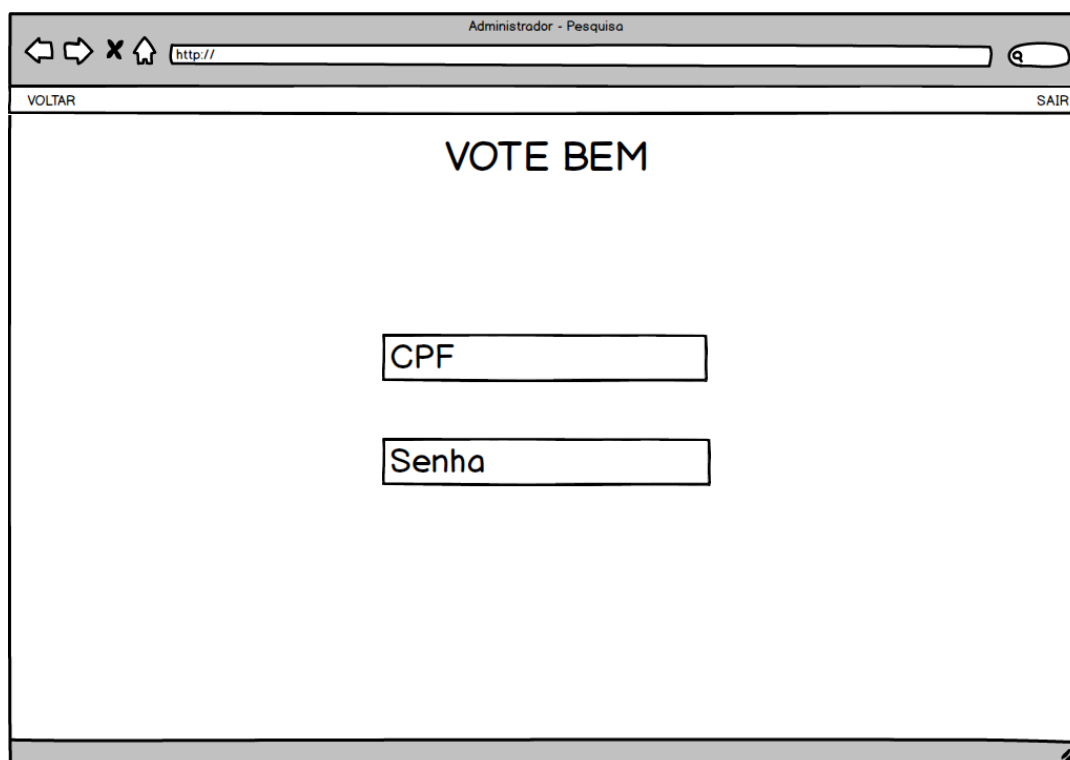


FIGURA 47 – TELA DE LOGIN.
FONTE: os Autores (2015).

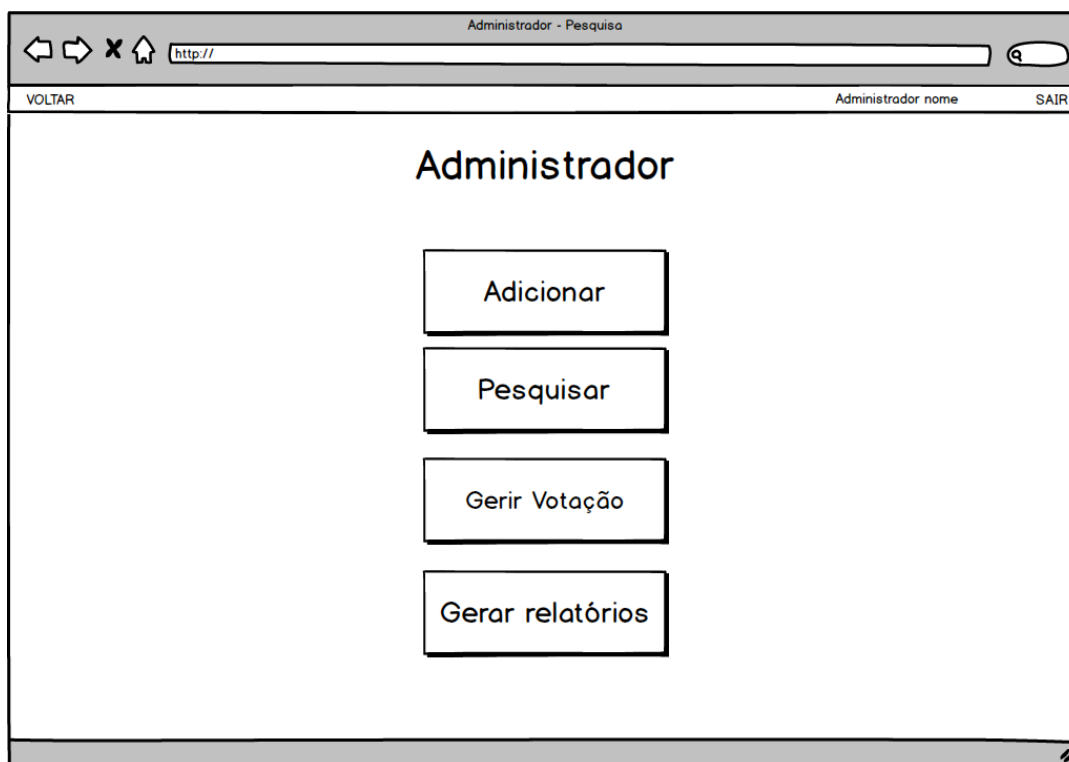


FIGURA 48 – MENU INICIAL DO ADMINISTRADOR.
FONTE: os Autores (2015).

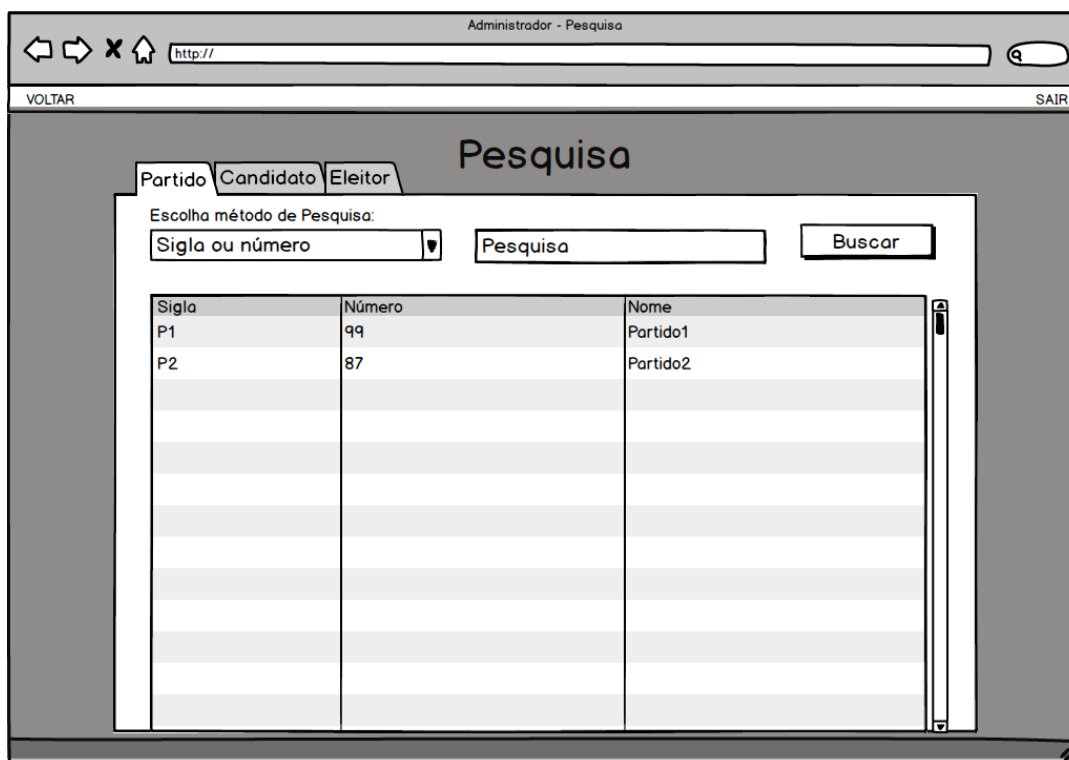


FIGURA 49 – PESQUISA DE PARTIDOS.
FONTE: os Autores (2015).

Administrador - Pesquisa

VOLTAR SAIR

Pesquisa

Partido Candidato Eleitor

Escolha método de Pesquisa:

Nome, Número Pesquisa Buscar

Cargo	Número	Partido	Nome
Prefeito	99	Partido1	Nome_nome
Deputado	99123	Partido1	Nome_nome
Vereador	97122	Partido2 Nome_nome	
Vereador	99545	Partido1	Nome_nome

FIGURA 50 – PESQUISA DE CANDIDATO.
FONTE: os Autores (2015).

Administrador - Pesquisa

VOLTAR SAIR

Pesquisa

Partido Candidato Eleitor

Escolha método de Pesquisa:

Nome, Título, CPF Pesquisa Buscar

Nome	Título	CPF	Zona	Sessão	
Alisson f. Krul	XXXX	XXXXX	xXXX	XXXXX	
Alisson f. Krul	XXXX	XXXXX	xXXX	XXXXX	

FIGURA 51 – PESQUISA DE ELEITORES.
FONTE: os Autores (2015).

Administrador - Pesquisa

http://

VOLTAR Administrador nome SAIR

NOVA VOTAÇÃO

Deputados: Vagas

Vereadores: Vagas

Prefeito: ☐ 1 Vaga

Hora de Inicio:

Hora de Encerramento:

FIGURA 52 – AGENDAMENTO DA VOTAÇÃO.
FONTE: os Autores (2015).

Administrador - Pesquisa

http://

VOLTAR Administrador nome SAIR

Relatórios

FIGURA 53 – EMISSÃO DE RELATÓRIOS.
FONTE: os Autores (2015).

9 APÊNDICE II – TABELA DE ERROS DO SISTEMA

Número	Descrição
-1	CPF inválido
-2	Usuário não existe
-3	Falha na busca
-4	Senha incorreta
-5	Título Incorreto
-6	Nome inválido.
-7	Senhas não conferem
-8	Data invalida
-9	Usuários com menos de 16 não podem votar
-10	Seção invalida
-11	CEP inválido
-12	Campo de número de Endereço inválido.
-13	Usuario já cadastrado
-14	Campo(s) em branco
-15	Sessão invalida
-16	Partido já cadastrado
-17	Número de partido inválido
-18	Nome de partido
-19	Hora inválida
-20	Hora de término inválida
-21	Número de vereadores inválido
-22	Número de deputados estaduais inválido
-23	Número de deputados federais inválido
-24	Número de candidato inválido
-25	Nome de Candidato inválido
-26	Eleição não existe
-27	Eleição já cadastrada no mesmo dia
-28	Candidato com o mesmo número já cadastrado
-29	Cargo inválido
-30	Sigla de partido inválida

FIGURA 54 - TABELA DE ERRO.

Fonte: Os autores (2015);