

UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

HABIBU ABDULLAHI

MARCELO MAZURKY

RODRIGO TEMISKI MUNIZ

Viralata: Sistema de Gerenciamento de Adoção de Animais

CURITIBA
2014

UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

HABIBU ABDULLAHI

MARCELO MAZURKY

RODRIGO TEMISKI MUNIZ

Viralata: Sistema de Gerenciamento de Adoção de Animais

Trabalho de Conclusão de Curso apresentado como requisito parcial para conclusão do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná.

Orientador: Prof. Msc. Razer A. N. Rojas Montaño

CURITIBA
2014

RESUMO

Este trabalho apresenta o desenvolvimento de um sistema de gerenciamento de adoção de animais, capaz de atender a demanda de um adotante. Este sistema, baseado na plataforma *Java Web*, atua auxiliando nas tarefas de gerenciamento de adotante, animais, adoção de animais e geração de relatórios de vários tipos, tais como relatório geoanalítico, relatório de abandono de animais e relatório de adoções. O nome de sistema, denominado *Viralata* por sua equipe desenvolvedora, utiliza um servidor remoto para armazenamento de seus dados, e efetua as operações de criação, alteração, consulta e exclusão de dados.

Palavras Chave: *Java Web*, comunicação, servidor, adotante.

ABSTRACT

This paper presents the development of a system created for adopting animals, able to meet the demand of an adopter. The system is based on the Java web platforms works to help and simplified the pet adaptation process the person whom like to adopt or donate to another person to adapt, pet management and generate various types of reports such as geoanalysis report, graphical report for pet abundance and statistical report of pet adopters. The system called Viralata by its developer team, uses a remote server to store the data, and performs the operations of creation of data, modification of data, query the data and deletion of the data.

Key words: Java Web, communication, server, pet adaption.

AGRADECIMENTOS

Aos nossos familiares e amigos, que sempre nos apoiaram nos momentos difíceis e sempre nos motivaram quando necessário. Aos professores do curso, pelos conhecimentos a nós repassados ao decorrer dos últimos anos, que muito auxiliaram no desenvolvimento deste trabalho.

Aos grandes exemplos de sucesso da área de Tecnologia de Informação, que sempre nos incentivaram a dar o máximo e a nunca desistir de nosso sonhos. Em especial, à equipe envolvida, que se dedicou completamente ao desenvolvimento deste projeto, superando todos os desafios que encontraram durante o desenvolvimento.

LISTA DE SIGLAS

EE – *Enterprise Edition*
GNU – *GNU is Not Unix*
GPL – *GNU General Public License*
IDE – *Integrated Development Environment*
RAM – *Random Access Memory*
ROM – *Read-Only Memory*
SQL – *Structured Query Language*
STS – *Spring Tool Suite*
OMG – *Object Management Group*

SUMÁRIO

1. INTRODUÇÃO	1
1.1 DESCRIÇÕES DO ADOTANTE	2
1.2 OBJETIVOS DO PROJETO	2
1.3 ORGANIZAÇÕES DO TRABALHO	2
2. FUNDAMENTAÇÃO TEÓRICA	4
2.1 ADOÇÃO	4
2.2 HISTÓRICOS DAS LEIS DE ADOÇÃO NO BRASIL	5
2.3 ADOÇÕES DE ANIMAIS	7
3. METODOLOGIA	10
3.1 MODELAGENS DE SISTEMA	10
3.2 METODOLOGIAS DE DESENVOLVIMENTO DO PROJETO	10
3.2.1 FASES DE DESENVOLVIMENTO DO PROJETO	11
3.2.2 VANTAGENS E DESVANTAGENS DO MODELO	13
3.3 PLANOS DE ATIVIDADES	14
3.3.1 DIAGRAMA <i>WBS</i>	14
3.3.2 DIAGRAMA DE <i>GANTT</i>	15
3.4 PLANO DE RISCOS	15
3.5 MATERIAIS	16
3.5.1 MATERIAIS DE AMBIENTE E DESENVOLVIMENTO	17
3.5.1.1 MÁQUINAS E MATERIAIS	17
3.5.1.2 LINGUAGEM DE PROGRAMAÇÃO	18
3.5.1.3 SISTEMA GERENCIADOR DE BANCO DE DADOS	18
3.5.1.4 SERVIDOR DE APLICAÇÃO	19
3.5.1.5 AUTOMAÇÃO DE COMPILAÇÃO	19

3.5.1.6 SISTEMA DE VERSIONAMENTO	20
3.5.1.7 AMBIENTE INTEGRADO DE DESENVOLVIMENTO	20
3.5.2 MATERIAIS DE MODELAGEM DO PROJETO.....	21
3.5.2.1 FERRAMENTA PARA CRIAÇÃO DE DIAGRAMAS <i>WBS</i>	22
3.5.2.2 FERRAMENTA PARA CRIAÇÃO DE DIAGRAMAS DE <i>GANTT</i>	22
3.6 DESENVOLVIMENTOS DO PROJETO	22
4. ANÁLISE DE REQUISITOS.....	26
4.1 REQUISITOS PRINCIPAIS	26
4.1.2 ITENS A SEREM CONSIDERADOS	27
4.2.1 SUGESTÕES.....	28
4.2 PROPOSTAS DE SOLUÇÃO.....	29
4.2.1 GERAÇÃO DE RELATÓRIOS GERENCIAIS	29
4.2.2 MAPEAMENTO DAS ÁREAS VISITADAS ATRAVÉS DE GEORREFERENCIAMENTO	30
4.2.3 CRIAÇÃO DE UM PRONTUÁRIO DE ATENDIMENTO PARA ANIMAIS.....	31
4.2.4 PLANO DE AÇÃO E PROJETO DE RESGATE DOS ANIMAIS	31
5. APRESENTAÇÃO DE SISTEMA	32
5.1 INSTALAÇÃO E CONFIGURAÇÃO DO AMBIENTE DE DESENVOLVIMENTO	32
5.2 TELAS DO SISTEMA	35

6. CONSIDERAÇÕES FINAIS	59
6.1 CONCLUSÃO.....	59
REFERÊNCIAS BIBLIOGRÁFICAS	61
APÊNDICE – DOCUMENTAÇÃO DO SOFTWARE	63
DOCUMENTAÇÃO DO PROJETO	63
Diagrama WBS.....	63
Tabela de Gantt.....	65
DIAGRAMAS UML	68
Diagrama de caso de uso.....	68
Especificação de caso de uso	69
Diagrama de Classes	94
Diagrama de Sequencias	96
Diagrama de Estado.....	104
Diagrama de Entidade Relacionamento	106
Tabela de dicionário de dados	108
Protótipos de telas.....	128

1. INTRODUÇÃO

Com o número crescente de animais vivendo nas ruas nos últimos anos, surgiu a necessidade de elaborar um sistema a fim de gerar soluções no processo de adoção de animais. Para tanto, é necessário que o sistema gerencie desde os processos de abrigo de animais até o recolhimento na rua, incluindo também o processo de adoção. Este setor não possui uma gama muito extensa de programas mais específicos para gerenciamento de adoção de animais, porque, muitas vezes existe uma necessidade de marcações manuais para o controle de adoções. Levando este fator em conta, considera-se de grande importância a existência de um sistema que facilite estes processos, tornando o trabalho da adoção de animais fácil e ágil.

Para que o sistema possa atuar de uma maneira eficiente, primeiramente os dados dos animais são organizados e classificados, a fim de se ter um melhor controle dos mesmos, e também para que outros setores relacionados à medicina veterinária possam ter acesso a esses dados, tornando possível uma movimentação dos animais para outros locais (transferência de um abrigo para outro, por exemplo).

Existe também a necessidade de se ter uma correta distribuição do espaço disponível para abrigar os animais. Por conta do grande volume de recolhimentos, muitas vezes os animais que são recolhidos acabam dependendo de abrigos de terceiros, caso o contrário, são devolvidos às ruas.

Em seguida, é necessário que seja definida uma estratégia para alavancar a adoção de animais considerados de baixa procura, tais como vira-

latas, animais com deformidades ou de maior porte, etc. Estes são os que compõem a maioria dos animais retidos no abrigo da prefeitura.

1.1 DESCRIÇÕES DO ADOTANTE

O adotante deste projeto é uma pessoa física ou jurídica, tais como organizações não governamentais (ONG), que apresentam interesse em adotar um animal.

1.2 OBJETIVOS DO PROJETO

O objetivo geral do projeto é desenvolver uma aplicação de controle de adoções e recolhimento de animais, tornando o processo de gerenciamento de adoção mais ágil e eficiente.

Para que estes objetivos gerais sejam atendidos, é necessário que os seguintes objetivos específicos estejam em conformidade:

- Servidor de aplicação e banco de dados configurados e em execução;
- Computador ou dispositivo móvel conectado à rede.

1.3 ORGANIZAÇÕES DO TRABALHO

No capítulo 2 será abordada a questão do histórico das leis de adoções, tipos, finalidades e demais aspectos, a fim de dar um melhor entendimento ao trabalho. O capítulo 3 descreve as metodologias utilizadas no desenvolvimento do projeto, ferramentas utilizadas durante sua confecção, divisão das atividades entre a equipe e, por fim, uma descrição sobre o

andamento do desenvolvimento do trabalho. No capítulo 4 encontra-se uma análise de requisitos para desenvolvimento de sistema, o próximo capítulo 5 será apresentação geral do sistema, com protótipos das telas do sistema, ilustrando suas funcionalidades, e como deve ser encaminhada sua instalação. Por fim, o capítulo 6 apresenta as considerações finais dos resultados obtidos com o projeto, e possíveis aplicações.

2. FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo apresentar os principais conceitos relacionados à adoção de animais, assim como o processo que envolve o gerenciamento dessas adoções no meio social.

2.1 ADOÇÃO

Adoção (do latim *ad optare* - optar, escolher), dentro do atual âmbito civil brasileiro, é o ato civil através do qual um indivíduo ou casal assume uma pessoa com idade de até 18 anos à data da adoção referente ao Código Civil Brasileiro. (CCB de 1916, Art. 375).

Esta prática é comum entre casais que não podem ter filhos, seja por infertilidade, problemas de saúde que afetem a gravidez (como má formação do útero na mulher, por exemplo), casais homossexuais que optam por não utilizar uma “barriga de aluguel”, ou mesmo pessoas que, por vontade própria de ajudar crianças órfãs, assumem as responsabilidades legais sobre o adotado.

Segundo o procurador de justiça Domingos Calderaro (1964), “a adoção é, mesmo, preferível à delegação do pátrio poder, guarda ou tutela, em face do vínculo mais profundo que firma entre o adotante e o adotado, propiciando a este maior integração sócio familiar”.

2.2 HISTÓRICOS DAS LEIS DE ADOÇÃO NO BRASIL

A primeira lei que regia a prática da adoção no Brasil data do ano de 1916. Àquela época, a adoção era absurdamente difícil de ser efetuada, pois o Art. 375 do Código Civil (COELHO, BRUNA FERNANDES, 2011) vigente determinava que:

- Só podiam adotar aqueles com idade mínima de cinquenta anos, sem descendentes legítimos ou legitimados e deveria ser, ao menos, dezoito anos mais velho que o adotado;
- A adoção conjunta só era possível se ambos fossem casados;
- Era exigido o consentimento da pessoa que tivesse a guarda do adotado;
- Eram causas para a dissolução da adoção as convenções entre as partes ou a ingratidão do adotado contra o adotante;
- Exceto quanto aos impedimentos para convolar núpcias, o parentesco se dava apenas entre o adotante e o adotado;
- Os efeitos gerados pela adoção não seriam extintos pelo nascimento posterior de filhos legítimos, exceto se a concepção tivesse precedido o momento da adoção;
- Com o nascimento de filhos legítimos, a herança do adotado seria reduzida à metade do que coubesse a cada um dos filhos;

- Os direitos e deveres resultantes do parentesco natural permaneceriam, exceto o poder familiar, que se transferia ao pai adotivo.

Até o advento do código de menores, as adoções eram regidas pelo Código Civil.

Em 1953, o senador Mozart Lago apresentou um Projeto de Lei que modificava as normas para adoção, o que viria a se tornar a Lei nº 3.133/57, em 1957. De acordo com as novas regras, a idade mínima do adotante foi reduzida para trinta anos, a diferença mínima de idade entre o adotante e o adotado passou de 18 para 16 anos, e pessoas com filhos poderiam também adotar, o que conferia ao ato da adoção um caráter mais assistencial. Porém, o adotado não possuía direito sucessório reconhecido caso o adotante já tivesse filhos legítimos, legitimados ou mesmo reconhecidos, e o acordo de adoção poderia ser rompido a qualquer momento.

A Lei nº 4.655, de 1965, passou a prover a legitimação adotiva, a qual se aplicava aos menores em situação irregular, seja por conduta própria do adotado (delitos), familiar (maus tratos) ou social (abandono). Isso lhes garantia os mesmos direitos cabíveis aos filhos legítimos do adotante, quando houvessem. Porém, a prática da adoção ainda era muito dificultada devido à burocracia envolvida.

Por meio do Código de Menores, instituído através da Lei nº 6.697 em 1979, passamos a existir dois tipos de adoção no meio jurídico nacionais: adoção simples e adoção plena.

O primeiro tipo, também conhecido como restrito, era regulamentado pelo Código Civil e abrangia os processos de adoção daqueles com idade entre 18 e 21 anos.

Após a introdução da Lei 6.697/79, a adoção de menores deixou de ser interesse jurídico principal do adotante e alterou-se o instrumento para lhe dar validade, passando então a depender da participação ativa do Estado por meio de autorização judicial, preenchendo as formalidades necessárias para consumação do ato adotivo; com esta alteração, passou a ser protegido o bem-estar e a pessoa do adotado menor.

O segundo tipo, também conhecido como pleno, substituiu a legitimação adotiva, prevista na lei nº 4.655/65 - era apenas efetuada com o consentimento dos pais ou representante legal do adotando e precedida estágio de convivência com a criança ou adolescente por prazo fixado pelo juiz; este tipo de adoção era irrevogável a partir do trânsito em julgado da sentença constitutiva, esta precedida de instrução processual e também de uma instrução psicossocial.

A adoção simples e a plena deixaram de existir (pelo Código Civil atual e pela Lei 8.069/90), visto que se aplicará a todos os casos da adoção. A adoção passa a ser irrestrita, trazendo importantes reflexos nos direitos da personalidade e nos direitos sucessórios.

2.3 ADOÇÕES DE ANIMAIS

No Brasil, não existe uma lei específica que rege os processos de adoção para animais, tanto doméstica quanto silvestres. No entanto, existem regulamentações a respeito do comércio desses animais, a fim de se ter um controle sobre as operações de compra e venda, bem como restringir o comércio de animais silvestres, em risco de extinção ou mesmo domésticos.

Em Curitiba, foi sancionada a Lei nº 13.914 que garante a proibição da criação comercial de animais em todo o município de Curitiba, considerando que o morador não resida em área rural. (Luciano Ducci, 2011).

Obviamente, há diferenças entre adoções de crianças adolescentes e adoções de animais.

A adoção de animais procede-se da seguinte forma:

- Entrevista: No primeiro contato o tutor é submetido a uma entrevista; a ONG ou protetor buscará saber se o tutor já tem ou teve animais, se a residência possui espaço e segurança para o animal, se há crianças ou mais pessoas na casa. Estas perguntas são realizadas para saber se o animal se adequará a sua nova família;
- Entrega: É comum o acompanhamento do voluntário no momento da entrega do animal ao seu novo lar; este acompanhamento é realizado para que se verifique a reação do animal em seu novo lar, assim como sua adaptação à casa e à família;
- Contrato (ou formulário de adoção): No momento da adoção é comum a entrega para o tutor de um termo de responsabilidade sobre o animal. Este termo especifica as necessidades do animal e o novo tutor se compromete a cuidar do animal adotado durante toda sua vida; caso o novo tutor não possa sustentar o animal em algum momento, ou na ocasião de algum imprevisto, o tutor se compromete também a informar o doador, para que este encontre um novo lar para o animal;
- Acompanhamento: Mesmo após a adoção, o contato com o doador pode continuar, pois é comum o mesmo pedir por fotos e notícias sobre seu

animalzinho, sua adaptação e verificar se os tutores estão felizes com a presença do bichinho em casa.

3. METODOLOGIA

Serão apresentadas neste capítulo as metodologias utilizadas para organização do projeto, assim como os materiais utilizados no desenvolvimento do sistema. Também serão descritos o desenvolvimento geral do projeto, seus riscos e suas características.

3.1 MODELAGENS DE SISTEMA

A modelagem de sistema utilizada foi a *Unified Modeling Language* (UML). A UML permite que sejam criados diagramas padronizados, utilizando-se de um conjunto de técnicas de notação gráfica para criação de modelos visuais de sistemas de *softwares* criados sob o paradigma de orientação a objeto. Segundo Grady, James e Ivar (2005, p.xiii), [...] A UML proporciona uma forma-padrão para a preparação de planos de arquitetura de projetos de sistemas, incluindo aspectos conceituais tais como processos de negócios e funções do sistema, além de itens concretos como as classes escritas em determinada linguagem de programação, esquemas de bancos de dados e componentes de *software* reutilizáveis.

3.2 METODOLOGIAS DE DESENVOLVIMENTO DO PROJETO

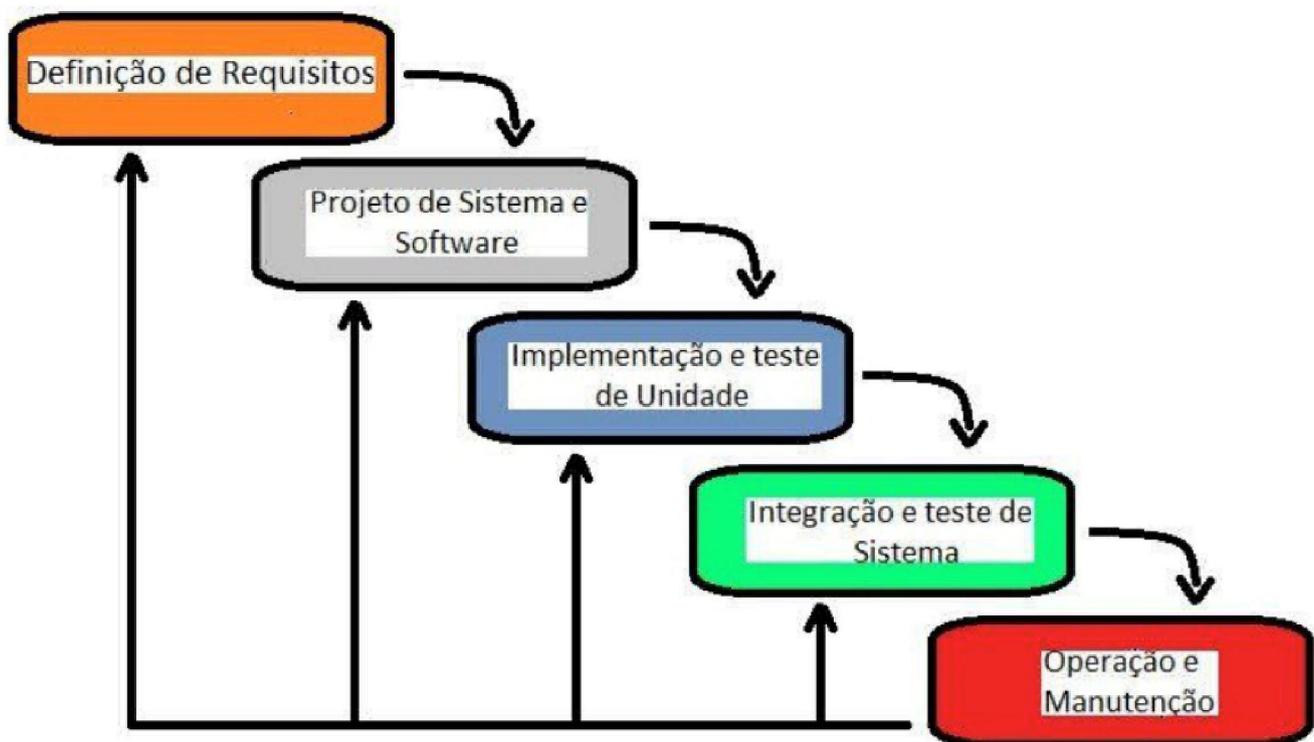
A metodologia escolhida para o desenvolvimento do projeto foi o modelo cascata que é também conhecido como ciclo de vida clássico.

Segundo Rogers S. Pressman (1995, p.32), o paradigma do ciclo de vida requer uma abordagem sistemática, sequencial ao desenvolvimento do

software, que se inicia no nível do sistema e avança ao longo da análise, projeto, codificação, teste e manutenção.

3.2.1 FASES DE DESENVOLVIMENTO DO PROJETO

A figura 001 mostra as fases ou descrição do modelo utilizado ao longo de desenvolvimento.



Fonte: *web*

A seguir serão detalhadas as cinco fases de desenvolvimento de projeto utilizado conforme figura 001 de cascata:

a. Análise e definição dos requisitos ou projeto de sistema e software

Nesta etapa, estabelecem-se os requisitos do produto que se deseja desenvolver, o que consiste usualmente nos serviços que se devem fornecer, limitações e objetivos do *software*. Sendo isso estabelecido, os requisitos

devem ser definidos de uma maneira apropriada para que sejam úteis na etapa seguinte. Esta etapa inclui também a documentação e o estudo da facilidade e da viabilidade do projeto com o fim de determinar o processo de início de desenvolvimento do projeto do sistema; pode ser vista como uma concepção de um produto de *software* e também como o início do seu ciclo de vida.

O projeto do sistema é um processo de vários passos que se centraliza em quatro atributos diferentes do sistema: estruturas de dados, arquitetura do *software*, detalhes precedentes e caracterização das interfaces. O processo de projeto representa os requisitos de uma forma que permita a codificação do produto (é uma prévia etapa de codificação). Da mesma maneira que a análise dos requisitos, o projeto é documentado e transforma-se em uma parte do *software*.

b. Implementação do sistema e teste de unidade

Esta é a etapa em que são criados os programas. Se o projeto possui um nível de detalhe elevado, a etapa de codificação pode implementar-se automaticamente. A princípio, sugere-se incluir um teste unitário dos módulos nesta etapa; nesse caso, as unidades de código produzidas são testadas individualmente antes de passar a etapa de integração e teste global.

c. Teste do sistema

Concluída a codificação, começa a fase de teste do sistema. O processo de teste centraliza-se em dois pontos principais: as lógicas internas do *software* e as funcionalidades externas. Esta fase decide se foram solucionados erros de comportamento do *software* e assegura que as entradas definidas produzam resultados reais que coincidam com os requisitos especificados.

d. Operação e Manutenção

Essa etapa consiste na correção de erros que não foram previamente detectados, em melhorias funcionais e de preferência outros tipos de suporte.

A etapa de manutenção à parte do ciclo de vida do produto de *software* não pertence estritamente ao seu desenvolvimento. Melhorias e correções podem ser consideradas como parte do desenvolvimento.

As etapas descritas são as principais, porém existem sub-etapas dentro de cada etapa, as quais diferem muito de um projeto para outro. Também é possível que certos projetos de *software* exijam a incorporação de uma etapa extra ou a separação de uma etapa em outras etapas.

Com certeza, todas essas variações do modelo Cascata possuem o mesmo conceito básico: a ideia de que uma etapa fornece saída que serão usadas como entradas para a etapa seguinte. Portanto, o processo de desenvolvimento de um produto de *software* de acordo com o modelo Cascata é simples de conhecer e controlar.

Outras atividades que também são levadas em consideração em cada uma das etapas de desenvolvimento do *software*: a documentação, a verificação e a administração das etapas. A verificação, por sua vez, é necessária para que uma etapa forneça os dados corretos para a etapa seguinte. Já a administração, efetua a gestão e o controle da etapa.

3.2.3 VANTAGENS E DESVANTAGENS DO MODELO

Algumas das vantagens de utilizar modelo de cascata: as fases ficam numa ordem sequencial, pois cada etapa deve terminar antes da inicio da próxima etapa. Também, as atividades identificadas durante o processo de

desenvolvimento são fundamentais e ficam na ordem certa.

A desvantagem de modelo de cascata é a impossibilidade de fornecimento de *feedback* entre as fases, bem como da atualização ou redefinição das fases anteriores.

3.3 PLANOS DE ATIVIDADES

O plano de atividades demonstra o fluxo de atividades executadas neste projeto, baseando-se na metodologia escolhida pela equipe. O plano de atividades neste projeto se baseia no diagrama *Work Breakdown Structure* (WBS) e de *Gantt*, que serão detalhados e evidenciados abaixo.

3.3.1 DIAGRAMA WBS

O diagrama WBS realiza a decomposição hierárquica do projeto em fases, entregas e pacotes de trabalho em uma estrutura de árvore, o que permite a visualização do esforço necessário para o desenvolvimento do projeto (CRITICAL TOOLS, 1991).

As etapas da metodologia de desenvolvimento do projeto podem ser visualizadas no diagrama WBS evidenciado no anexo A - *Work Breakdown Structure*. Estas etapas - análise e levantamento de requisitos, documentação, montagem do ambiente e programação/confecção do sistema - estão subdivididas em tarefas que serão executadas entre membros da equipe com uma reunião semanal com orientador do projeto, auxiliando na organização do desenvolvimento do projeto.

3.3.1 DIAGRAMA DE GANTT

O diagrama de *Gantt*, ou calendário de *Gantt*, é um meio de se gerenciar o tempo e esforços dispensados em um projeto. Por meio dele é possível ter uma estimativa de prazos de entrega, quantias de matéria prima a serem utilizadas e esforços gerais empregados em cada tarefa. Trata-se de um instrumento inventado em 1917 por Henry L. GANTT.

Como é possível visualizar no apêndice B - Diagrama de *Gantt*, o diagrama exibe uma lista com as tarefas registradas, suas datas de início e término e uma linha do tempo com as sequências de iterações, registrando todas as tarefas realizadas. As linhas do tempo em cores vermelhas ilustram caminho crítico onde poderia ter algum risco de tempo de desenvolvimento. Para evitar este tipo de problema, alocamos mais recursos de desenvolvimento nesta etapa.

3.4 PLANO DE RISCOS

Em seu livro sobre análise de riscos, Rogers S. Pressman [PRESSMAN95, p.131] destaca seguinte afirmativas.

Quando o risco é considerado no contexto da engenharia de *software*, os três pilares conceituais...[...] estão sempre em evidência. O futuro é nossa preocupação – quais riscos poderiam fazer com que o projeto de *software* saísse torto? A mudança é nossa preocupação – como as mudanças no requisitos do cliente, nas tecnologias de desenvolvimento, nos

computadores de destino e em todas as demais entidades ligadas ao projeto afetarão o sucesso global e o cumprimento do cronograma? Por fim, quantas pessoas devem ser envolvidas e quanta ênfase sobre a qualidade é suficiente?

Considerando essas afirmativas, a equipe optou por planejar e criar a tabela de riscos com as estratégias preventivas e o nível de probabilidade de ocorrência e impacto no projeto. Como mostra a figura 002 abaixo.

Número	Risco	Estratégia	Probabilidade	Impacto
1	Atraso no projeto	Procurar dar continuidade ao projeto dentro dos prazos, efetuando as entregas nas datas pré estabelecidas e dentro de prazos reais e exatamente calculados	Possível	Crítico
2	Falta de conhecimentos técnicos	Partindo da análise dos requisitos técnicos, definir as técnicas que serão empregadas e reunir todo o material necessário ao emprego e aprendizado do conhecimento correspondente	Possível	Catastrófico
3	Ausência de recursos para desenvolvimento	Manter as ferramentas necessárias em bom estado - antivírus nos computadores, por exemplo - ou uma opção alternativa de recurso em cada etapa do projeto	Raro	Insignificante
4	Perda de dados	Manter backup diário dos arquivos vitais do projeto	Baixo	Crítico
5	Ambiente de desenvolvimento mal configurado	Verificar e, se necessário, testar as configurações dos equipamentos, ambiente de desenvolvimento, teste e ferramentas de documentação	Baixo	Baixo
6	Ausência do orientador durante o projeto	No caso de ausência do orientador, buscar com o mesmo uma solução para manter o contato e dar continuidade ao projeto e avaliações periódicas	Baixo	Crítico
7	Acidente com um dos membros da equipe	Procurar chegar com antecedência no dia da avaliação pela banca, para minimizar impactos com acidentes de trânsito. Em caso de acidente prévio, usar uma solução de vídeo conferência para que o membro acidentado possa fazer sua apresentação à distância	Baixo	Crítico
8	Mudança de Requisitos	Fechar um contrato bem detalhada com os requisitos, significa que o projeto vai ser começado depois de análise de todos os requisitos	Possível	Crítico

FIGURA 002 – Diagrama de Riscos

3.5 MATERIAIS

Nesta seção, serão descritos os materiais utilizados para o desenvolvimento geral do projeto. A prioridade de *softwares* utilizados no desenvolvimento se deu para os de licença gratuita, que se mostraram equivalentes aos *softwares* de licença paga, porém o uso de *softwares* de

licença proprietária também foi necessário, tendo em vista a familiarização com estes *softwares* e a necessidade de utilização de funcionalidades não encontradas em *softwares* gratuitos de mesma finalidade.

Os materiais estão organizados em duas partes: materiais de ambiente e desenvolvimento e materiais de modelagem do projeto.

3.5.1 MATERIAIS DE AMBIENTE E DESENVOLVIMENTO

Os materiais de ambientes e desenvolvimento utilizados para o desenvolvimento do sistema e testes serão descritos nesta seção.

3.5.1.1 MÁQUINAS E MATERIAIS

As máquinas e materiais utilizadas para desenvolvimento, testes e virtualização de servidor com a estimativa de custo neste projeto podem ser visualizadas na figura 003.

Máquina	Sistema Operacional	Funcionalidade	Hardware	Custo
<u>Notebook 1</u>	Windows 8	Desenvolvimento da aplicação Testes de desenvolvimento	Modelo: Acer Aspire E1-571-6601 Tela: 15,6 polegadas Processador: Intel Core i3 2.3 GHZ Memória RAM: 4 GB Memória de Disco Rígido: 500 GB	possui
<u>Notebook 2</u>	Arch Linux Fedora 19 Schrödinger's Cat Debian Wheezy 7.2	Desenvolvimento da aplicação Testes de desenvolvimento	Modelo: Megaware Krypton C-Series Tela: 13.1 polegadas Processador: Intel Celeron 1.1 GHZ Memória RAM: 2 GB Memória de Disco Rígido: 320 GB	possui
<u>Notebook 3</u>	X Marvericks	Desenvolvimento da aplicação Testes de desenvolvimento	Modelo: MacBook Pro Tela: 13 polegadas Processador: Intel Core 2 Duo 2.26 GHZ Memória RAM: 2 GB Memória de Disco Rígido: 160 GB	possui
<u>Máquina Virtual (Alocada na UFPR)</u>	Ubuntu Server 13.04	Servidor de Aplicação Servidor de Banco de Dados para testes	Modelo: N/A Tela: N/A Processador: (2 processadores) Intel Xeon 2.4 GHZ Memória RAM: 4 GB Memória de Disco Rígido: 25 GB	gratuito
<u>Servidor Dedicado para Hospedagem d</u>	-	Servidor de Aplicação Servidor de Banco de Dados	-	R\$ 15,00 por mês
<u>Licença de PGModelar</u>	-	Modelagem de Script de Banco de Dados	-	R\$ 8,00
<u>Gasto de Recursos</u>	-	Compra de Materias e Energia	-	R\$ 250,00
TOTAL				R\$ 273,00

Figura 03 - Máquinas Utilizadas no Projeto

3.5.1.2 LINGUAGEM DE PROGRAMAÇÃO

A linguagem de programação utilizada no desenvolvimento deste projeto foi *Java* na versão 7.0. (Oracle Java SE, 2014).

Java é uma linguagem de programação com paradigma orientado a objetos. As aplicações desenvolvidas em *Java* são compiladas para *bytecode* - arquivos de classe, que podem ser executados em qualquer máquina virtual *Java*, independente da arquitetura. (Java Documentation, 2014).

Segundo os autores do livro “Use a cabeça! Servlets & JSP”, Bryan Basham, Kathy Sierra e Bert Bates (2008, p.1), para criar a verdadeira aplicação web, precisa dessas linguagens porque atualmente os usuários preferem uma página dinâmica com interatividades e customização [Traduzido pela equipe].

3.5.1.3 SISTEMA GERENCIADOR DE BANCO DE DADOS

O sistema gerenciador de banco de dados utilizado foi o *PostgreSQL* em sua versão 9.1.

O *PostgreSQL* é um sistema de gerenciamento de banco de dados que utiliza a linguagem de consulta estruturada (*SQL*) como interface, baseado na licença pública geral (*GPL*). Sua utilização pode ser feita via linha de comando ou graficamente. O banco de dados deste projeto está localizado no servidor remoto.

A opção pela escolha do *PostgreSQL* neste projeto foi devida à sua

facilidade de uso e compatibilidade com o ambiente integrado de desenvolvimento utilizado no projeto.

3.5.1.4 SERVIDOR DE APLICAÇÃO

O servidor de aplicação *web* utilizado neste projeto foi o *Apache Tomcat* em sua versão 8.

O *Apache Tomcat* é um servidor de aplicação *web* de código aberto - licenciado sob a licença *Apache License 2.0*, desenvolvido para a plataforma *web*.

O servidor de aplicação deste projeto está localizado na máquina virtual e será disponibilizado online utilizando um domínio próprio para os adotantes.

A escolha pelo *Apache Tomcat* como servidor de aplicação deste projeto se deu pela sua gratuidade de uso, suporte para várias interfaces, desempenho e facilidade de configuração.

3.5.1.5 AUTOMAÇÃO DE COMPILAÇÃO

A automação de compilação neste projeto está sendo realizada pela ferramenta *Maven*.

A configuração do *Maven* é realizada de maneira fácil: um arquivo do tipo *XML (Xtensible Markup Language)* é utilizado para descrever as definições do projeto, tais como: dependências, ordem de compilação e *plug-ins* necessários para a compilação. (MAVEN PROJECT, 2014).

O *Maven* foi utilizado neste projeto por possuir várias funcionalidades que facilitam o gerenciamento de módulos, tais como integração com a IDE *Spring Tool Suite* para desenvolvimento com *Model View Controller* (MVC) e componentes externos no projeto.

3.5.1.6 SISTEMA DE VERSIONAMENTO

O sistema de versionamento utilizado no projeto foi o *Git* na versão 1.8.0. O *Git* é um *software* de versionamento livre, distribuído sob licença *GNU*.

A grande vantagem do uso do *Git* em relação a outros *softwares* de versionamento é que cada diretório de trabalho do *Git* é um repositório que possui um histórico completo de alterações que não depende de acesso a uma rede ou servidor. (JOHN GOLD, 2011).

A escolha do *Git* para este projeto se deu pela sua portabilidade, compatibilidade com protocolos e sistemas atuais e pela sua integração com o ambiente integrado de desenvolvimento utilizado neste projeto.

3.5.1.7 AMBIENTE INTEGRADO DE DESENVOLVIMENTO

O ambiente integrado de desenvolvimento (IDE) utilizado neste projeto foi o *Spring Tool Suite* (STS), baseado no Eclipse.

O STS é um IDE de código aberto, multi-plataforma, direcionado para o desenvolvimento na linguagem de programação *Java*, porém suporta várias outras por meio da utilização de *plug-ins*.

A principal característica do *Spring Tool Suite* (STS) é a orientação de

desenvolvimento baseado em *plug-ins*, adicionando funcionalidades ao *software* e tornando a programação mais ágil e intuitiva. A Figura 004 mostra uma imagem da IDE em execução.

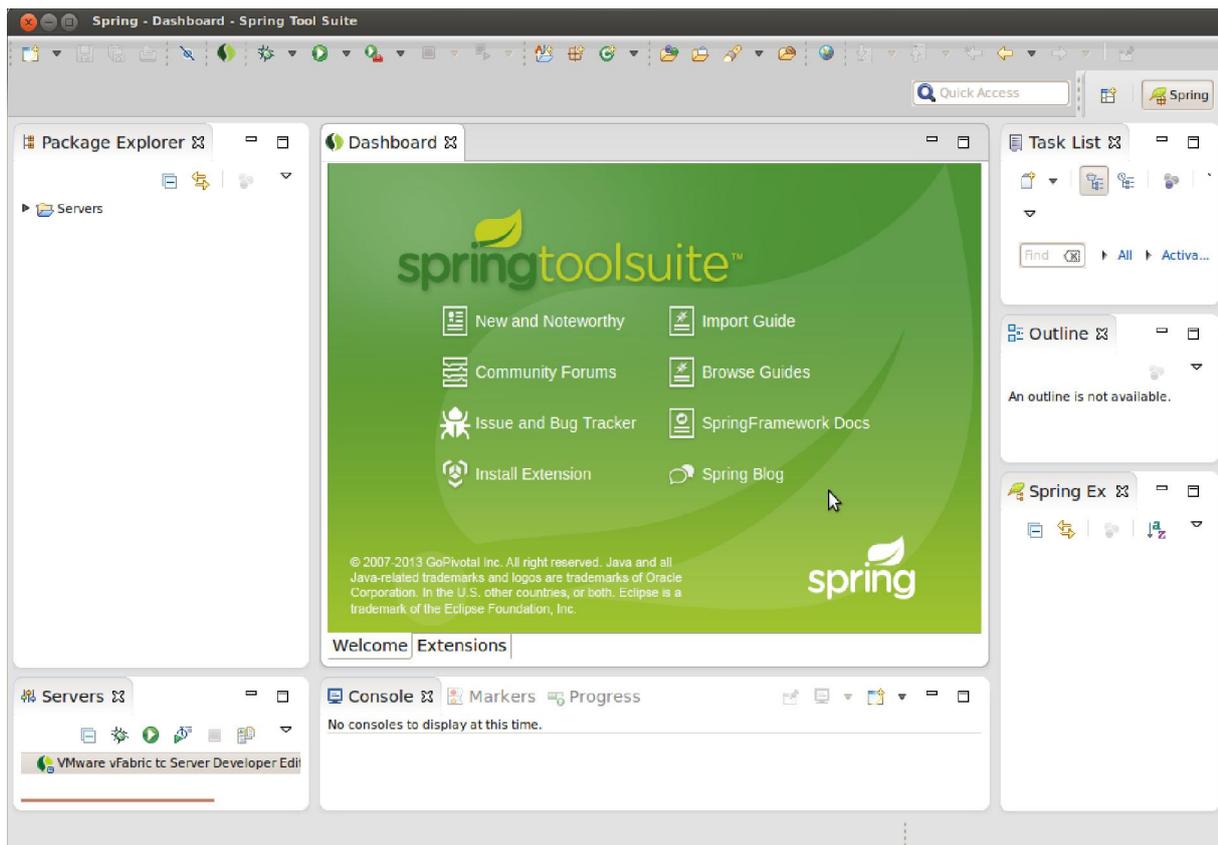


Figura 004 – Spring Tool Suite IDE

3.5.2 MATERIAIS DE MODELAGEM DO PROJETO

Serão descritos nesta seção, os materiais utilizados para a modelagem do projeto.

3.5.2.1 FERRAMENTA PARA CRIAÇÃO DE DIAGRAMAS DE GANTT

A ferramenta utilizada para a criação de diagramas de *Gantt* foi o *ProjectLibre*, uma variação do software *OpenProject*, ambos de licença livre e código aberto. (PM HUT, 2009).

O *ProjectLibre* permite a criação de calendários para gerenciamento de tarefas com diversas funcionalidades adicionais, tais como carga horária por tarefa, pessoal envolvido e previsão de data e hora de entrega, auxiliando a obter uma visão do avanço do projeto.

3.5.2.2 FERRAMENTA PARA MODELAGEM DE DIAGRAMAS UML

A ferramenta para modelagem de *UML* utilizada na criação dos diagramas de sequência foi o *Astah Community* - uma opção completa para a criação de diagramas UML, oferecendo suporte também para a criação de diagramas de atividade, caso de uso, componentes, dentre outros diagramas de análise e design. (MANUAL, 2012).

3.6 DESENVOLVIMENTOS DO PROJETO

A ideia inicial do projeto foi concebida a partir da necessidade de desenvolver um sistema que fosse capaz de auxiliar no processo de gerenciamento de adoção de animais.

Com base nas necessidades principais do cliente solicitante do sistema, foram levantados os requisitos do projeto, ao mesmo tempo em que

era elaborado o plano de trabalho. Sabendo que haveriam pouquíssimas soluções que fossem capazes de resolver essa questão, a equipe buscou *softwares* e sistemas que atendiam a requisitos similares, tais como adoteumamigo.org e animaisos.org aos que deveriam ser satisfeitos.

Após uma análise mais aprofundada a respeito das necessidades e restrições do cliente, foi estabelecido que a melhor alternativa para atender aos requisitos seria um sistema *web*, o qual é acessível de diversos dispositivos e locais. A partir deste ponto, já seria possível desenvolver o sistema, o qual a equipe batizou de Viralata – Sistema de Gerenciamento de Adoção de Animais.

No desenvolvimento efetivo deste sistema, foi necessário montar um ambiente de infraestrutura que suportasse as necessidades da equipe em realizar tarefas de programação e testes. Para isso, utilizando um servidor local, foram criados uma base de dados e um servidor de aplicação, para fins de testes finais.

Com os objetivos definidos e o ambiente de desenvolvimento e testes configurado, passou-se a elaborar a documentação necessária para construção do sistema, proporcionando uma visão mais ampla do projeto como um todo e norteando como e quais atividades serão desenvolvidas. Esta documentação está especificada através dos itens encontrados nos apêndices, e estão dispostos de acordo com a Tabela 1.

Apêndice	Descrição
A	Diagrama WBS
B	Diagrama Gantt
C	Diagrama de Casos de Uso
D	Especificação de Casos de Uso
E	Diagrama de Classes
F	Diagrama de Sequência
G	Diagrama de Estado
H	Diagrama Entidade Relacionamento
I	Tabela de Dicionário de Dados
J	Prototipagens das Telas

TABELA 1 – Listagem de Apêndices

A fim de garantir um bom desempenho do sistema durante o desenvolvimento, foi iniciada a fase de testes, a qual acabou apontando possíveis defeitos e inconformidades no funcionamento do programa, permitindo que os mesmos pudessem ser corrigidos, além de indicar pontos onde seria viável uma otimização, ou mesmo mudança de modo de funcionamento. Para tanto, o *servidor* foi testado primeiramente, a fim de se descobrir se haviam falhas que pudessem comprometer o funcionamento do sistema; os testes foram feitos através do acesso de suas funções via navegador, e verificando se eram feitas as devidas transações com o banco de dados, e se esses dados eram inseridos e recuperados corretamente. Em seguida, testou-se o sistema, para que fosse verificada sua correta

comunicação e integração com o *servidor*, além de ser testado seu comportamento em diferentes dispositivos, de diferentes resoluções de tela.

4. ANÁLISE DE REQUISITOS

Neste capítulo serão detalhados os requisitos e os dados levantados durante a entrevista com o público alvo de sistema e os limites/escopos de projeto.

4.1 REQUISITOS PRINCIPAIS

O principal problema constatado durante as entrevistas foi a questão da falta de espaço e pouca procura por animais de idade mais avançada e com determinadas características físicas.

Para se desenvolver um sistema que possa atuar de uma maneira eficiente e eficaz, os dados dos animais devem ser corretamente organizados e classificados, a fim de se ter um melhor controle dos mesmos, e também para que outros setores relacionados possam ter acesso à esses dados, simplificando uma movimentação e controle dos animais para outros locais.

A partir deste requisito, surgiu a necessidade de se ter uma correta distribuição do espaço disponível para abrigar os animais por conta do grande volume de recolhimentos, pois, segundo dados de uma pesquisa feita pelo órgão *JusBrasil*, há mais de 30 milhões de animais abandonados no Brasil. (ANDA, 2013).

Em seguida, é necessário que seja definida uma estratégia para alavancar a adoção de animais considerados de baixa procura, tais como viralatas, animais com deformidades ou de maior porte, etc. Estes são os que compõem a maioria dos animais retidos no abrigo da prefeitura.

E, por fim, também é necessário definir a organização de feiras de adoção, as quais são pouco comuns. Neste requisito, há também o ponto que poucos animais são levados para estas feiras, o que as tornam pouco eficientes quando realizadas.

4.1.2 ITENS A SEREM CONSIDERADOS:

A seguir, são listados os itens a serem considerados durante o desenvolvimento e implementação deste projeto.

- Pouco espaço para os animais, possuindo uma capacidade de 'estoque' limitada;
- Maioria dos animais sob tutela de terceiros; em caso de falta de espaço os animais acabam sendo devolvidos para a rua;
- Feiras de adoção raras (1 vez por mês, cerca de 10 por ano), uma das únicas maneiras de se levar os animais à adoção;
- Preferência por filhotes, raças pequenas e pelagem clara; animais fora desses requisitos acabam ficando em estoque;
- Dos animais cuidados pela prefeitura, uma porcentagem muito baixa é levada para adoção nas feiras;
- Necessidade de um mecanismo de controle de vacinação, controle de verminoses e castração;
- Necessidade de se manter outros dados dos animais, tais como: nome, idade aproximada, raça, espécie (cão, gato, etc.), porte, cor, pelagem,

características físicas, resenha, foto, localidade (onde ele está sob cuidado), registro de vacina anti-rábica e múltipla, e data de ambas, registro de castração, controle de ecto e endoparasitas, nome da medicação aplicada e qual finalidade (vermífugo, anti-pulgas, etc), dados de controle e existência de patologias (contra sarna, por exemplo);

- Necessidade de controle de abandono de animais previamente adotados (monitoramento);
- Necessidade de se informar a situação do animal (óbito, desaparecido, etc.);
- Registro dos adotantes e temporários, e classificação (pessoa física, ONG, órgão público, etc.);
- Necessidade de manter dados pertinentes de adotantes, tais como: CPF, nome, RG, e-mail, renda familiar, número de pessoas acima e abaixo de 12 anos residindo em casa, complemento (casa, apartamento), se existe espaço adequado, preferência por raça, sexo ou porte de animal, se o interessado oferece transporte, etc.

4.2.1 SUGESTÕES

Os itens a seguir são algumas das sugestões levantadas durante levantamento de análise de requisitos.

- Contagem de total de adoções e adoções em um certo período.

- Histórico de adotante e temporário (ex: quem não cuida não pode adotar futuramente);
- Implementação futura
 - i. Característica marcante do animal (cicatriz, por exemplo).
 - ii. Chipagem (Identificação dos animais usando *microchip*).
 - iii. Relatório de relação dos animais sob tutela da Prefeitura.

4.2 PROPOSTAS DE SOLUÇÃO

Partindo dos requisitos anteriormente levantados, pensou-se em uma solução para cada um, em conjunto com a equipe que cuida dos animais, com o orientador da equipe de desenvolvimento e entre os próprios membros da referida equipe. Após a apresentação de diversas ideias e conceitos, chegou-se a diversas propostas de solução.

4.2.1 GERAÇÃO DE RELATÓRIOS GERENCIAIS

Para melhor controle de sistema e ajuda na tomada de decisões gerenciais, o sistema terá os seguintes tipos de relatórios:

- Relação dos animais adotados e seus tutores;
- Relação das áreas visitadas e maior incidência de abandonos/adoções
- Relatório geanalítico, um tipo de *Business Intelligence* (BI) no formato de relatório para mostrar as áreas utilizando mapa.

- Relatório por período entre determinadas datas.

Através da geração de relatórios gerenciais é possível obter um melhor controle e organização dos dados referentes aos animais e recursos utilizados, tais como medicação, espaço disponível para abrigo, etc.

4.2.2 MAPEAMENTO DAS ÁREAS VISITADAS ATRAVÉS DE GEORREFERENCIAMENTO.

De acordo com o grupo, o ônibus usado para o transporte dos veterinários realiza um trajeto predeterminado pelos bairros de Curitiba. Foi sugerido, então, que os trajetos efetuados fossem mapeados, e que com base nesses trajetos, fossem indicados e classificados os pontos de importância, ou seja, aqueles que apresentam alguma característica relevante (como maior índice de animais abandonados, maior frequência de recolhimento, etc). Também com base nesse mapa, foi sugerido ao grupo que pudessem ser indicados pontos onde há uma melhor aceitação de uma possível feira de adoção (locais que concentrem muitos adotantes conhecidos), o que auxiliaria na otimização do processo de organização das feiras anteriormente citadas. Isso acarreta também numa liberação de espaço do abrigo da prefeitura, ou mesmo da criação de novos espaços. Sabendo onde se localiza uma quantidade satisfatória de interessados em adoções, a probabilidade de se conseguir um auxílio com espaços cedidos por terceiros aumenta.

4.2.3 CRIAÇÃO DE UM PRONTUÁRIO DE ATENDIMENTO PARA ANIMAIS

Para manter o estado de saúde de animais, o sistema vai manter cadastro de todos os medicamentos feito no módulo de prontuário.

4.2.4 PLANO DE AÇÃO E PROJETO DE RESGATE DOS ANIMAIS

O plano de ação consiste basicamente em um itinerário através de uma rota fixa pelos bairros de Curitiba, do qual os animais (cães em sua maioria) são recolhidos e passam por uma rotina de avaliação, que inclui controle de ectoparasitas e endoparasitas, castração e vacinação anti-rábica e múltipla. Após isso, os mesmos são recolhidos e mandados para o canil municipal.

Uma vez sob tutela da prefeitura, os animais são mantidos em abrigos (o canil municipal é o principal, havendo também o jardim zoológico de Curitiba), até que sejam adotados diretamente por ONG's, pessoas físicas, ou seja, enviados para feiras de adoção, realizadas ocasionalmente.

No capítulo 5 – Apresentação do Sistema serão descritos os aspectos gerais e específicos do sistema desenvolvido.

5. APRESENTAÇÃO DE SISTEMA

Neste capítulo será explicado como instalar o sistema a partir do CD de instalação, um detalhamento do sistema em funcionamento com imagens das páginas *web* (páginas JSP) através das quais serão formalmente apresentadas todas as funcionalidades do sistema, havendo também a explanação de determinados recursos. Também será explicado o procedimento necessário para o uso do sistema, e, por fim, um detalhamento de como é feita a integração do sistema desenvolvido com um servidor *Java*.

5.1 INSTALAÇÃO E CONFIGURAÇÃO DO AMBIENTE DE DESENVOLVIMENTO

Tanto a instalação quanto a configuração do ambiente de desenvolvimento serão descritas para que sejam feitas dentro do ambiente *Windows*, embora a instalação também funcione no ambiente *Linux*, que não será descrita em virtude da existência de um grande número de distribuições (variantes) do sistema, inviabilizando uma descrição precisa de instalação para cada caso. De qualquer maneira, bastará para o usuário de *Linux* instalar os *softwares* descritos da forma que sua distribuição permitir e seguir os passos tal como é feito no ambiente *Windows*.

Antes de tudo, dois *softwares* são necessários: o servidor de aplicação *Tomcat* e a interface gráfica *pgAdmin* para o sistema gerenciador de banco de dados *PostgresSQL*. As versões utilizadas são a versão 8.0 para o *Tomcat* e v1.18.1 para o *pgAdmin*. Estes *softwares* podem ser baixados através dos sites

<<http://tomcat.apache.org/download-70.cgi>. e

<<http://www.postgresql.org/ftp/pgadmin3/release/>. respectivamente. É

fundamental ter o kit de desenvolvimento *Java* (JDK) instalado, o qual pode ser baixado no site da *Oracle*, em

<<http://www.oracle.com/technetwork/pt/java/javase/downloads/jdk7-downloads-1880260.html>.

Para a configuração do ambiente de desenvolvimento, outro *software* necessita ser instalado, a IDE *Spring Tool Suite* (STS), disponível em <https://spring.io/tools/sts>.

A instalação do *Java* JDK e do STS é simples e não necessita muita explicação; entretanto, ao instalar o STS, o mesmo acompanha um servidor de aplicação próprio como padrão. Assim sendo, para evitar transtornos de configuração, deve-se atentar para definir que a porta padrão do serviço seja a de número 8080, a qual é utilizada pela aplicação. Quanto ao *pgAdmin*, o usuário e senha utilizados são as palavras '*postgres*' e '*admin*' respectivamente, mas é possível alterar a configuração. É importante verificar se a porta mencionada não está em uso por outros aplicativos. Em caso afirmativo, recomenda-se encerrar estes processos ou então mudar o número da porta dos outros aplicativos.

O CD de instalação contém os seguintes itens:

- Uma pasta intitulada *Viralata*
- Uma pasta intitulada *Documentação*
- Um arquivo de extensão *.war* chamado *viralata*

- Uma pasta chamada TCCViralata
- Um arquivo de extensão .sql chamado viralata
- Um arquivo de extensão .txt chamado leia-me

Primeiramente, com o *pgAdmin* instalado, é necessário criar uma base de dados chamada viralata. Com isso, a importação da base de dados usada no projeto, que está no arquivo viralata.sql, deve ser feita para dentro da base recém criada. Com a base de dados já configurada corretamente, o próximo passo é abrir o arquivo viralata.war e alterar se for necessário, as configurações da base de dados para o projeto se a senha e usuário do pgAdmin tiverem nomes diferentes. Para isso, utiliza-se um programa de extração de arquivos compactados, como o *Winrar*, por exemplo, para abrir o arquivo viralata.war e então navegar até o diretório WEB-INF/classes/util. Feito isso, haverá um arquivo chamado *ConnectionFactory.java* que poderá ser aberto em qualquer editor de textos. Dentro dele, os valores para usuário e senha podem ser alterados para refletir a configuração atual do *pgAdmin* de quem for instalar o sistema. Caso não seja usada senha para o *pgAdmin*, basta deixar o campo da variável senha sem nenhum argumento.

Com o *Spring Tool Suite* instalado e aberto, o projeto do servidor, que é a pasta TCCViralata deverá ser importado por meio da opção *import* dentro do menu *file* da IDE. Feito isso, o projeto aparecerá no *Project Explorer* do STS. Em seguida, o próximo passo é configurar o *Apache Tomcat* para funcionar dentro da IDE, o qual pode ser feito clicando com o botão direito do mouse sobre o projeto e em seguida em *new* e em *other*, onde será possível adicionar um servidor, escolhendo para isso a opção *server*. Será requisitado o diretório

onde o *Apache Tomcat* está instalado e ao término deste processo o servidor estará adicionado ao projeto. Deve-se ter atenção para parar o servidor caso ele esteja rodando, para que o STS passe a gerenciar a inicialização do servidor.

O último passo é abrir o projeto. Ao clicar com botão direito na raiz do projeto, deve-se selecionar a opção '*Run As*'; aparecerá uma janela com diversas opções de execução, devendo-se selecionar a opção '*Run on Server*'. Assim, o sistema entrará em execução, e automaticamente será carregada a página principal do sistema. Na seção a seguir serão explicadas as páginas/telas do sistema.

5.2 TELAS DO SISTEMA

A primeira página do sistema a ser carregada é página de *login* (acesso ao sistema).

Após o *login* bem sucedido, o usuário será encaminhado para a página principal do sistema, a qual difere dependendo do tipo do usuário. Nesta página algumas informações gerais são exibidas e é possível navegar para outras páginas ou fazer *logout* (sair do sistema) por meio de um menu localizado na parte superior da página. Caso o perfil do usuário seja o de adotante, ele é redirecionado para a área de adotante com os seguintes menus: "Perfil", onde são mostrados os dados cadastrais de adotante, sendo possível realizar a atualização de dados; a página "Minha Solicitações", onde são carregados todos os históricos de solicitações de adoções de animais realizados pelo

referido usuário. O menu “Busca” carrega uma página para que o adotante possa buscar um animal pelo nome, raça, cor, ou espécie, exibindo em seguida o resultado de busca; se a busca retorna dados de animais, há um botão para mostrar detalhes do animal. Os animais são classificados de acordo com seu *status*, o qual determina as ações possíveis para o mesmo. Se o *status* for “livre”, o adotante pode clicar no botão “solicitar” e o animal automaticamente é colocado na fila de adoção; caso o status esteja como “adotado”, o adotante não poderá efetuar uma solicitação; no caso do *status* “em processo de adoção”, o adotante pode ficar na fila; caso o candidato anterior à adoção não tenha sua solicitação aprovada, o segundo na fila pode adotar. Existe também, na área de adotante a tela de “Dúvidas”, para tirar dúvidas sobre perguntas frequentes e menu “Contato” para entrar em contato com o administrador do sistema através de um formulário com mensagem.

Porém, se o acesso for feito por administrador de sistema, o usuário será redirecionado para a área de administrador, com outros menus e telas diferentes. O primeiro menu, denominado “cadastros”, contém outros sub-menus, tais como cadastro de animais, cadastro de feiras e cadastro de prontuários, onde todos exibem um formulário com os campos a serem preenchidos. Também em sequência existem mais três sub-menus para manter os sub-menus mencionados anteriormente, o próximo é menu “Adoções” contém dois sub-menus de solicitações de adoções e adoções pendente. O terceiro menu na área de administrador é menu “Relatórios” o menu relatórios tem quatro sub-menus a seguir, Relatório de Adoções, Relatório de Abandono, Relatório de Adotante e por fim sub-menu Relatório Geoanalítico, por último na barra do menus está o menu “Adotante” onde ficam as listas de adotantes no

sistema e também nesta página o administrador pode bloquear ou desbloquear um adotante conforme a necessidade. Tentativas de acesso ao sistema sem que ocorra uma validação de dados por meio da página de *login* são barradas, sendo exibida uma mensagem de acesso negado. Em tentativas de acesso à páginas não existentes dentro do sistema também haverá redirecionamento para uma página específica que informará o usuário sobre o erro.

Viralata

Cadastre-se

Duvidas

Contato

Login

VIRALATA

Viralata: Sistema de Adoção de Animais

Login de Usuario

Login

Usuário

Senha

Senha

Entrar

Recuperar Senha

Novo Usuário? Realizar cadastro no Viralata

© Viralata 2014 todos os direitos reservados www.viralata.com.br

Figura 005. Tela inicial/principal do sistema.

Como mostrado na figura acima, ao iniciar o sistema ou ao entrar no domínio do sistema, automaticamente será carregada a tela principal e se o usuário já tem cadastro pode entrar utilizando seus dados cadastrais, senão o usuário deve fazer um cadastro no sistema através de uma inscrição simples clicando o menu "Cadastre-se" ou clicando o link "Novo Usuário? Realizar

cadastro no Viralata”, depois disso, um formulário vai ser aberto para preencher com os dados de usuário. A próxima figura 006 : “selecionar tipo de pessoa” irá mostrar os passos para fazer cadastro como novo usuário.

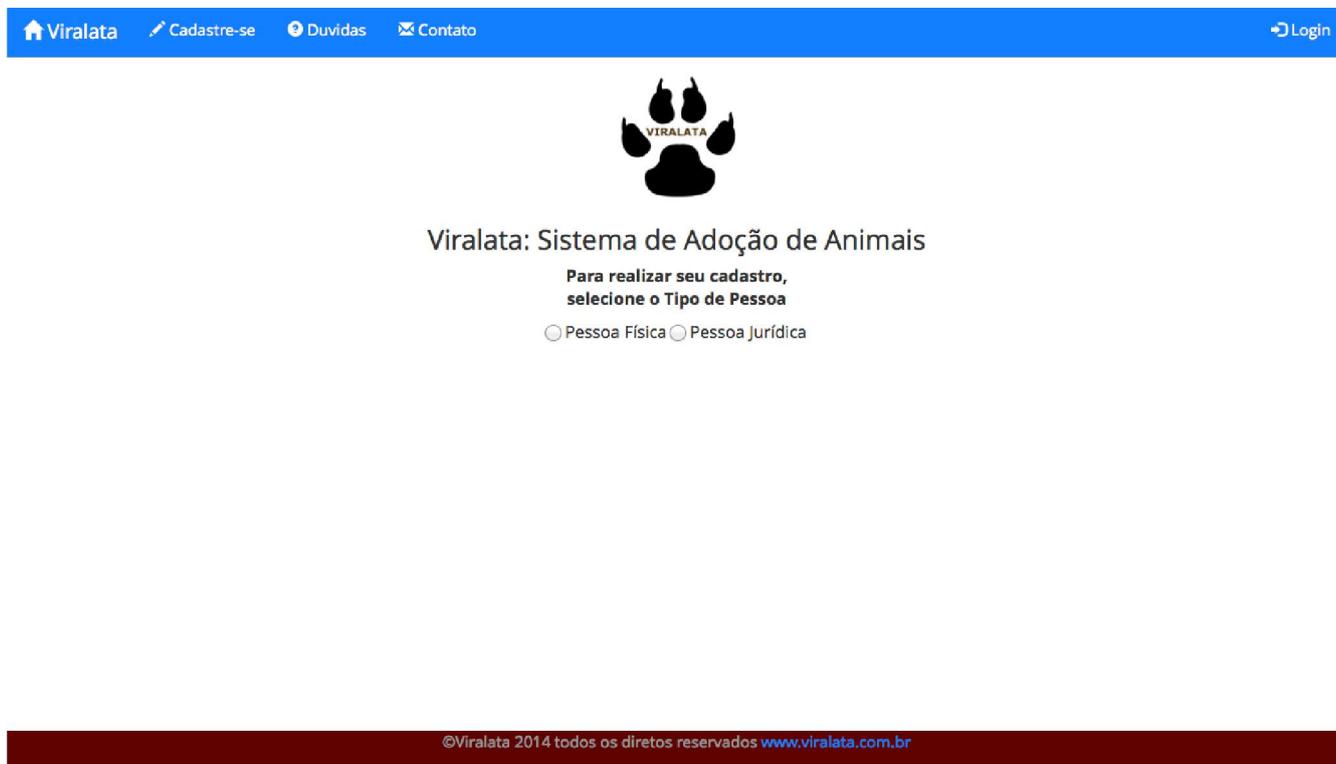


Figura 006. Tela “Selecionar tipo de pessoa”.

Depois de clicar no menu “*Cadastre-se*”, a próxima tela a ser carregada é tela para “*selecionar tipo de pessoa*” - para simplificar o processo, o sistema aceita *dois tipos de usuário*: pessoa física ou jurídica. Para redirecionar ao formulário de cadastro, basta selecionar uma opção e automaticamente o sistema abrirá o formulário de cadastro como mostram as figuras 06,07 e 08.

[Viralata](#) [Cadastre-se](#) [Duvidas](#) [Contato](#) [Login](#)



Viralata: Sistema de Adoção de Animais

Cadastro de Pessoa Física, Campos com * são obrigatorios

Nome*	Nome Completo
CPF*	CPF
RG*	RG
E-mail*	E-mail
Data Nascimento*	Data de Nascimento
Telefone Celular*	Telefone Celular

Figura 008. Formulário para cadastro de pessoa física parte 1.

[Viralata](#) [Cadastre-se](#) [Duvidas](#) [Contato](#) [Login](#)

Endereço Atual

CEP*	Cep	Consultar CEP
Rua:	Rua	
Numero:	Número	
Complemento:	Complemento	
Bairro:	Bairro	
Cidade:	Cidade	
Estado:	Estado	
Tipo Moradia*		
Area Moradia*	Área em m ²	
Telefone*	Telefone Residencial	

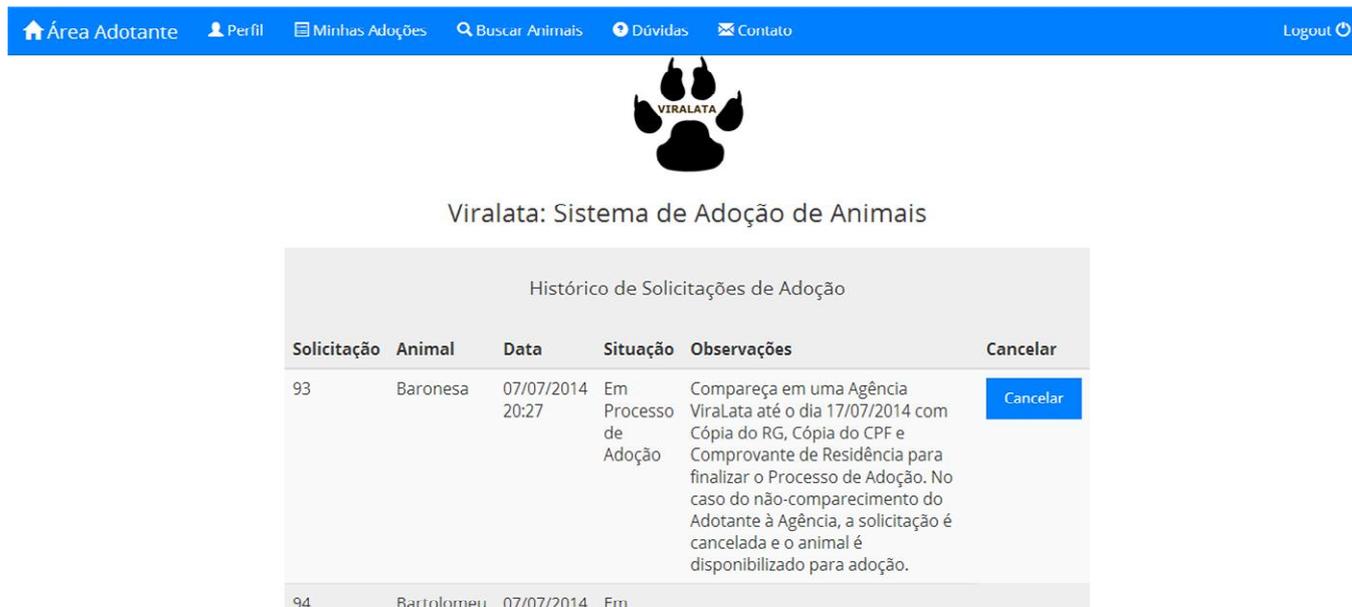
Figura 009. Formulário para cadastro de pessoa física parte 2.

Dados Socioeconômicos	
Renda Familiar*	<input type="text" value="Renda Familiar"/>
Número de Crianças em Casa*	<input type="text" value="Número de Crianças em Casa"/>
Quantidade de Animais*	<input type="text" value="Quantidade de Animais"/>
Usuário	
Login*	<input type="text" value="Login"/>
Senha*	<input type="text" value="Senha"/>
Confirmar Senha*	<input type="text" value="Confirmar Senha"/>
<input type="button" value="Realizar Cadastro"/> <input type="button" value="Limpar"/>	

© Viralata 2014 todos os direitos reservados www.viralata.com.br

Figura 010. Formulário para cadastro de pessoa física parte 3.

Depois de preencher todos os campos obrigatórios com dados válidos, ao clicar no botão “Realizar Cadastro”, o sistema vai validar e armazenar os dados no banco de dados para usos afins. Rapidamente, o usuário receberá um e-mail de confirmação de cadastro no sistema. O formulário de cadastro de pessoa jurídica é semelhante com o cadastro da pessoa física, com a diferença em alguns campos, como: CNPJ, Razão Social, Nome fantasia etc. As figuras a seguir irão mostrar recursos de área de adotante(usuário de sistema).



Histórico de Solicitações de Adoção

Solicitação	Animal	Data	Situação	Observações	Cancelar
93	Baronesa	07/07/2014 20:27	Em Processo de Adoção	Compareça em uma Agência Viralata até o dia 17/07/2014 com Cópia do RG, Cópia do CPF e Comprovante de Residência para finalizar o Processo de Adoção. No caso do não-comparecimento do Adotante à Agência, a solicitação é cancelada e o animal é disponibilizado para adoção.	Cancelar
94	Bartolomeu	07/07/2014	Em		

Figura 011. Tela home adotante.

Ao logar no sistema como adotante, a primeira página a ser carregada é área de adotante, onde terão alguns menus(módulos) com funções personalizadas, como é exibida na figura 009.



Atualizar Perfil Adotante

Dados Pessoais

Nome	Ricardo Alves
CPF	456.789.415-61
RG	15.222.456-4
Email	mmazurky@gmail.com
Data Nascimento	18/04/1990
Celular	(41)9838-1284

Figura 011. Perfil adotante/usuário parte 1.

Área Adotante Perfil Minhas Adoções Buscar Animais Duvidas Contato Logout

Endereço Cadastrado

CEP	80020-070
Rua	FREDERICO FARIA DE OLIVEIRA
Numero	50
Complemento	APTO 10
Bairro	CENTRO
Cidade	AGUDOS DO SUL
Estado	PR
Tipo Moradia	Avenaria
Area Moradia	1450.0
Telefone Fixo	(41)3025-7841

Endereço Adicional 1

CEP	Cep
-----	-----

Consultar CEP

Figura 012. Perfil adotante/usuário parte 2.

Dados Socioeconômicos

Renda Familiar	5000.0
Crianças em Casa	2
Quantidade de Animais	4

Atualizar Dados de Login

Login	pf
Senha Atual	Senha Atual
Nova Senha	Nova Senha
Confirmar Senha	Confirmar Nova Senha

Redefinir Senha

Alterar Limpar

Figura 013. Perfil adotante/usuário parte 3.

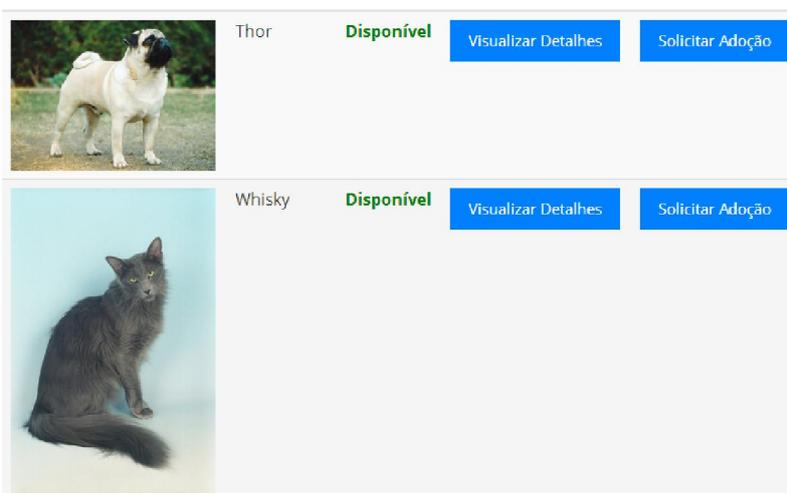
Usuário(adoptante) pode atualizar seus dados cadastrais através de uso de menu “Perfil” como mostram as imagens de figura 011,012 e 013 acima.

Figura 014. Tela para buscar animais.

Ao clicar no menu “Buscar Animais”, a tela apresentada acima será carregada e o adotante tem quatro (4) opções de busca que são: nome, espécie, raça ou cor.

Imagem	Nome	Status	Detalhes	Solicitar
	Baronesa	Disponível	Visualizar Detalhes	Solicitar Adoção

Figura 015 Resultado de busca animais parte 1.



©Viralata 2014 todos os direitos reservados www.viralata.com.br

Figura 016 Resultado de busca parte 2.

Depois que o adotante clicar em busca, as figuras acima serão carregadas exibindo os resultados; caso nenhuma opção seja selecionada, uma página com mensagem de aviso aparecerá informando que não existe a opção selecionada.

A screenshot of a contact form on the Viralata website. The form is titled 'Favor enviar suas reclamações e sugestões através deste formulário.' (Please send your complaints and suggestions through this form). It includes fields for 'Nome' (Name), 'E-mail', and 'Assunto' (Subject), which is a dropdown menu currently showing 'Selecione' (Select). Below these is a large text area for 'Mensagem' (Message). At the bottom, there are 'Enviar' (Send) and 'Limpar' (Clear) buttons. A footer note says 'Por gentileza, escreva suas sugestões para melhor atendê-lo.' (Please write your suggestions to better serve you). The website's navigation bar is visible at the top, and a copyright notice is at the bottom.

Figura 017. Tela para contato.

O usuário poderá entrar em contato com o administrador do sistema através do preenchimento do formulário acima.



[Viralata](#) [Cadastre-se](#) [Duvidas](#) [Contato](#) [Login](#)

O melhor ambiente para adoção de animais

Viralata é um sistema voltado com objetivo de simplificar e agilizar todos os processos de adoção de animais. Ao tornar-se membro, terá acesso aos recursos disponibilizados gratuitamente.

[Cadastre-se hoje!](#)

Perguntas Frequentes

Como posso adotar?
Resposta: Após tornar-se um membro em nosso sistema, terá acesso para buscar e solicitar a adoção de vários tipos de animais de estimação.

Como posso doar um animal?
Resposta: Para doar um animal, por gentileza entrar em contato conosco preenchendo o formulário abaixo, informando ponto de recolhimento.

[Doar Animal!](#)

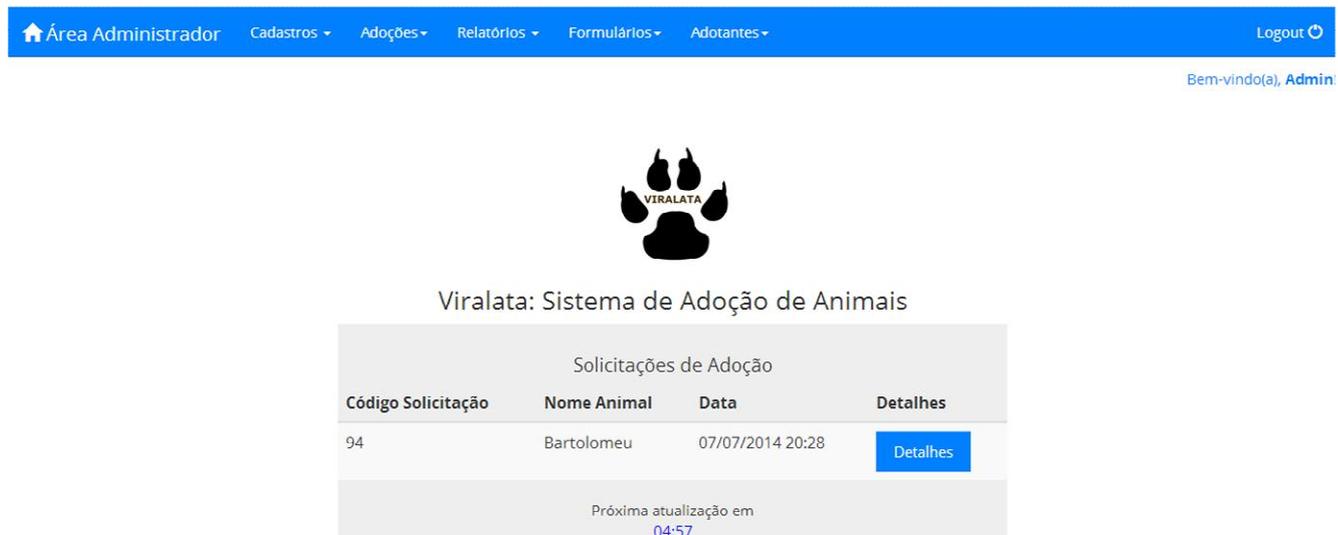
Solicitei uma adoção quanto tempo demora para minha solicitação ser aprovada?
Resposta: As solicitações de adoção levam em média de cinco(5) a dez(10) dias úteis para serem analisadas e aprovadas.

Quantos animais posso adotar?
Resposta: Um adotante pode adotar mais de duas a tres animais, forem adotante com uma reputação negativa, poderia ser bloqueado.

© Viralata 2014 todos os direitos reservados www.viralata.com.br

Figura 018. Tela para tirar duvidas frequentes.

Para sanar algumas dúvidas, existe uma página com algumas perguntas e respostas frequentes, como mostra a Figura 019 acima. A seguir serão mostradas as imagens das telas do administrador do sistema.



Área Administrador Cadastros Adoções Relatórios Formulários Adotantes Logout

Bem-vindo(a), Admin!

VIRALATA

Viralata: Sistema de Adoção de Animais

Solicitações de Adoção

Código Solicitação	Nome Animal	Data	Detalhes
94	Bartolomeu	07/07/2014 20:28	Detalhes

Próxima atualização em 04:57

Figura 019. Tela Home Administrador.

A primeira página a ser mostrada quando um administrador realiza *login*, aparece a lista de solicitações feitas pelos adotantes, como mostra a figura



Área Administrador Cadastros Adoções Relatórios Formulários Adotantes Logout

Bem-vindo(a), Admin!

- Cadastrar Animais
- Cadastrar Feiras
- Cadastrar Prontuário
- Manter Animais
- Manter Feiras
- Manter Prontuários

VIRALATA

acima.

Figura 020. Menu Cadastros.

A área de administrador tem menus e sub-menus para gerenciar os recursos de sistema como mostra a figura acima; o menu cadastro serve para cadastrar animais, feiras, prontuários e também para manter os respectivos sub-menus.



Figura 021. Menu Adoções.

O menu adoções funciona para o administrador visualizar e manter alguns sub-menus, como mostra a imagem acima.



Figura 022. Menu Relatórios.

Para gerar relatórios, o sistema conta com o menu Relatórios, que funciona para gerar quatro tipos de relatórios diferentes.

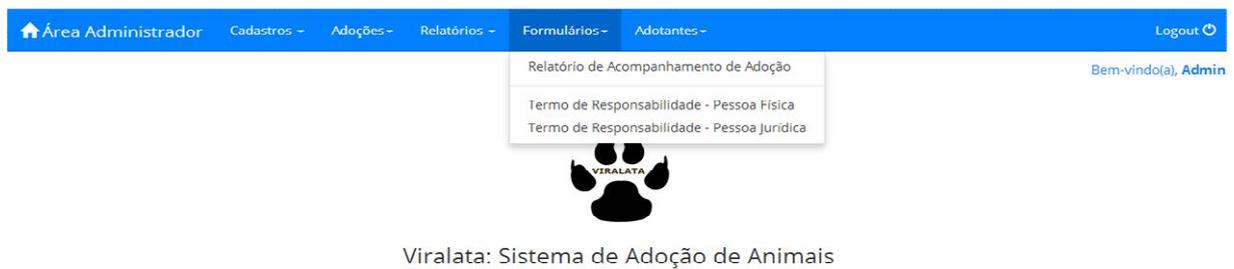


Figura 023. Menu Formulários.

Foi implementado também um menu específico para gerar formulários a fim de executar alguns processos, como mostra a figura 024 acima.



Figura 025. Menu Adotantes.

Caso haja a necessidade de um administrador bloquear ou desbloquear um adotante, basta abrir o sub-menu "manter adotantes" que vai carregar a lista de adotantes cadastrados no sistema e clicar no botão bloquear ou desbloquear conforme figura 026.



Figura 027. Detalhes liberar adoções parte 1.

Área Administrador Cadastros - Adoções - Relatórios - Formulários - Adotantes - Logout

Cidade: Curitiba

Estado: PR

Tipo Moradia: Alvenaria

Área Moradia: 900.0

Telefone: (41)3029-1284

Reputação

Reputação: 243

<Voltar Liberar Adoção

©Viralata 2014 todos os direitos reservados www.viralata.com.br

Figura 028. Detalhes liberar adoções parte 2.

Depois que o administrador clicar no botão “Detalhes”, as páginas acima serão carregadas para visualizar e analisar os dados de animais e de adotante; no final, o administrador pode decidir e clicar no botão liberar adoção assim que o fluxo terminar. Este módulo pode ser encontrado no menu “Adoções -> sub-menu Liberar Adoções”.

Área Administrador Cadastros Adoções Relatórios Formulários Adotantes Logout

Bem-vindo(a), Admin!

Viralata

Viralata: Sistema de Adoção de Animais

Dados do Animal

Nome:

Data de Nascimento*:

Sexo: **Porte:**

Espécie: **Raça:**

Cor: **Pelagem:**

Definições: Possui Não Possui

Figura 029. Tela cadastrar animais parte 1.

Foto: nenhum arquivo selecionado

Prontuário

Registro Anti-Rábica:

Registro Múltipla:

Registro Castração:

Registro Parasitas:

Patologia:

Veterinário Responsável:

Data Registro:

Figura 030. Tela cadastrar animais parte 2.

Para cadastrar um animal, basta clicar no menu cadastros, após selecionar sub-menu "cadastrar animais", os formulários de telas acima serão

carregados e o administrador poderá preencher os campos necessários e clicar no botão realizar cadastro; ao clicar no botão, os dados serão validados e armazenados no banco de dados se forem válidos, caso contrário uma mensagem de erro será mostrada na tela para corrigir.

Área Administrador Cadastros Adoções Relatórios Formulários Adotantes Logout

Bem-vindo(a), Admin!

Viralata

Viralata: Sistema de Adoção de Animais

Cadastrar Feira

Nome Feira:

Data Início:

Horário Início:

Data Encerramento:

Horário Encerramento:

Figura 031. Tela cadastro de feiras parte 1.

Organizador Responsável:

Possíveis Locais da Feira

Local:

Animais Relacionados

Imagem	Nome	Raça	Selecionar
	Pit Bull	Alano Espanhol	<input type="checkbox"/>
	Wild Cat	Abissínio	<input type="checkbox"/>

Divulgação

Enviar convite para os Adotantes: Sim Não

Realizar Cadastro **Limpar**

Figura 032. Tela cadastro de feiras parte 2.

Ao abrir o sub-menu “cadastrar feira”, o formulário de figuras acima será carregado e o administrador poderá preencher os campos e selecionar os animais a serem levados para realização de feira.

Área Administrador Cadastros Adoções Relatórios Formulários Adotantes Logout

Viralata: Sistema de Adoção de Animais

Selecione um Animal para atualizar seus Dados

Animais: Bartolomeu

Remover Imagem

Nome: Bartolomeu

Figura 033. Manter dados de animais parte 1.

Para atualizar ou excluir os dados do animal, o administrador pode

Área Administrador Cadastros Adoções Relatórios Formulários Adotantes Logout

Código: Gato13

Situação: Em processo de Adoção

Data de Nascimento: 02/03/2012

Sexo: Fêmea Porte: Grande

Espécie: Gato Raça: California spe

Cor: Amarelo Pelagem: Média

Deformidade: Deformidade

Resenha: Resenha

Alterar Foto: Escolher arquivo Nenhum arquivo selecionado

Alterar

Figura 034. Manter dados de animais parte 2

selecionar um animal específico na lista de animais cadastrados. Ao selecionar , serão carregados os dados do animal e ele pode atualizar ou excluir conforme sua necessidade.



Área Administrador Cadastros Adoções Relatórios Formulários Adotantes Logout

VIRALATA

Viralata: Sistema de Adoção de Animais

Selecione uma Feira para atualizar seus dados

Feiras: New Feira

Dados da Feira

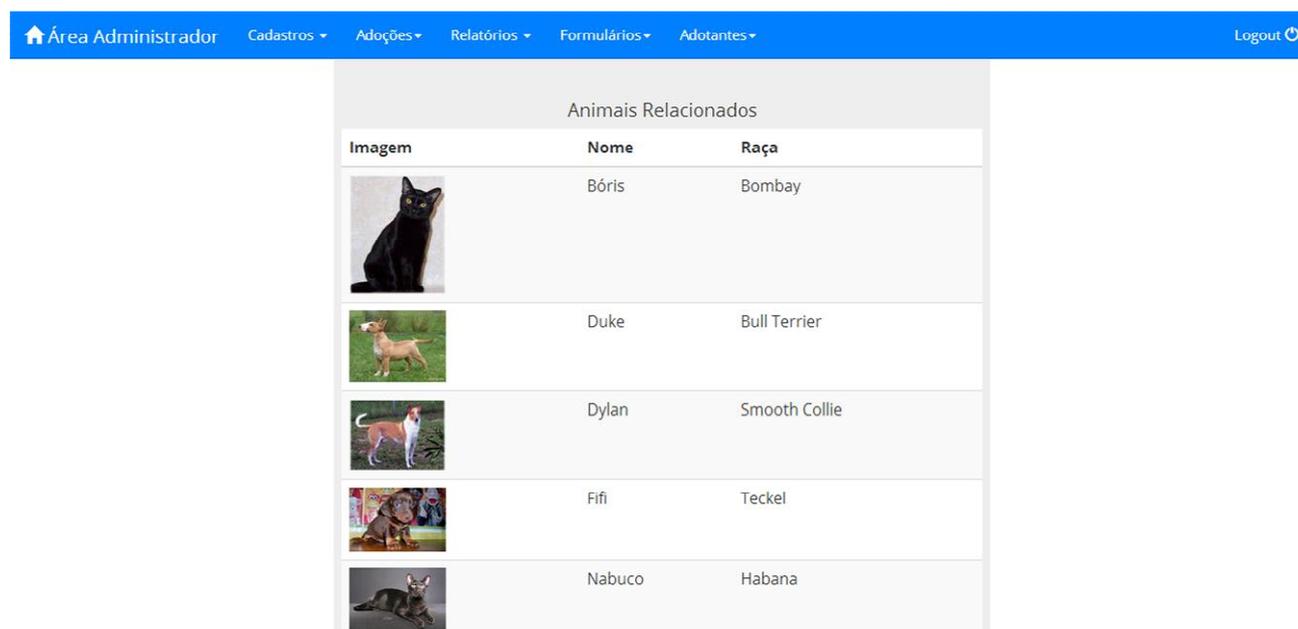
Nome: New Feira

Data Início: 05/07/2014

Horário Início: 19:20

Data: 06/07/2014

Figura 035. Manter feiras parte 1.



Área Administrador Cadastros Adoções Relatórios Formulários Adotantes Logout

Animais Relacionados

Imagem	Nome	Raça
	Bóris	Bombay
	Duke	Bull Terrier
	Dylan	Smooth Collie
	Fifi	Teckel
	Nabuco	Habana

Figura 036. Manter feiras parte 2.

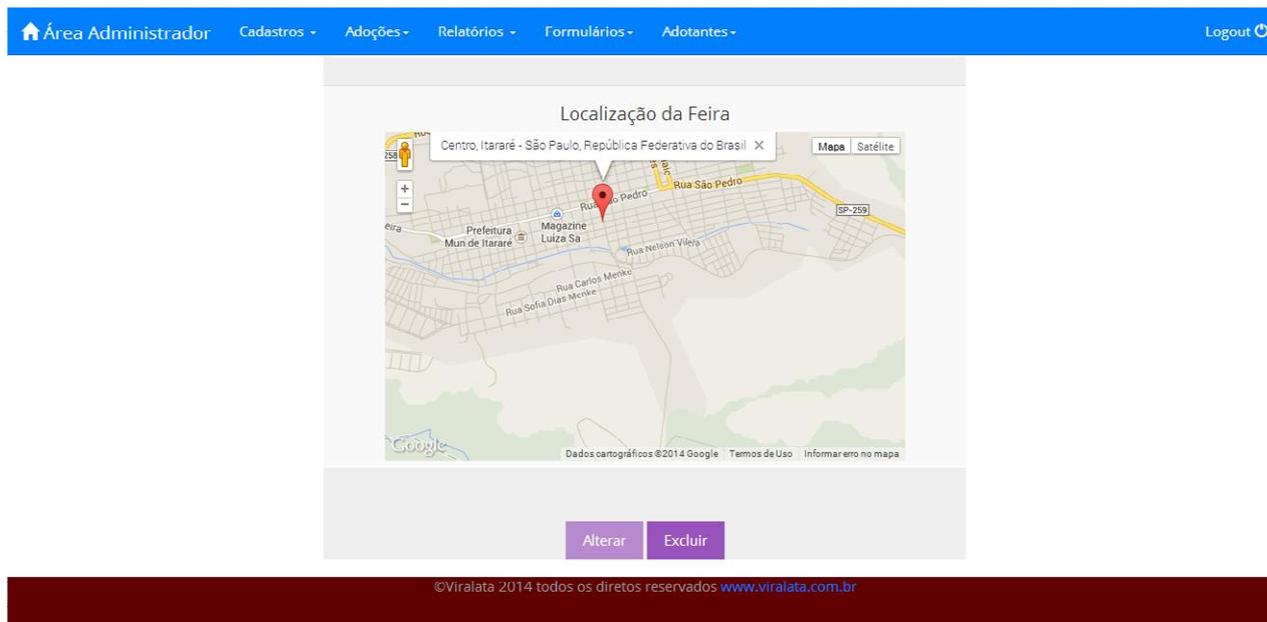


Figura 037. Manter feira parte 3.

Depois que uma feira for cadastrada, um administrador pode atualizar ou excluí-la conforme as figuras acima.



Figura 038. Relatório de abandono de animais.

O relatório de abandono de animais é utilizado para o administrador visualizar dados estatísticos para saber as porcentagens de animais



abandonados por espécies.

Figura 039. Formato PDF de relatório de abandono.

Caso o administrador clicar no botão "exportar para PDF", será gerado um arquivo.pdf de mesmo modelo conforme a figura 039.

Área Administrador Cadastros ▾ Adoções ▾ Relatórios ▾ Formulários ▾ Adotantes ▾ Logout

VIRALATA

Viralata: Sistema de Adoção de Animais

Intervalo máximo: 6 meses

Data Inicial:

Data Final:

Gerar Relatório

Data Adoção	Nome Adotante	Nome Animal	Espécie	Raça
07/07/2014 21:23	Cleiton Silva	Duke	Cachorro	Bull Terrier

Exportar para PDF

©Viralata 2014 todos os direitos reservados www.viralata.com.br

Figura 040. Relatório de adoções.

Para o administrador de sistema gerar um relatório sobre dados de adotantes que adotaram animais utilizando intervalo de tempo/data, basta selecionar “opção” e preencher os campos de data inicial e data final. Também pode ser exportado o relatório gerado no formato PDF, como mostra a figura 040.



Relatório de Adoções

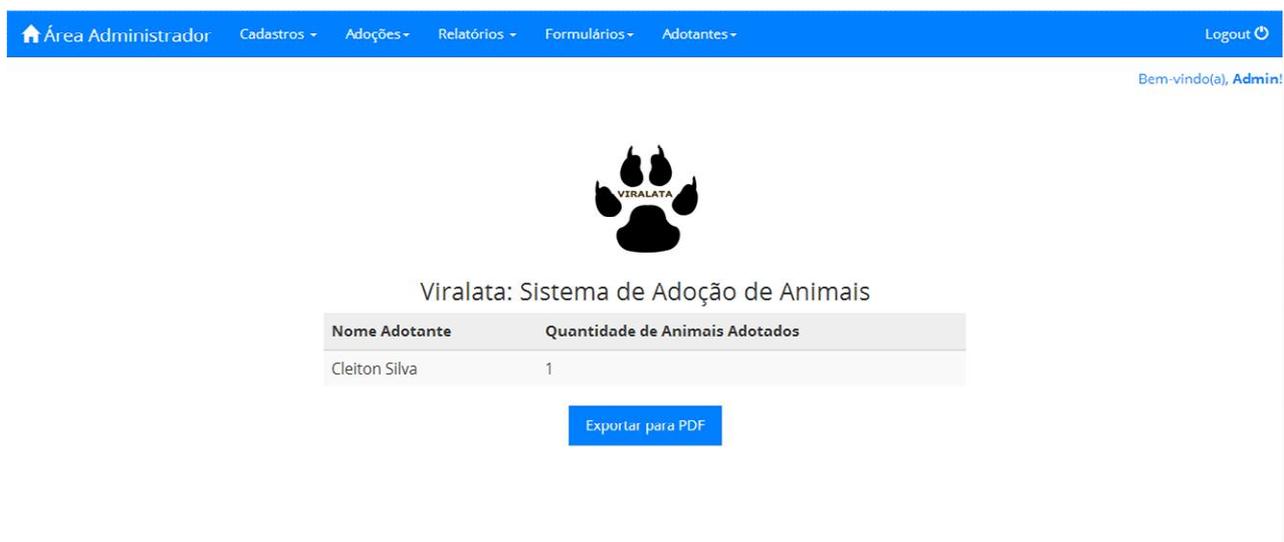


Período: 01/07/2014 a 08/07/2014

Data Adoção	Nome Adotante	Nome Animal	Espécie	Raça
08/07/14 22:53	Mirosmar Camargo	Thor	Cachorro	Pug
08/07/14 22:53	Cleiton Silva	Bartolomeu	Gato	California spangled
07/07/14 21:23	Cleiton Silva	Duke	Cachorro	Bull Terrier

Figura 041. Formato PDF de relatório de adoções.

Um arquivo no formato PDF pode ser exportado para imprimir relatório de adoções feitas dentro de um intervalo de determinado tempo, como mostra a figura 041 acima.



Área Administrador Cadastros Adoções Relatórios Formulários Adotantes Logout

Bem-vindo(a), Admin!

VIRALATA

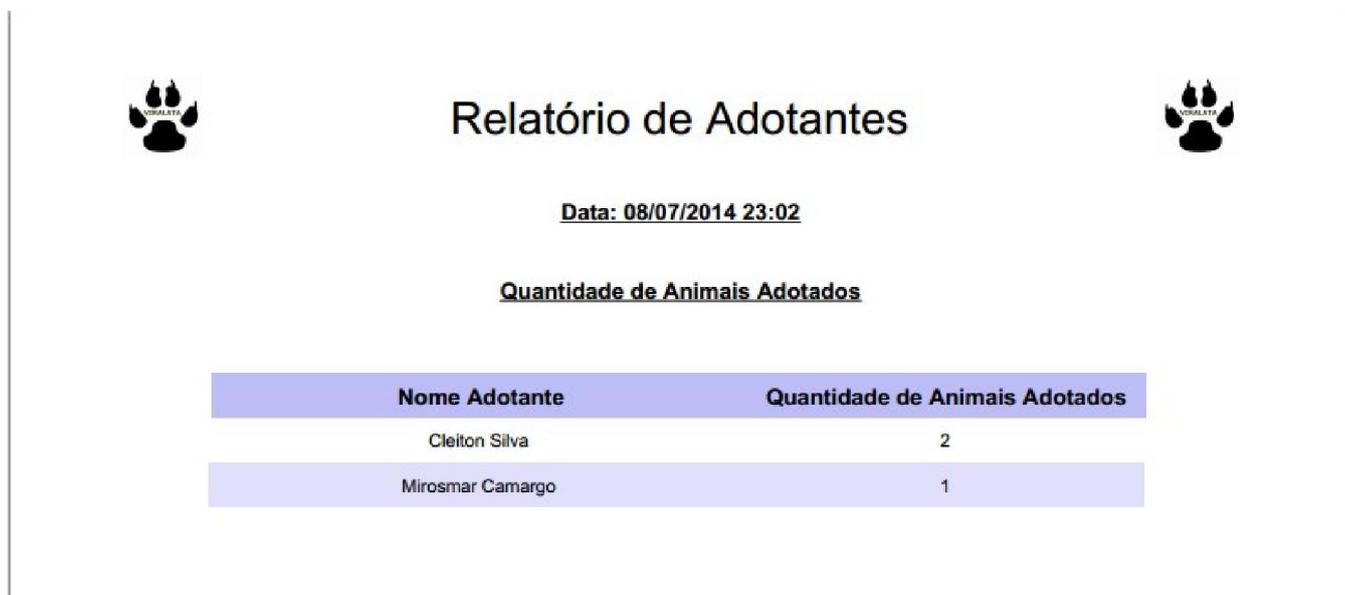
Viralata: Sistema de Adoção de Animais

Nome Adotante	Quantidade de Animais Adotados
Cleiton Silva	1

Exportar para PDF

Figura 042. Relatório de adotantes.

O sistema viralata consta com módulo de relatório de adotantes. Funciona para o administrador saber quais são os adotantes que mais adotam, no caso de tomada de algumas decisões. Pode ser gerado/exportado no formato PDF também.



VIRALATA

Relatório de Adotantes

VIRALATA

Data: 08/07/2014 23:02

Quantidade de Animais Adotados

Nome Adotante	Quantidade de Animais Adotados
Cleiton Silva	2
Mirosmar Camargo	1

Figura 043. Formato PDF de relatório de adotantes.

A figura acima mostra arquivo.pdf depois que o administrador clica no botão exportar para PDF.

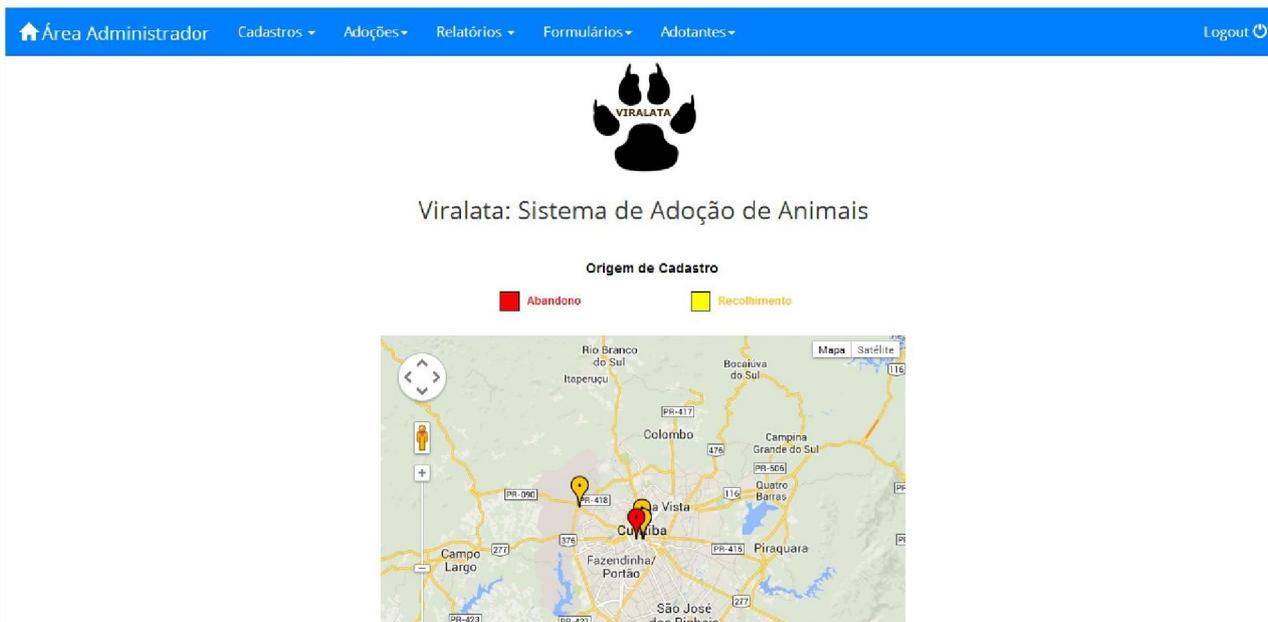


Figura 044. Relatório Geoanalítico.

Para um administrador saber onde há maior número de abandono e recolhimento de animais, é gerado um relatório geoanalítico utilizando um mapa, que exibirá em vários pontos, facilitando para administrador. A figura 044 acima mostra a tela.

6. CONSIDERAÇÕES FINAIS

Primeiramente, gostaríamos de enfatizar que o processo de desenvolvimento do projeto foi de caráter enriquecedor para o conhecimento dos envolvidos. Com ele, foi possível aprender diversas coisas, desde conhecimentos na área de programação quanto de gerenciamento de projetos, através da pesquisa e execução do trabalho.

6.1 CONCLUSÃO

Ao longo do desenvolvimento do sistema, a equipe de desenvolvedores identificou diversos erros e falhas tanto no sistema quanto na concepção da ideia inicial do projeto, bem como no processo de elaboração do mesmo. Através de depurações no código fonte e conversas com o cliente, pôde-se chegar a um resultado satisfatório para ambas as partes. Devido ao fato de o processo de adoção de animais ter alguns diferenciais, foi necessário o empenho dos integrantes em entender todo o processo mencionado, o que, para nós, serviu como um exercício de análise de requisitos e de negócio.

Quanto ao desempenho do sistema, concluímos juntamente ao cliente, que ele cumpre com o especificado durante o início dos trabalhos. Através de pequenos testes foi possível aperfeiçoar e adequar as funcionalidades que nos foram solicitadas; além disso, o *feedback* do cliente foi fundamental para o sucesso do trabalho.

Como mencionado anteriormente, o resultado alcançado foi, a princípio, satisfatório para ambas as partes. Como o projeto elaborado é destinado à implementação e uso em grande escala por diversas pessoas, nos

resta saber se o mesmo satisfará também os futuros usuários e beneficiados pelo sistema.

REFERÊNCIAS BIBLIOGRÁFICAS

ROGER S. PRESSMAN. **Engenharia de Software**, Terceira Edição, Pearson Makron Books, São Paulo, 2005.

GRADY BOOCH, JAMES RUNBARG, IVAR JACOBSON. **UML guia do usuário**, Segunda Edição, Elsevier Editora Ltda, Rio de Janeiro, 2006.

GUILLEANES T.A. G. **UML 2 Guia prático**, Novatec Editora, 2007.

SIERRA, KATHY. **Use a cabeça Java**, Guia de Aprendizagem, Segunda Edição, Alta Books, 2005.

BRAYAN BASHAM, KATHY SIERRA, BERT BATES. **Head First servlets and JSP**, Second Edition, O'Reilly, 2008.

LEI Nº 13.914 - Sobre comércio de animais em Curitiba - como denunciar. Disponível em: <<http://www.adotebicho.com.br/artigos/45-adocao-animais/183-leicomerciocuritiba>>. Acesso em: 28/03/2014.

ADOÇÃO (CF, Art. 227, §§ 5º e 6º, Lei 8.069 de 13.7.90, Arts. 39 a 52). Disponível em: <<http://www.dji.com.br/civil/adocao.htm>>. Acesso em: 28/03/2014

ADOÇÃO À LUZ DO CÓDIGO CIVIL de 1916. Disponível em: <http://www.ambito-juridico.com.br/site/index.php?n_link=revista_artigos_leitura&artigo_id=9266>. Acesso em: 28/03/2014

HISTÓRIA DA ADOÇÃO no mundo. Disponível em: <<http://www.senado.gov.br/noticias/Jornal/emdiscussao/adocao/contexto-da-adocao-no-brasil/historia-da-adocao-no-mundo.aspx>>. Acesso em 01/04/2014

LEI Nº 4.655 - de 2 de junho de 1965, DOU de 15/06/1965 – Revogada.
Disponível em:
<<http://www010.dataprev.gov.br/sislex/paginas/42/1973/..%5C1965%5C4655.htm>>. Acesso em 02/04/2014

OBJECT MANAGEMENT GROUP(OMG): Recurso por UML. Disponível em:
<<http://www.omg.org/spec/UML/>>. Acesso em 06/07/2014

APACHE MAVEN PROJECT, Maven em 5 minutos. Disponível em:
<<http://maven.apache.org/guides/getting-started/maven-in-five-minutes.html>> . Acesso em 06/07/2014

CRITICALS TOOLS, Project Planning and Graphing Software, for WBS.
Disponível em: <<http://www.criticaltools.com/wbschartprosoftware.htm>>.
Acesso em 06/07/2014

APÊNDICE A

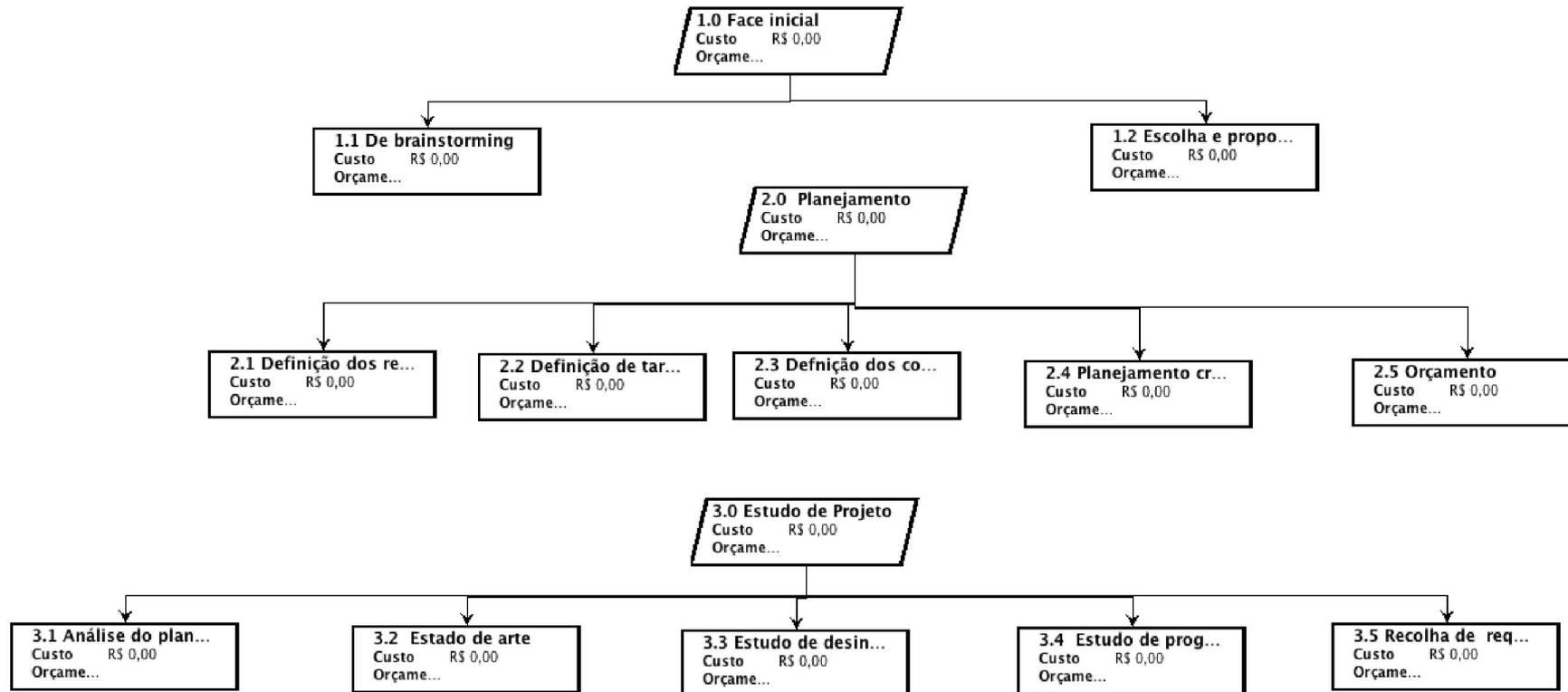


DIAGRAMA WBS PARTE 1

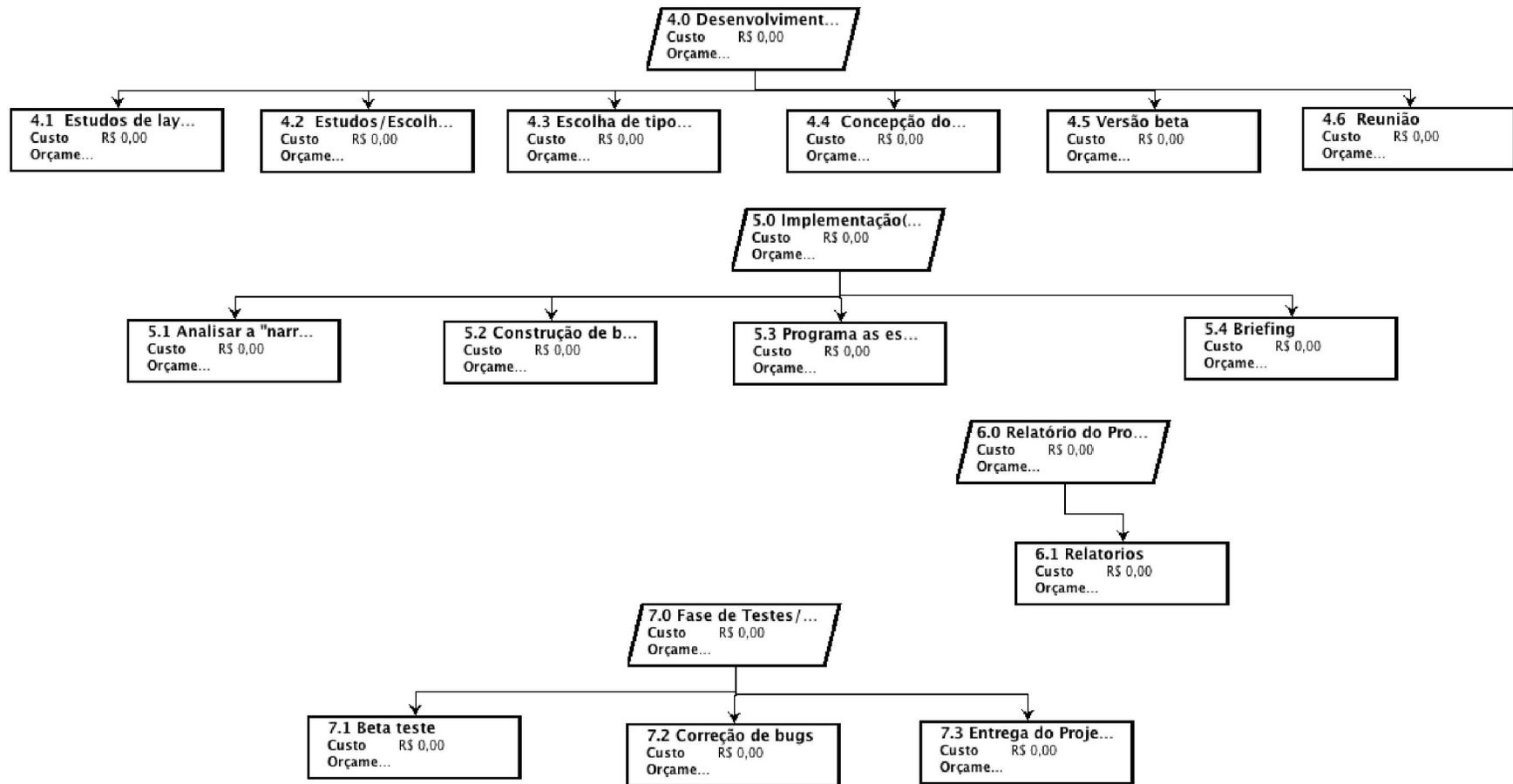


DIAGRAMA WBS PARTE 2

APÊNDICE B

		Nome	Duração	Início	Término	Predecessoras	Nome do Recurso
1		1.0 Face inicial	11 dias?	24/03/14 08:00	07/04/14 17:00		Habibu;Rodrigo;Marcelo
2		1.1 De brainstorming	5 dias?	24/03/14 08:00	28/03/14 17:00		Rodrigo
3		1.2 Escolha e proposta do conceito do projeto a ser desenvolver	6 dias?	31/03/14 08:00	07/04/14 17:00	2	Rodrigo
4		2.0 Planejamento	10,5 dias?	22/04/14 08:00	06/05/14 13:00		Habibu;Rodrigo;Marcelo
5		2.1 Definição dos requisitos funcionais	2,5 dias?	22/04/14 08:00	24/04/14 13:00		Habibu;Rodrigo
6		2.2 Definição de tarefas para membros da equip	1 dia?	24/04/14 13:00	25/04/14 13:00	5	
7		2.3 Defnição dos conteúdos a serem desenvolver	1 dia?	25/04/14 13:00	28/04/14 13:00	6	
8		2.4 Planejamento cronologico do projeto(Diagrama Gantt)	1 dia?	28/04/14 13:00	29/04/14 13:00	7	
9		2.5 Orçamento	5 dias?	29/04/14 13:00	06/05/14 13:00	8	Habibu;Rodrigo;Marcelo
10		3.0 Estudo de Projeto	22 dias?	28/04/14 08:00	27/05/14 17:00		Habibu;Rodrigo;Marcelo
11		3.1 Análise do planejamento	5 dias?	28/04/14 08:00	02/05/14 17:00		
12		3.2 Estado de arte	1 dia?	05/05/14 08:00	05/05/14 17:00	11	
13		3.3 Estudo de desing a ser desenvolver	1 dia?	06/05/14 08:00	06/05/14 17:00	12	
14		3.4 Estudo de programação a aplicar	10 dias?	07/05/14 08:00	20/05/14 17:00	13	
15		3.5 Recolha de requisistos(contéudo)	5 dias?	21/05/14 08:00	27/05/14 17:00	14	Habibu;Rodrigo;Marcelo
16		4.0 Desenvolvimento do Design	25 dias?	22/05/14 08:00	25/06/14 17:00		Habibu;Rodrigo;Marcelo
17		4.1 Estudos de layout	3 dias?	22/05/14 08:00	26/05/14 17:00		
18		4.2 Estudos/Escolha de cores	1 dia?	27/05/14 08:00	27/05/14 17:00	17	
19		4.3 Escolha de tipografia	1 dia?	28/05/14 08:00	28/05/14 17:00	18	
20		4.4 Concepção do Design	5 dias?	28/05/14 08:00	03/06/14 17:00	18	
21		4.5 Versão beta	20 dias?	29/05/14 08:00	25/06/14 17:00	19	
22		4.6 Reunião	5 dias?	04/06/14 08:00	10/06/14 17:00	20	
23		5.0 Implementação(Programação)	37 dias?	28/05/14 08:00	17/07/14 17:00		Habibu;Rodrigo;Marcelo
24		5.1 Analisar a "narrativa" projetual	3 dias?	28/05/14 08:00	30/05/14 17:00		
25		5.2 Construção de base de dados necessarios	5 dias?	29/05/14 08:00	04/06/14 17:00	19	
26		5.3 Programa as especificações funcionais	30 dias?	05/06/14 08:00	16/07/14 17:00	25	
27		5.4 Briefing	1 dia?	17/07/14 08:00	17/07/14 17:00	26	
28		6.0 Relatório do Projeto	1 dia?	17/07/14 08:00	17/07/14 17:00	26	Habibu
29		6.1 Relatorios	1 dia?	17/07/14 08:00	17/07/14 17:00		
30		7.0 Fase de Testes /avaliação de projeto	7 dias?	17/07/14 08:00	25/07/14 17:00		Marcelo;Rodrigo
31		7.1 Beta teste	1 dia?	17/07/14 08:00	17/07/14 17:00	26	
32		7.2 Correção de bugs	5 dias?	18/07/14 08:00	24/07/14 17:00	31	
33		7.3 Entrega do Projeto	1 dia?	25/07/14 08:00	25/07/14 17:00	32	

DIAGRAMA DE GANTT PARTE 1

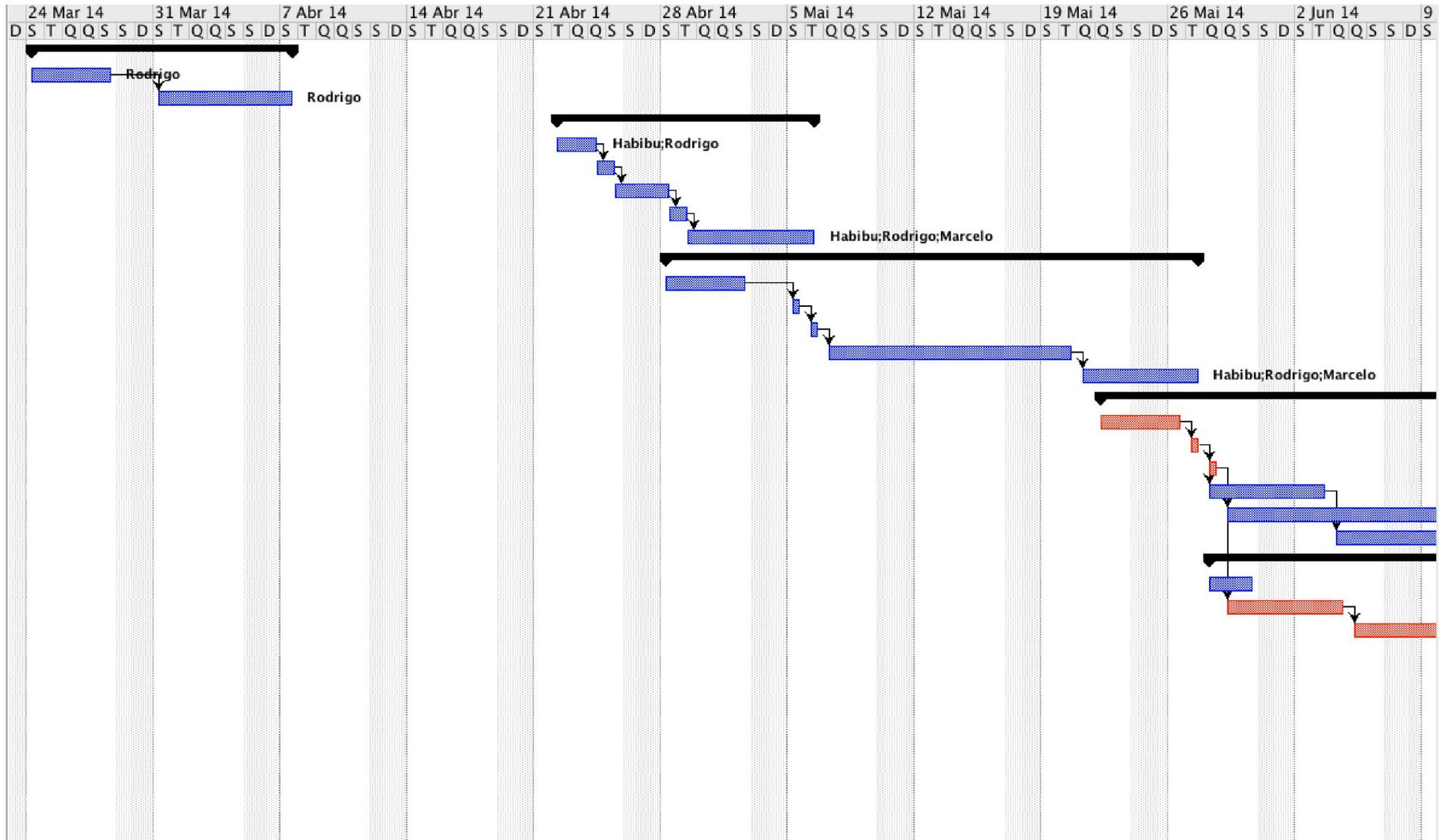


DIAGRAMA DE GANTT PARTE 2

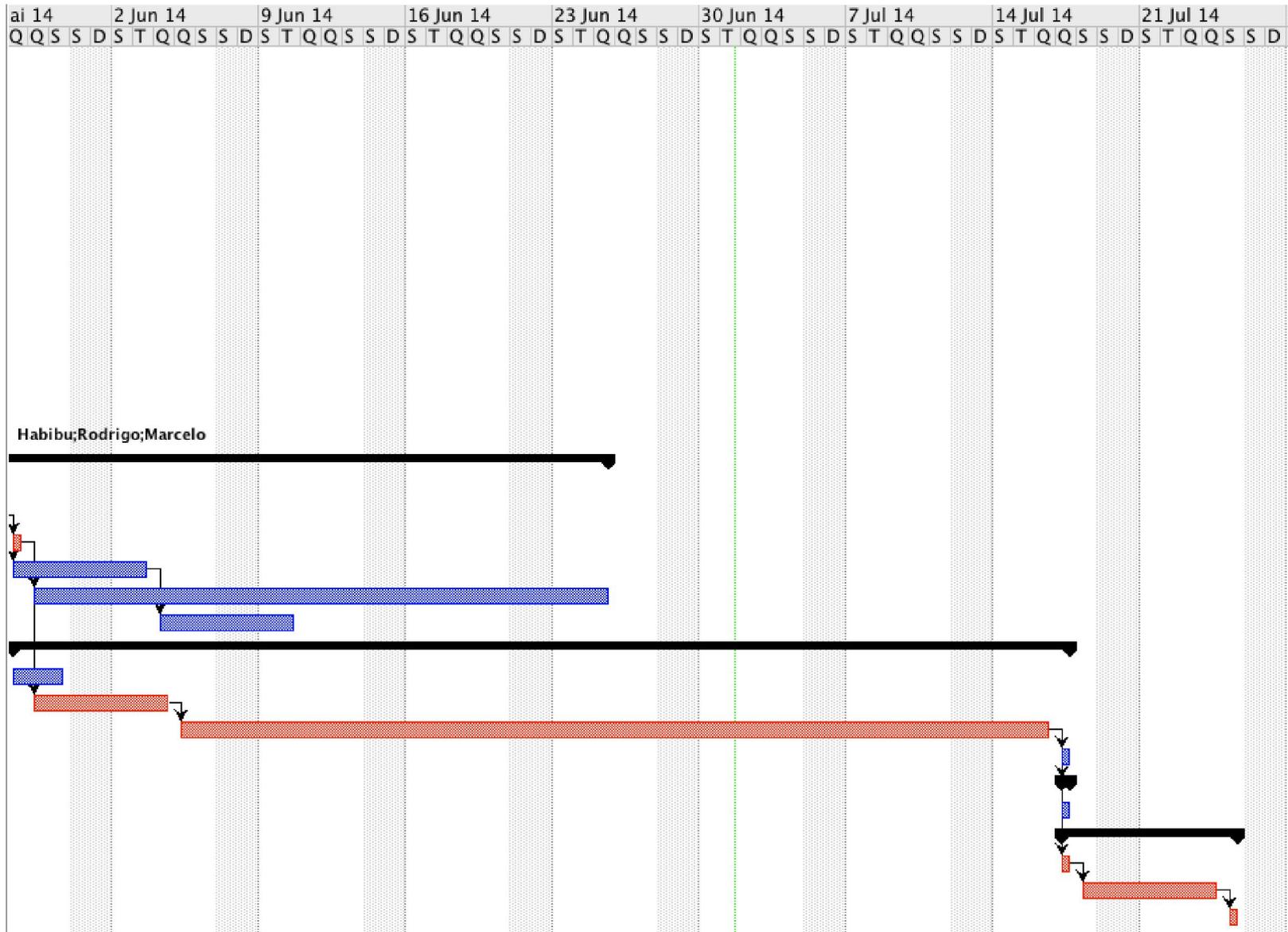


DIAGRAMA DE GANTT PARTE 3

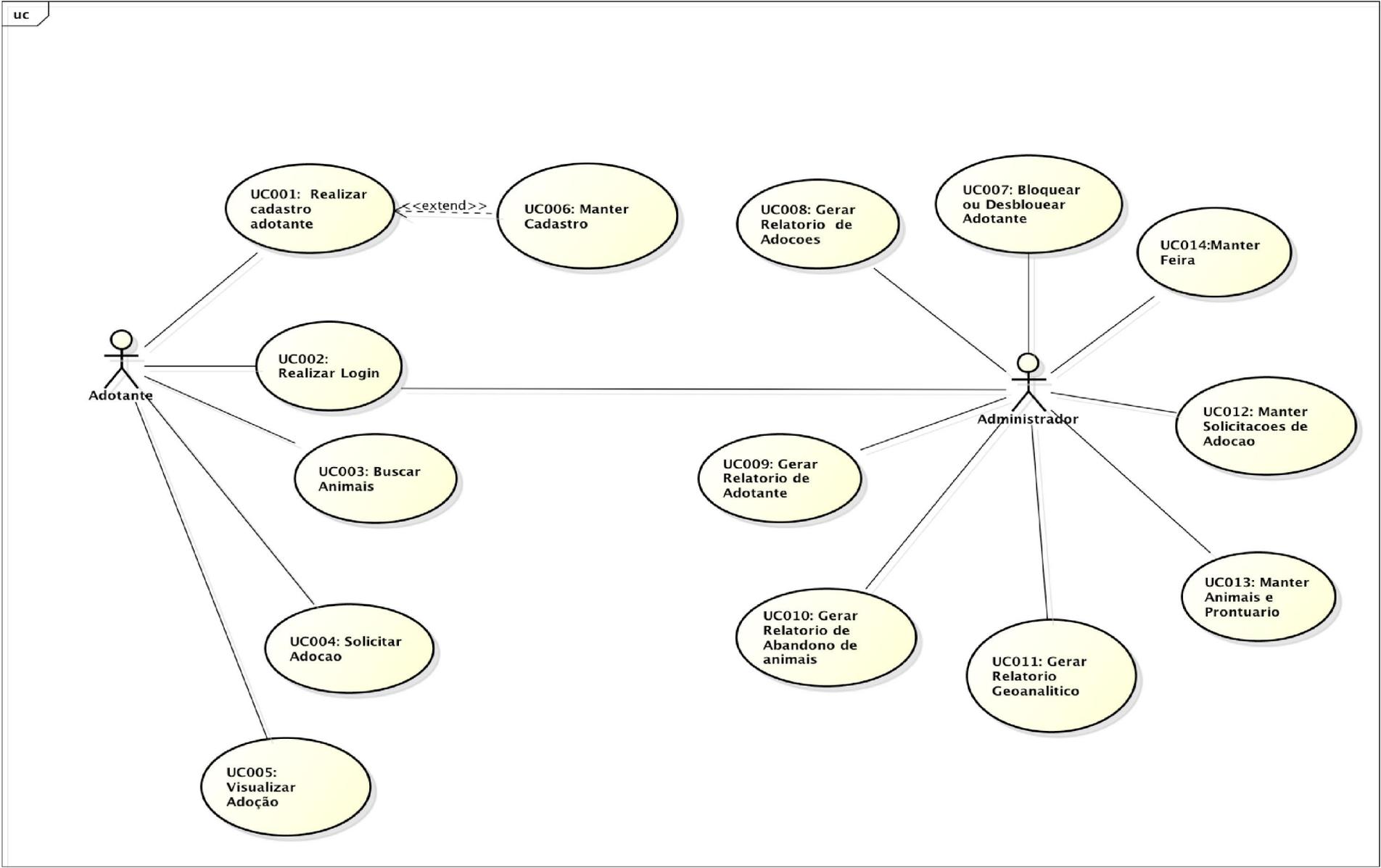


Diagrama de Caso de Uso

APÊNDICE D

Especificação de Caso de Uso

UC001 – Realizar Cadastro do Adotante

Descrição

Este caso de uso serve para o Adotante realizar seu cadastro no sistema.

Data View

DV1 - Tela Principal

DV2 - Dados Cadastrais

DV3 - Seleção do Tipo de Pessoa

Pré-condições

Não há.

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Realizar Cadastro do Adotante

Ator Primário

Adotante

Fluxo de Eventos Principal

1. O Sistema apresenta a tela “Tela Principal” com os campos “Usuário” e “Senha”. **(DV1)**
2. O Adotante seleciona a opção “Realizar Cadastro”.
3. O Sistema apresenta a tela “Seleção do tipo de Pessoa”. **(DV3)**
4. O Sistema apresenta a tela “Dados Cadastrais” com os campos para Cadastro do Adotante. **(DV2)**

5. O Adotante preenche os campos de cadastro e clica no botão “Cadastrar”. **(E1)**
6. O Sistema apresenta mensagem de confirmação de cadastro “Deseja confirmar seu cadastro?” com os botões “Sim/Cancelar”. **(E1)**
7. O Adotante seleciona a opção “Sim”. **(A1)**
8. O Sistema salva o cadastro do Adotante e apresenta mensagem “Cadastro realizado com sucesso!”. **(DV1)(E2)(E3)(E4)(R1)(R2)**
9. O caso de uso é encerrado.

Fluxos Alternativos

A1. Adotante seleciona a opção “Cancelar”.

1. O Sistema retorna para o passo 3 do fluxo principal.

Fluxos de Exceção

E1. Adotante não preenche campos obrigatórios

1. O Sistema apresenta a mensagem “Campo <Nome do Campo> não preenchido.” e retorna para o passo de origem.

E2. Erro de rede:

1. O Sistema apresenta a mensagem: “Erro. Por favor, tente novamente.”.
2. O Sistema retorna ao passo de chamada.

E3. Login já existente no sistema:

1. O Sistema apresenta a mensagem: “Login já existente. Por favor, informe um novo login.”.
2. O Sistema retorna ao passo de chamada.

E4. Cpf/Cnpj já existente no sistema:

1. O Sistema apresenta a mensagem: “Usuário já existente no sistema.”.

2. O Sistema retorna ao passo de chamada.

Regras de Negócio

R1. O Sistema deve enviar *e-mail* de confirmação de cadastro realizado com sucesso para o Adotante.

R2. O Sistema não deve permitir o cadastro de usuários que já possuem *login* ou cpf/cnpj cadastrados.

Especificação de Caso de Uso

UC002 – Realizar Login

Descrição

Este caso de uso serve para o Usuário realizar *login*.

Data View

DV1 – Tela Principal

DV2 – Área do Adotante

DV3 – Área do Administrador

Pré-condições

1. Não há.

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Realizar *login* do Usuário.

Ator Primário

Adotante

Fluxo de Eventos Principal

1. O Sistema apresenta a tela “Tela Principal” com os campos “Usuário” e “Senha”. **(DV1)**

2. O Adotante preenche os campos “Usuário e Senha” e clica no botão “Login”. **(E1)(E3)**
3. O Sistema realiza o login e apresenta a tela “Área do Adotante”. **(DV2)**
(A1)(E2)
4. O caso de uso é encerrado.

Fluxos Alternativos

A1. *Login* realizado pelo administrador.

1. O Sistema realiza o *login* e apresenta a tela “Área do Administrador”.
(DV3)
2. O caso de uso é encerrado.

Fluxos de Exceção

E1. Adotante não preenche campos obrigatórios

1. O Sistema apresenta a mensagem “Campo <Nome do Campo> não preenchido.” e retorna para o passo de origem.

E2. Erro de rede:

1. O Sistema apresenta a mensagem: “Erro. Por favor, tente novamente.”.
2. O Sistema retorna ao passo de chamada.

E3. Dados Incorretos

1. O Sistema apresenta a mensagem: “Dados incorretos. Verifique seu *login* e senha”.
2. O Sistema retorna ao passo de chamada.

Regras de Negócio

Não há.

Especificação de Caso de Uso

UC003 – Buscar Animais

Descrição

Este caso de uso serve para o Adotante buscar animais para adoção.

Data View

DV1 – Área do Adotante

DV2 – Busca de Animais

Pré-condições

1. O Adotante deve ter realizado o *login*.

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Buscar animais.

Ator Primário

Adotante

Fluxo de Eventos Principal

1. O Sistema apresenta a tela “Área do Adotante”. **(DV1)**
2. O Adotante seleciona a opção “Buscar Animais”.
3. O Sistema apresenta a tela “Busca de Animais”. **(DV2) (E2)**
4. O Adotante preenche os campos de pesquisa e clica em “Pesquisar”. **(E3)**
5. O Sistema apresenta na tela os animais disponíveis para adoção. **(DV2)(E1)(E2)(R1)**
6. O Adotante visualiza as informações e clica em voltar. **(A1)**
7. O caso de uso é encerrado.

Fluxos Alternativos

A1. Adotante seleciona a opção “Solicitar Adoção”:

1. O Sistema estende o caso de uso “UC004 – Solicitar Adoção”.

Fluxos de Exceção

E1. Busca sem Resultados:

1. O Sistema apresenta a mensagem “No momento não há animais disponíveis para adoção com as características localizadas” e retorna para o passo 3 do Fluxo Principal.

E2. Erro de rede:

1. O Sistema apresenta a mensagem: “Erro. Por favor, tente novamente.”.
2. O Sistema retorna ao passo de chamada.

E3. Campo de busca não-preenchido:

1. O Sistema apresenta a mensagem “Verifique o preenchimento do campo 'Busca'” e retorna para o passo 3 do Fluxo Principal.

Regras de Negócio

R1. Não serão apresentados animais com status “Bloqueado” ou “Concluído”.

R2. O status do animal solicitado à adoção deve ser alterado para “Em processo de adoção”.

Especificação de Caso de Uso

UC004 – Solicitar Adoção

Descrição

Este caso de uso serve para o Adotante solicitar adoção de animais.

Data View

DV1 – Área do Adotante

DV2 – Busca de Animais

Pré-condições

1. O Adotante deve ter executado o caso de uso “UC003 - Buscar Animais”.

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Solicitar adoção de animais.

Ator Primário

Adotante

Fluxo de Eventos Principal

1. O Sistema apresenta um alerta de confirmação “Deseja confirmar a Solicitação de Adoção deste animal?”.
2. O Adotante seleciona a opção “Sim.” **(A1)**
3. O Sistema envia a solicitação para o agente responsável e apresenta a mensagem “Solicitação realizada com sucesso!”. **(E1)(E2)(R1)(R2)(R3)**
4. O caso de uso é encerrado.

Fluxos Alternativos

A1. Adotante seleciona a opção “Não”:

1. O Sistema retorna ao passo de chamada.

Fluxos de Exceção

E1. Erro de rede:

1. O Sistema apresenta a mensagem: “Erro. Por favor, tente novamente.”.
2. O Sistema retorna ao passo de chamada.

E2. Adotante com cadastro bloqueado:

1. O Sistema apresenta a mensagem: “Não foi possível completar sua solicitação. Por favor, entre em contato com os nossos agentes”.

2. O caso de uso é encerrado.

Regras de Negócio

R1. O status do animal solicitado à adoção deve ser alterado para “Em processo de adoção”.

R2. Não poderão ser solicitadas adoções à adotantes com o cadastro bloqueado.

R3. O Sistema deverá enviar *e-mail* de confirmação de solicitação de adoção para o Adotante.

Especificação de Caso de Uso

UC005 – Visualizar Adoções

Descrição

Este caso de uso serve para o Adotante visualizar solicitações de adoção de animais.

Data View

DV1 – Área do Adotante

DV2 – Minhas Adoções

Pré-condições

N/A

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Apresentar lista de solicitações de adoção de animais.

Ator Primário

Adotante

Fluxo de Eventos Principal

1. O Sistema apresenta a tela “Área do Adotante”. **(DV1)**
2. O Adotante seleciona a opção “Minhas Adoções”.
3. O Sistema apresenta a tela “Minhas Adoções” com a listagem de solicitações realizadas e seus status. **(E1)(DV2)**
4. O caso de uso é encerrado.

Fluxos Alternativos

N/A.

Fluxos de Exceção

E1. Erro de rede:

1. O Sistema apresenta a mensagem: “Erro. Por favor, tente novamente.”.
2. O Sistema retorna ao passo de chamada.

Regras de Negócio

N/A.

Especificação de Caso de Uso

UC006 – Manter Cadastro

Descrição

Este caso de uso serve para o Usuário manter o cadastro de Adotante.

Data View

DV1 - Dados Cadastrais

DV2 – Área do Adotante

DV3 – Área do Administrador

DV4 - Administração de Adotantes

Pré-condições

1. Não há.

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Manter cadastro.

Ator Primário

Adotante

Fluxo de Eventos Principal

1. O Sistema apresenta a tela “Área do Adotante”. **(DV2)**
2. O Adotante seleciona a opção “Perfil”.
3. O Sistema apresenta a tela “Dados Cadastrais” com os dados do Adotante logado. **(DV1) (E2)**
4. O Adotante altera os campos desejados e clica no botão “Alterar”. **(E1)**
5. O Sistema apresenta mensagem de confirmação de alteração “Deseja confirmar a alteração?” com os botões “Sim/Não”.
6. O Adotante seleciona a opção “Sim”. **(A1)**
7. O Sistema salva as alterações e apresenta mensagem “Alteração realizada com sucesso!”. **(DV1)(E1)(E2)**
8. O caso de uso é encerrado.

Fluxos Alternativos

A1. Adotante seleciona a opção “Não”.

1. O Sistema retorna para o passo de chamada.

Fluxos de Exceção

E1. Adotante não preenche campos obrigatórios

1. O Sistema apresenta a mensagem “Campo <Nome do Campo> não preenchido.” e retorna para o passo de origem.

E2. Erro de rede:

1. O Sistema apresenta a mensagem: “Erro. Por favor, tente novamente.”.
2. O Sistema retorna ao passo de chamada.

Regras de Negócio

N/A

Especificação de Caso de Uso**UC007 – Bloquear ou Desbloquear Adotantes****Descrição**

Este caso de uso serve para o usuário manter o cadastro de Adotante.

Data View

DV1 – Área do Administrador

DV2 - Administração de Adotantes

Pré-condições

1. Não há.

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Bloquear ou Desbloquear Adotante.

Ator Primário

Administrador

Fluxo de Eventos Principal

1. O Sistema apresenta a tela “Área do Administrador”. **(DV1)**
2. O Adotante seleciona a opção “Adotantes”.

3. O Sistema apresenta a tela “Administração de Adotantes” com campo para seleção de adotante. **(DV2) (E2)**
4. O Administrador seleciona o Adotante desejado.
5. O Sistema apresenta dados do Adotante selecionado com o Status do Adotante e o botão Bloquear. **(A1)(R1)**
6. O Administrador clica no botão “Bloquear”.
7. O Sistema efetua o bloqueio do Adotante e apresenta a mensagem “Adotante bloqueado com sucesso!”.
8. O caso de uso é encerrado.

Fluxos Alternativos

A1. Adotante previamente bloqueado

1. O Sistema apresenta dados do Adotante selecionado com o Status do Adotante e o botão Desbloquear. **(R1)**
2. O Administrador clica no botão “Desbloquear”.
3. O Sistema efetua o desbloqueio do Adotante e apresenta a mensagem “Adotante desbloqueado com sucesso!”.
4. O caso de uso é encerrado.

Fluxos de Exceção

E1. Adotante não preenche campos obrigatórios

1. O Sistema apresenta a mensagem “Campo <Nome do Campo> não preenchido.” e retorna para o passo de origem.

E2. Erro de rede:

1. O Sistema apresenta a mensagem: “Erro. Por favor, tente novamente.”.
2. O Sistema retorna ao passo de chamada.

Regras de Negócio

R1. O sistema deve apresentar o botão Bloquear se o status do usuário for desbloqueado ou apresentar o botão Desbloquear se o status do usuário for bloqueado.

Especificação de Caso de Uso**UC008 – Gerar Relatório de Adoções****Descrição**

Este caso de uso serve para o Administrador gerar relatório de adoções.

Data View

DV1 - Área do Administrador

DV2 - Relatório de Adoções

Pré-condições

1. O Administrador deve ter realizado o login.

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Gerar relatório de adoções.

Ator Primário

Administrador

Fluxo de Eventos Principal

1. O Sistema apresenta a tela “Área do Administrador”. (**DV1**).

2. O Administrador clica no menu “Relatórios” e seleciona a opção “Relatório de Adoções”.
3. O Sistema apresenta a tela “Relatório de Adoções” com os campos “Data Inicial” e “Data Final”. **(DV2)**
4. O Administrador preenche os campos e clica no botão “Gerar Relatório”.
(E1)(E2)(R1)
5. O Sistema apresenta o relatório na tela e o botão “Exportar para PDF”.
(E3)
6. O Administrador clica no botão “Exportar para PDF”.
7. O Sistema gera o relatório em formato PDF. **(E3)**
8. O Use Case é encerrado.

Fluxos Alternativos

N/A

Fluxos de Exceção

E1. Campos obrigatórios não preenchidos:

1. O Sistema verifica que um campo obrigatório não foi preenchido e apresenta a mensagem “Você precisa preencher o campo <Nome do Campo>”.
2. O Sistema retorna para o passo de origem.

E2. Período Inválido:

1. O Sistema verifica que o período selecionado é inválido e apresenta a mensagem “Período Inválido”.
2. O Sistema retorna para o passo de origem.

E3. Erro de rede:

1. O Sistema apresenta a mensagem: “Erro. Por favor, tente novamente.”

2. O Sistema retorna ao passo de chamada.

Regras de Negócio

R1. O período máximo para consulta é de 6 meses.

Especificação de Caso de Uso

UC009 – Gerar Relatório de Adotantes

Descrição

Este caso de uso serve para o Administrador gerar relatório de adotantes.

Data View

DV1 - Área do Administrador

DV2 - Relatório de Adotantes

Pré-condições

1. O Administrador deve ter realizado o *login*.

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Gerar relatório de adotantes.

Ator Primário

Administrador

Fluxo de Eventos Principal

1. O Sistema apresenta a tela “Área do Administrador”. **(DV1)**.
2. O Administrador clica no menu “Relatórios” e seleciona a opção “Relatório de Adotantes”.
3. O Sistema apresenta a tela “Relatório de Adotantes” . **(DV2)**
4. O Administrador preenche os campos e clica no botão “Gerar Relatório”.

5. O Sistema apresenta o relatório na tela e o botão “Exportar para PDF”.

(E1)

6. O Administrador clica no botão “Exportar para PDF”.

7. O Sistema gera o relatório em formato PDF. **(E1)**

8. O Use Case é encerrado.

Fluxos Alternativos

N/A

Fluxos de Exceção

E1. Erro de rede:

1. O Sistema apresenta a mensagem: “Erro. Por favor, tente novamente.”
2. O Sistema retorna ao passo de chamada.

Regras de Negócio

N/A

Especificação de Caso de Uso

UC010 – Gerar Relatório de Abandono

Descrição

Este caso de uso serve para o Administrador gerar relatório de abandono de animais.

Data View

DV1 - Área do Administrador

DV2 - Relatório de Abandono

Pré-condições

1. O Administrador deve ter realizado o *login*.

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Gerar relatório de abandono de animais..

Ator Primário

Administrador

Fluxo de Eventos Principal

1. O Sistema apresenta a tela “Área do Administrador”. **(DV1)**.
2. O Administrador clica no menu “Relatórios” e seleciona a opção “Relatório de Abandono”.
3. O Sistema apresenta a tela “Relatório de Abandono” . **(DV2)**
4. O Administrador preenche os campos e clica no botão “Gerar Relatório”.
5. O Sistema apresenta o relatório na tela e o botão “Exportar para PDF”.

(E1)

6. O Administrador clica no botão “Exportar para PDF”.
7. O Sistema gera o relatório em formato PDF. **(E1)**
8. O Use Case é encerrado.

Fluxos Alternativos

N/A

Fluxos de Exceção

E1. Erro de rede:

1. O Sistema apresenta a mensagem: “Erro. Por favor, tente novamente.”
2. O Sistema retorna ao passo de chamada.

Regras de Negócio

N/A

Especificação de Caso de Uso

UC011– Gerar Relatório Geoanalítico

Descrição

Este caso de uso serve para o Administrador gerar relatório geoanalítico.

Data View

DV1 - Área do Administrador

DV2 - Relatório Geoanalítico

Pré-condições

1. O Administrador deve ter realizado o *login*.

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Gerar relatório geoanalítico.

Ator Primário

Administrador

Fluxo de Eventos Principal

1. O Sistema apresenta a tela “Área do Administrador”. **(DV1)**.
2. O Administrador clica no menu “Relatórios” e seleciona a opção “Relatório Geonalítico”.
3. O Sistema apresenta a tela “Relatório Geonalítico” . **(DV2)**
4. O Administrador preenche os campos e clica no botão “Gerar Relatório”.
5. O Sistema apresenta o relatório na tela. **(E1)**
6. O Use Case é encerrado.

Fluxos Alternativos

N/A

Fluxos de Exceção

E1. Erro de rede:

1. O Sistema apresenta a mensagem: “Erro. Por favor, tente novamente.”
2. O Sistema retorna ao passo de chamada.

Regras de Negócio

N/A

Especificação de Caso de Uso

UC012 – Manter Solicitações de Adoção

Descrição

Este caso de uso serve para o Administrador manter solicitações de adoção.

Data View

DV1 - Área do Administrador

DV2 – Administração de Solicitações

DV3 – Detalhes da Solicitação de Adoção

Pré-condições

1. O Administrador deve ter realizado o login.

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Manter solicitações de adoção.

Ator Primário

Administrador

Fluxo de Eventos Principal

1. O Sistema apresenta a tela “Área do Administrador”. **(DV1)**.

2. O Administrador seleciona a opção “Solicitações de Adoções” no menu “Adoções”.
3. O Sistema apresenta a tela “Administração de Solicitações” com os botões “Aprovar e Negar”. **(DV2)**
4. O Administrador analisa a solicitação e seleciona a decisão desejada (Aprovar ou Negar). **(A2)**
5. O Sistema apresenta na tela campo para preenchimento de comentários sobre a decisão e o botão “Confirmar Decisão”. **(DV3)**
6. O Administrador clica no botão “Confirmar Decisão”. **(A2)**
7. O Sistema altera o status da solicitação no sistema e apresenta a mensagem “Solicitação finalizada com sucesso!”.
8. O Sistema envia e-mail com comprovante de solicitação aprovada para o Adotante informando sobre a decisão da solicitação e retorna para a tela “Área do Administrador”. **(A1)(DV1)**
9. O caso de uso é encerrado.

Fluxos Alternativos

A1. Solicitação de Adoção negada.

1. O Sistema envia *e-mail* para o Adotante informando sobre a negativa da solicitação de adoção.
2. O caso de uso é encerrado.

A2. Administrador clica no botão “Voltar”.

1. O caso de uso é encerrado.

Fluxos de Exceção

E1. Erro de rede:

1. O Sistema apresenta a mensagem: “Erro. Por favor, tente novamente.”

2. O Sistema retorna ao passo de chamada.

Regras de Negócio

N/A

Especificação de Caso de Uso

UC013 – Manter Animais

Descrição

Este caso de uso serve para o Administrador manter animais.

Data View

DV1 - Área do Administrador

DV2 - Administração de Animais

Pré-condições

1. O Administrador deve ter realizado o login.

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Manter animais.

Ator Primário

Administrador

Fluxo de Eventos Principal

1. O Sistema apresenta a tela “Área do Administrador”. **(DV1)**
2. O Administrador clica no menu “Cadastros” seleciona a opção “Manter Animais”.
3. O Sistema apresenta a tela “Administração de Animais”. **(DV2)**
4. O Sistema apresenta os dados do animal selecionado. **(DV2)**

5. O Administrador altera os campos desejados e clica no botão “Alterar”.
(A1)(E2)
6. O Sistema apresenta mensagem de confirmação de alteração “Deseja confirmar a alteração?” com os botões “Sim/Não”.
7. O Administrador seleciona a opção “Sim”. **(A4)**
8. O Sistema salva as alterações e apresenta mensagem “Animal alterado com sucesso!”. **(DV2) (E1)**
9. O caso de uso é encerrado.

Fluxos Alternativos

A1. Administrador seleciona a opção “Excluir”:

1. O Sistema apresenta mensagem de confirmação: “Deseja realmente excluir este animal?”.
2. O Administrador clica na opção “Sim”. **(A2)**
3. O Sistema realiza a exclusão do animal e apresenta mensagem “Animal excluído com sucesso!”. **(E1) (DV2)**
4. O caso de uso é encerrado.

A2. Adotante seleciona a opção “Não”.

1. O Sistema retorna para o passo de chamada.

Fluxos de Exceção

E1. Erro de rede:

1. O Sistema apresenta a mensagem: “Erro. Por favor, tente novamente.”.
2. O Sistema retorna ao passo de chamada.

E2. Administrador não preenche campos obrigatórios

1. O Sistema apresenta a mensagem “Campo <Nome do Campo> não preenchido.” e retorna para o passo de origem.

Regras de Negócio

Não há.

Especificação de Caso de Uso

UC014 – Manter Feiras

Descrição

Este caso de uso serve para o Administrador manter feiras.

Data View

DV1 - Área do Administrador

DV2 - Administração de Feiras

Pré-condições

1. O Administrador deve ter realizado o login.

Pós-condições

Após o fim normal deste caso de uso o sistema deve:

1. Manter feiras.

Ator Primário

Administrador

Fluxo de Eventos Principal

1. O Sistema apresenta a tela “Área do Administrador”. **(DV1)**
2. O Administrador clica no menu “Cadastros” seleciona a opção “Manter Feiras”.

3. O Sistema apresenta a tela “Administração de Feiras”. **(DV2)**
4. O Sistema apresenta os dados da feira selecionada. **(DV2)**
5. O Administrador altera os campos desejados e clica no botão “Alterar”.
(A1)(E2)
6. O Sistema apresenta mensagem de confirmação de alteração “Deseja confirmar a alteração?” com os botões “Sim/Não”.
7. O Administrador seleciona a opção “Sim”. **(A2)**
8. O Sistema salva as alterações e apresenta mensagem “Alteração realizada com sucesso!”. **(DV2) (E1)**
9. O caso de uso é encerrado.

Fluxos Alternativos

A1. Administrador seleciona a opção “Excluir”:

1. O Sistema apresenta mensagem de confirmação: “Deseja realmente excluir este feira?”.
2. O Administrador clica na opção “Sim”. **(A2)**
3. O Sistema realiza a exclusão do feira e apresenta mensagem “Exclusão realizada com sucesso!”. **(E1) (DV2)**
4. O caso de uso é encerrado.

A2. Adotante seleciona a opção “Não”.

1. O Sistema retorna para o passo de chamada.

Fluxos de Exceção

E1. Erro de rede:

1. O Sistema apresenta a mensagem: “Erro. Por favor, tente novamente.”.
2. O Sistema retorna ao passo de chamada.

E2. Administrador não preenche campos obrigatórios

1. O Sistema apresenta a mensagem “Campo <Nome do Campo> não preenchido.” e retorna para o passo de origem.

Regras de Negócio

Não há.

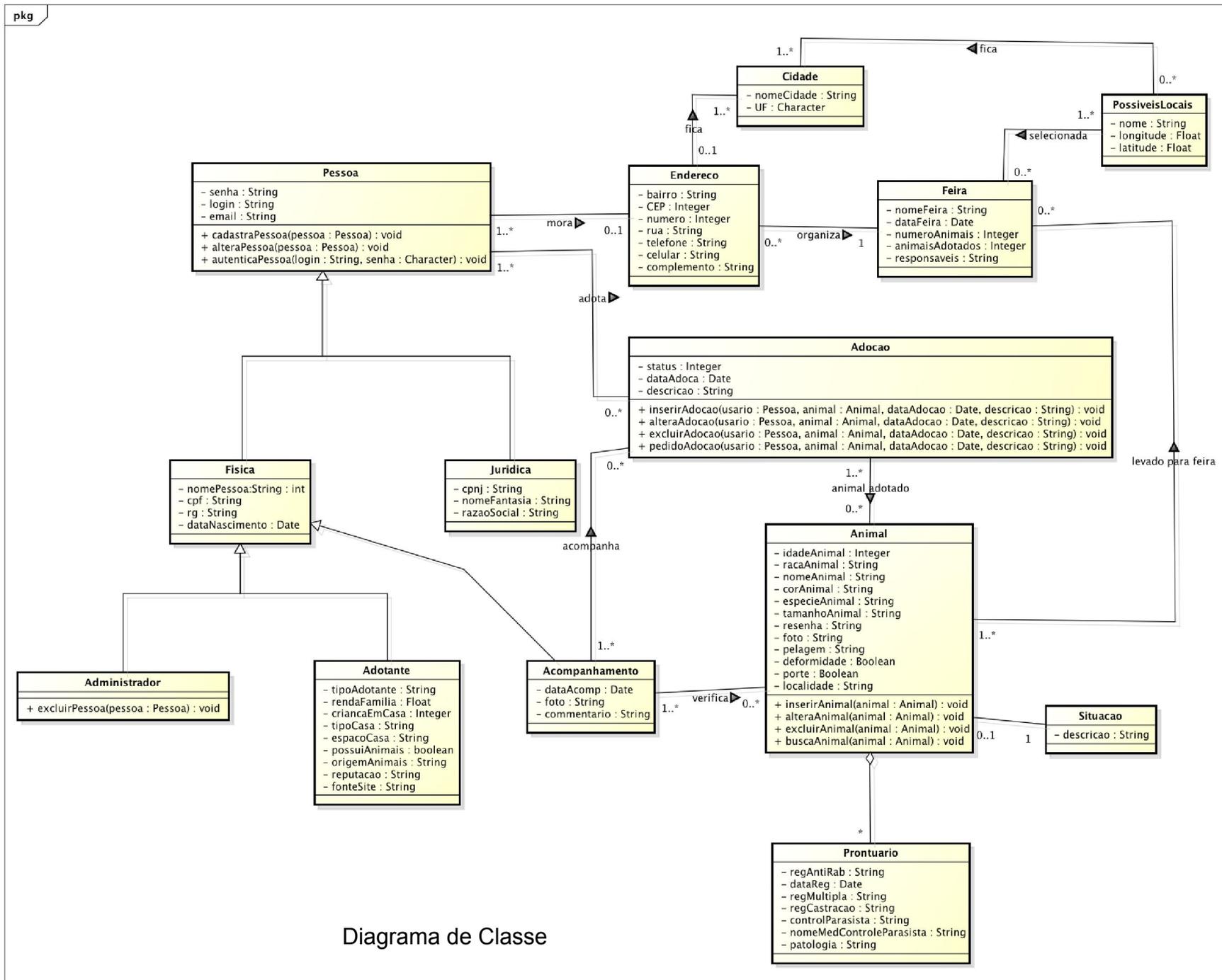


Diagrama de Classe

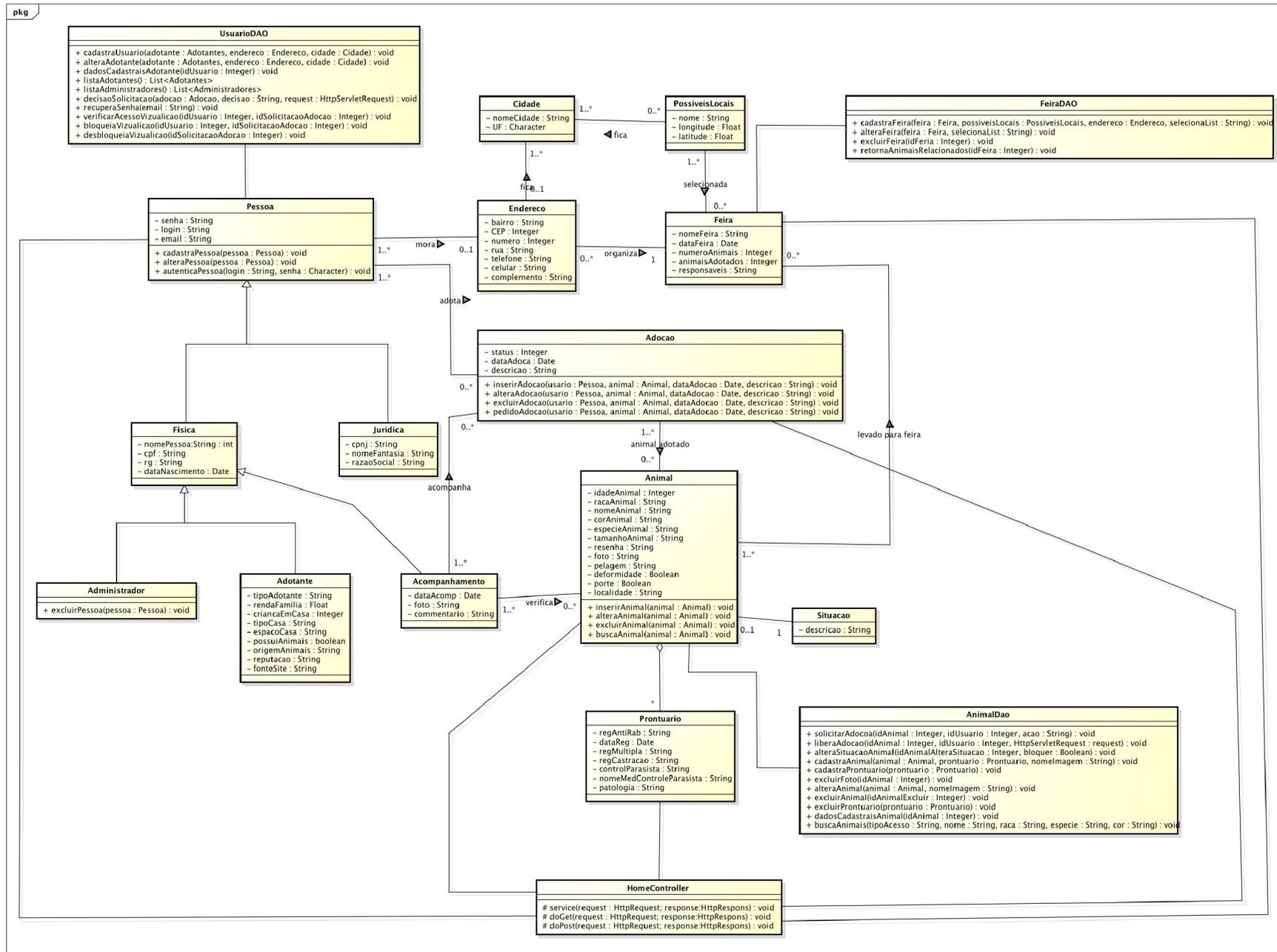


Diagrama de Classe de Implementação MVC e DAO

APÊNDICE F

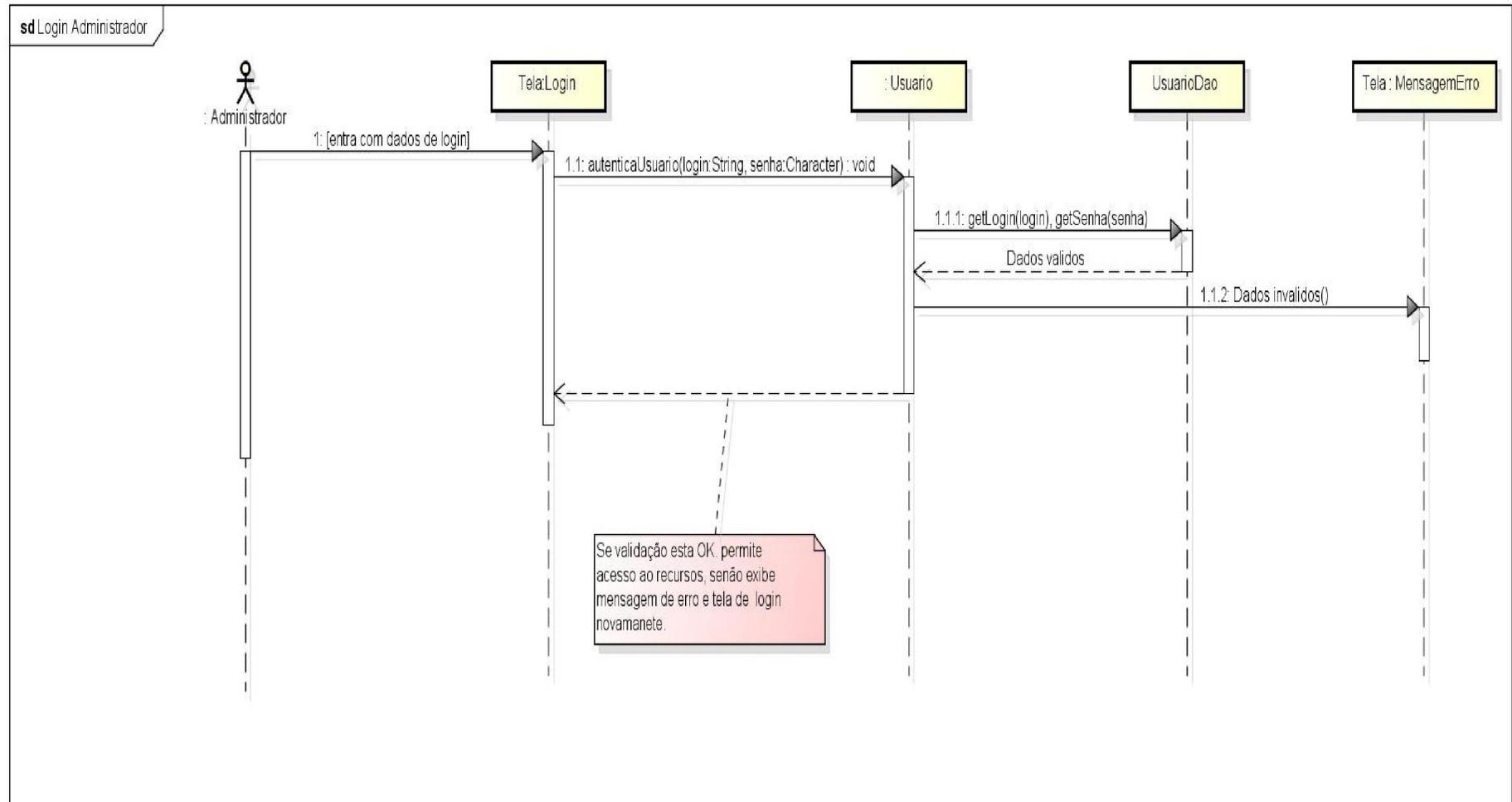


Diagrama de Sequência: Login Administrador

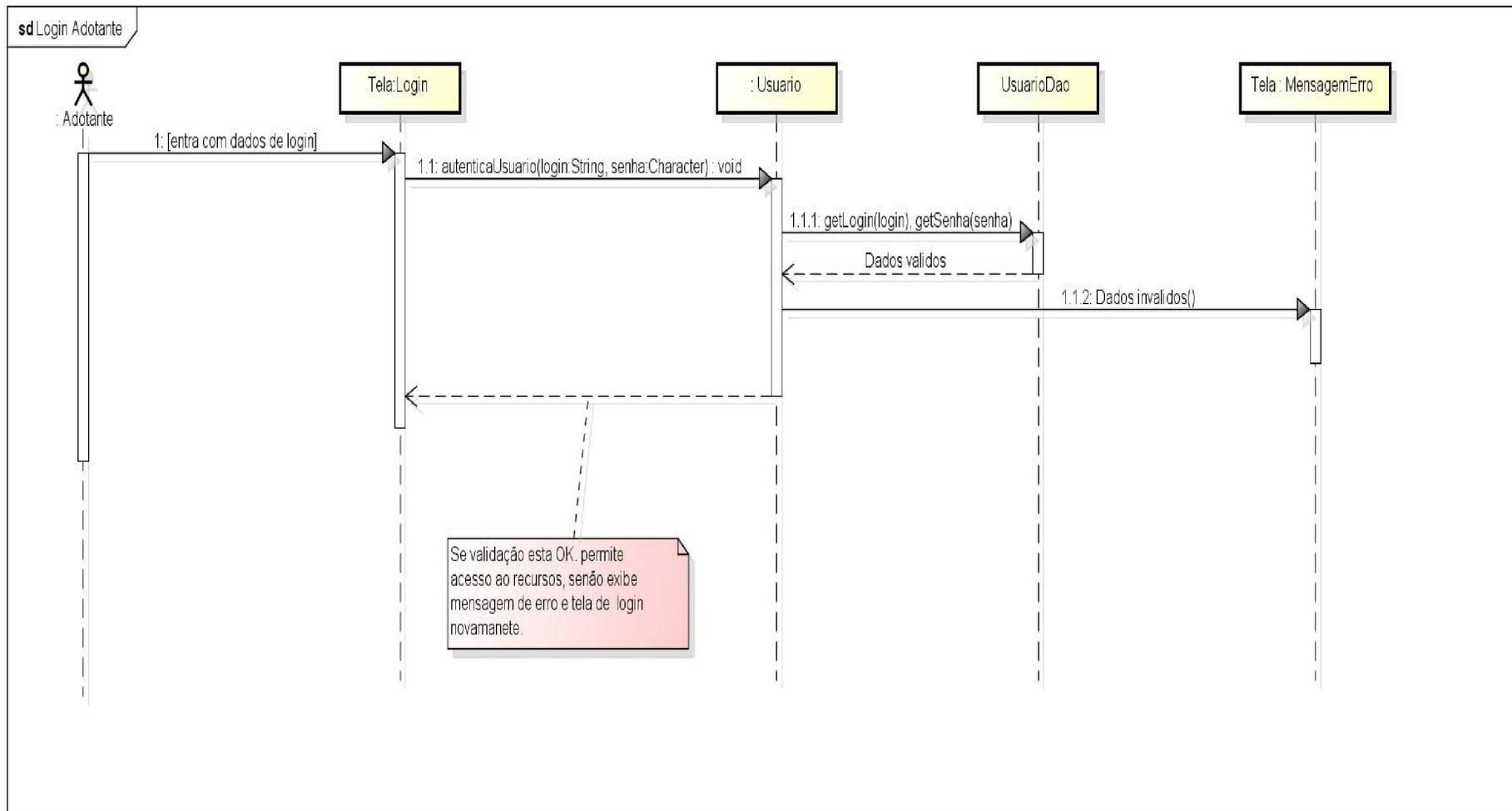


Diagrama de Sequência: Login Adotante

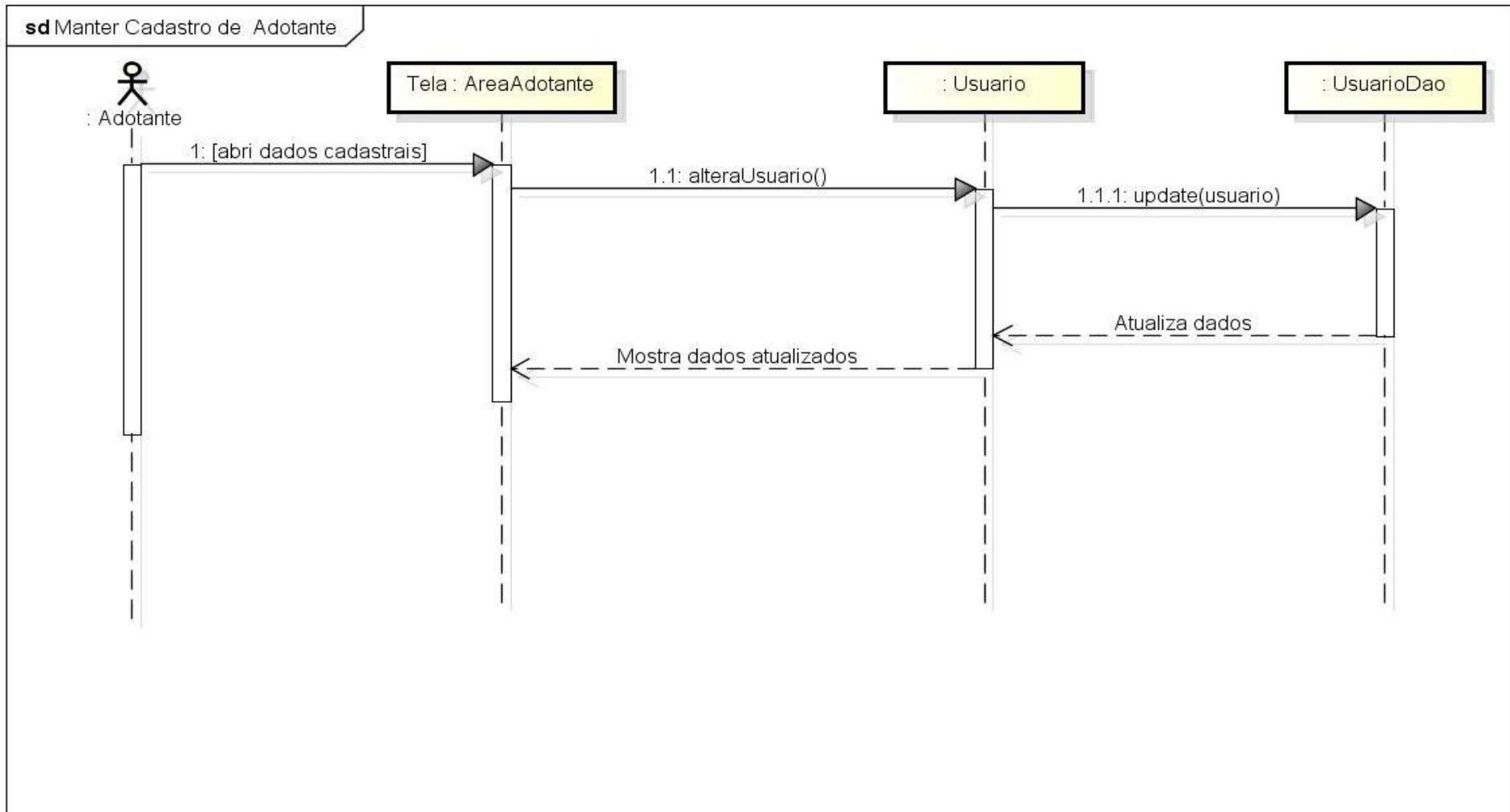


Diagrama de Sequência: Manter Cadastro de Adotante

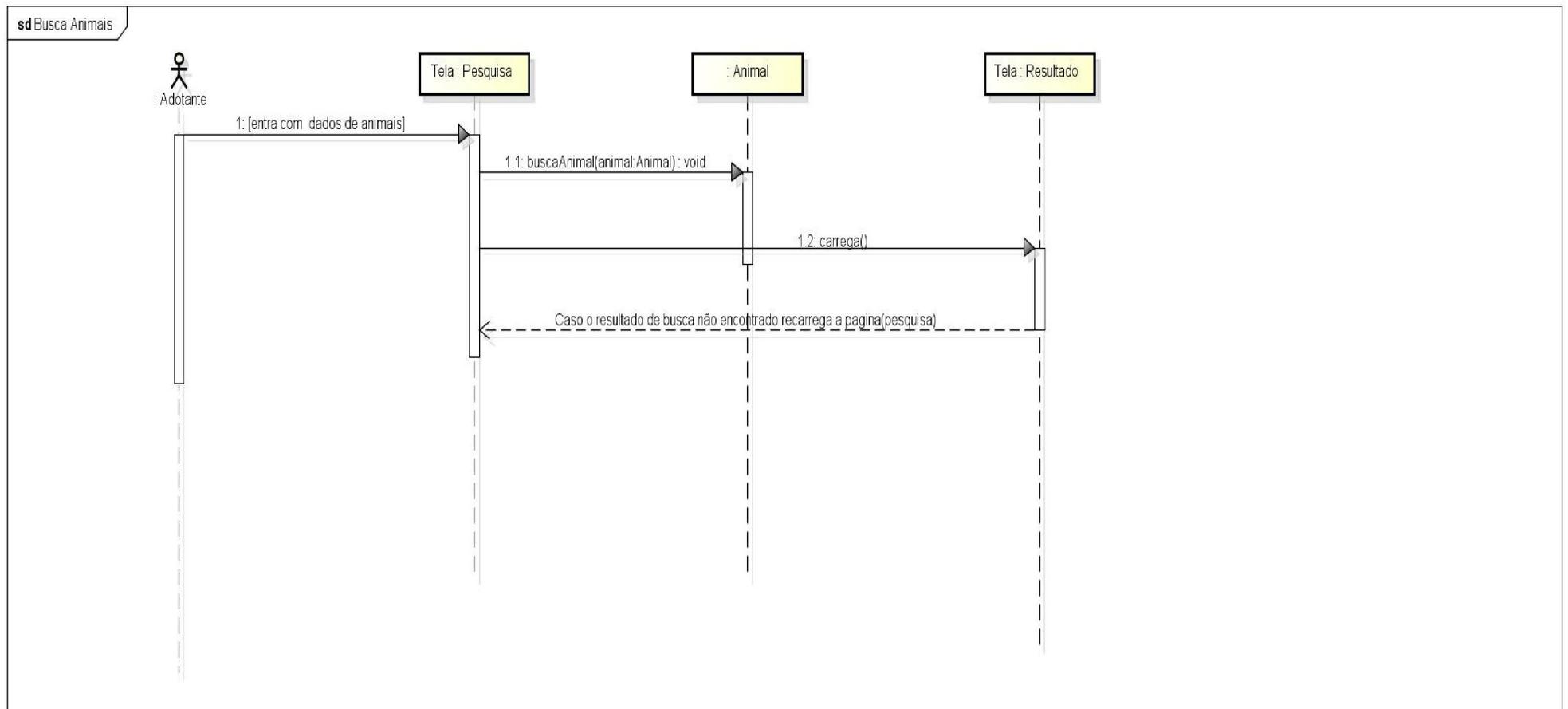


Diagrama de Sequência: Busca Animais

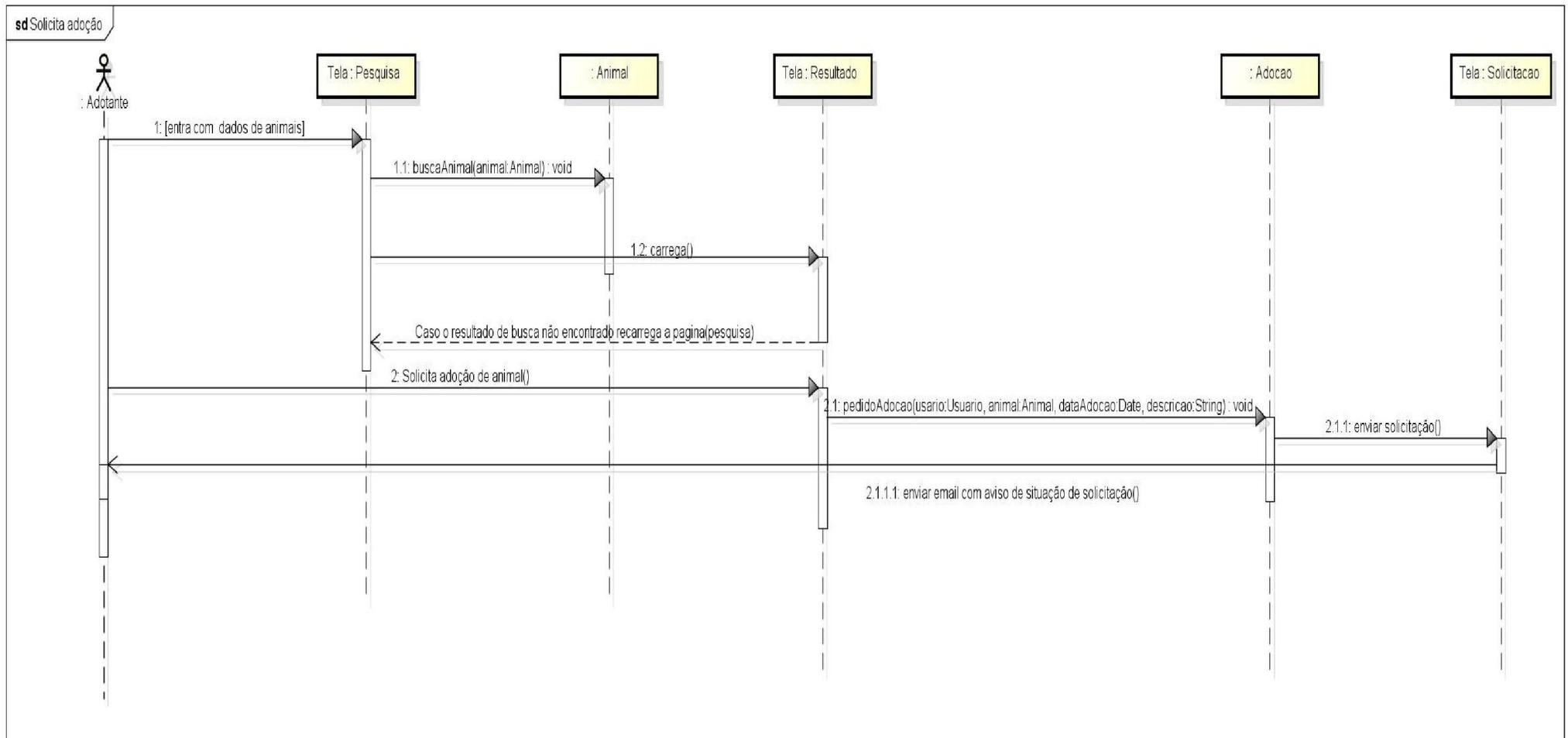


Diagrama de Sequência: Solicita Adoção

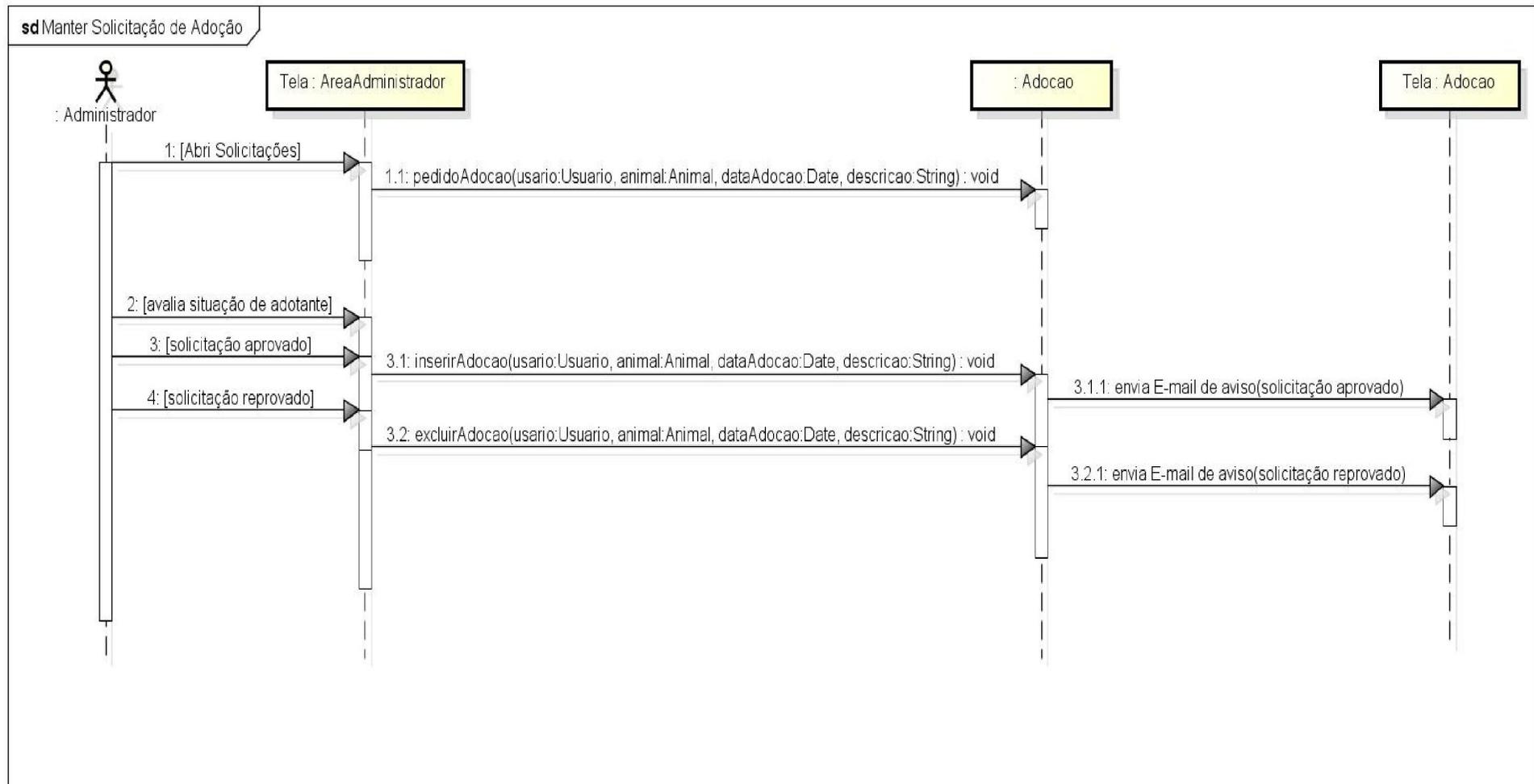


Diagrama de Sequência: Manter Solicitação de Adoção

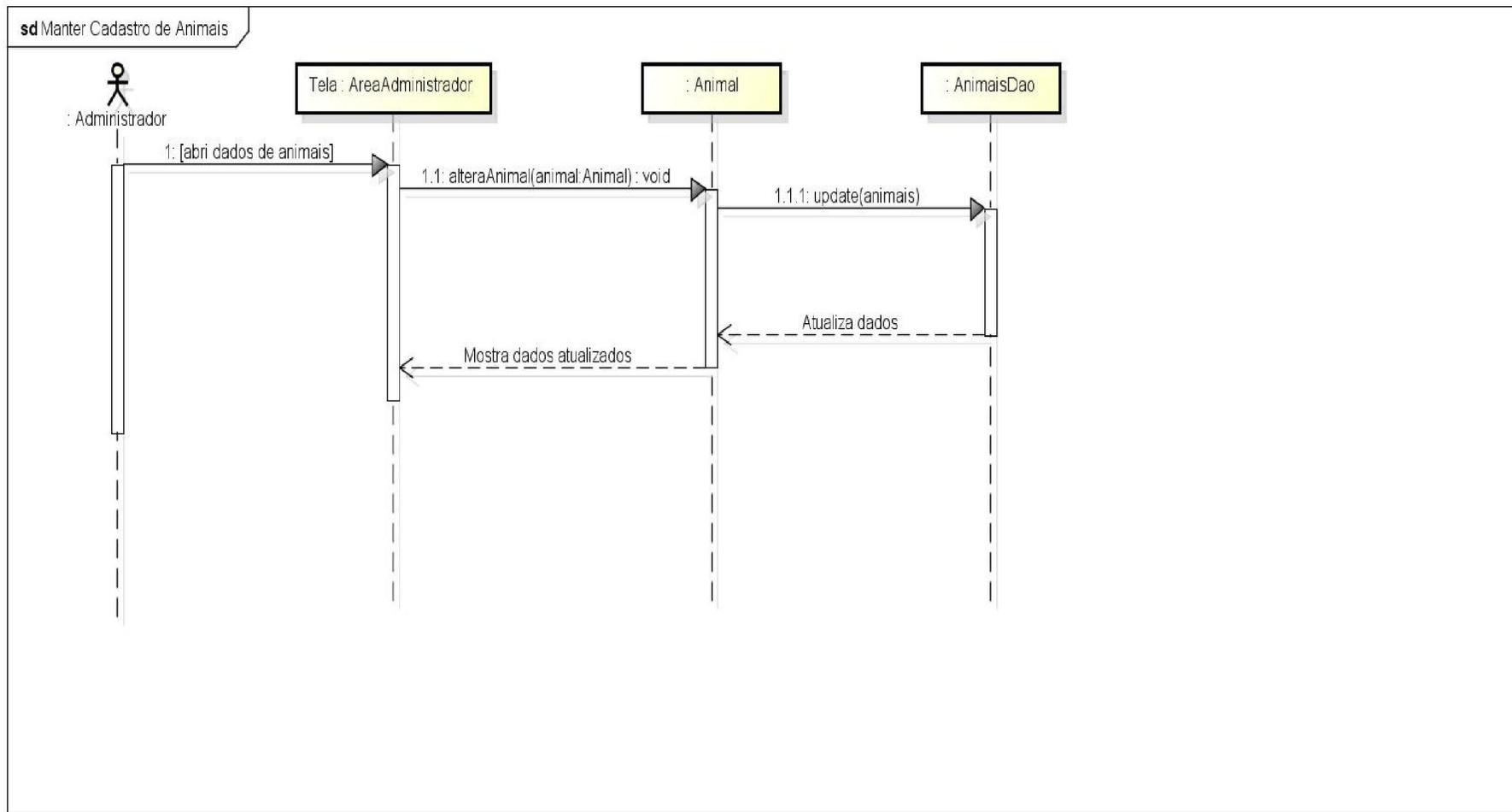
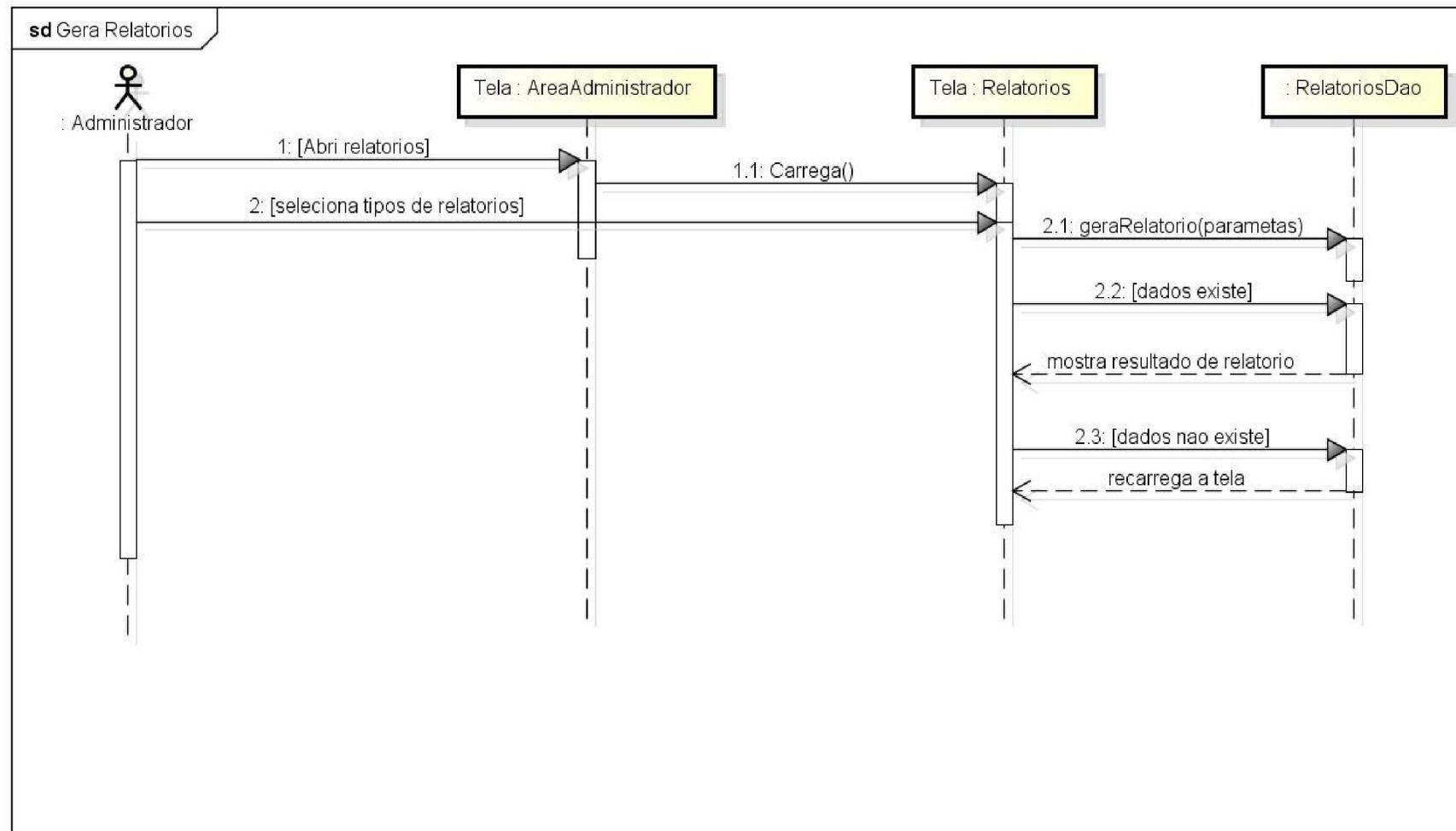


Diagrama de Sequência: Manter Cadastro de Animais



powered by Astah

Diagrama de Sequência: Gera Relatórios

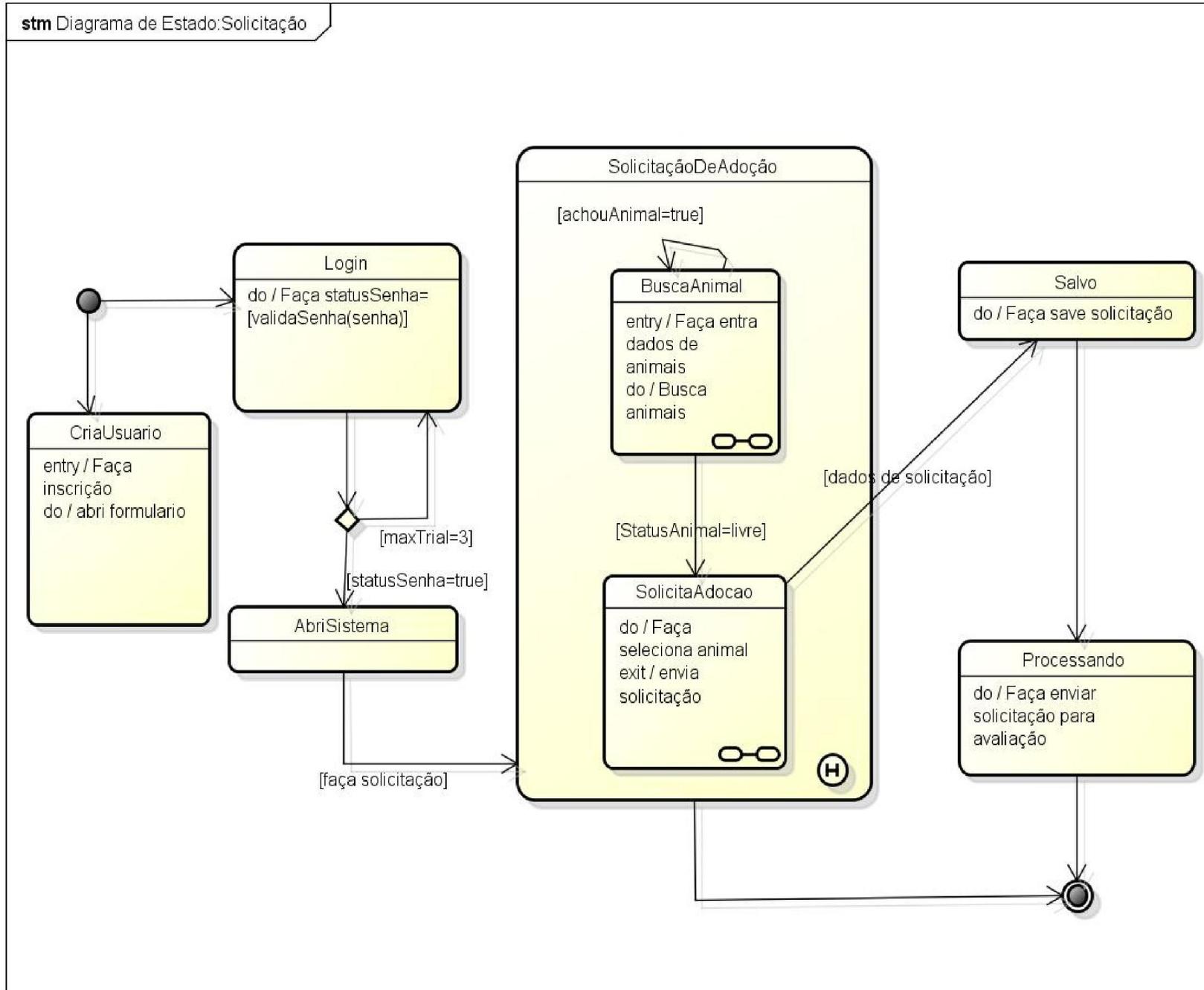


Diagrama de Estado: Solicita Adoção

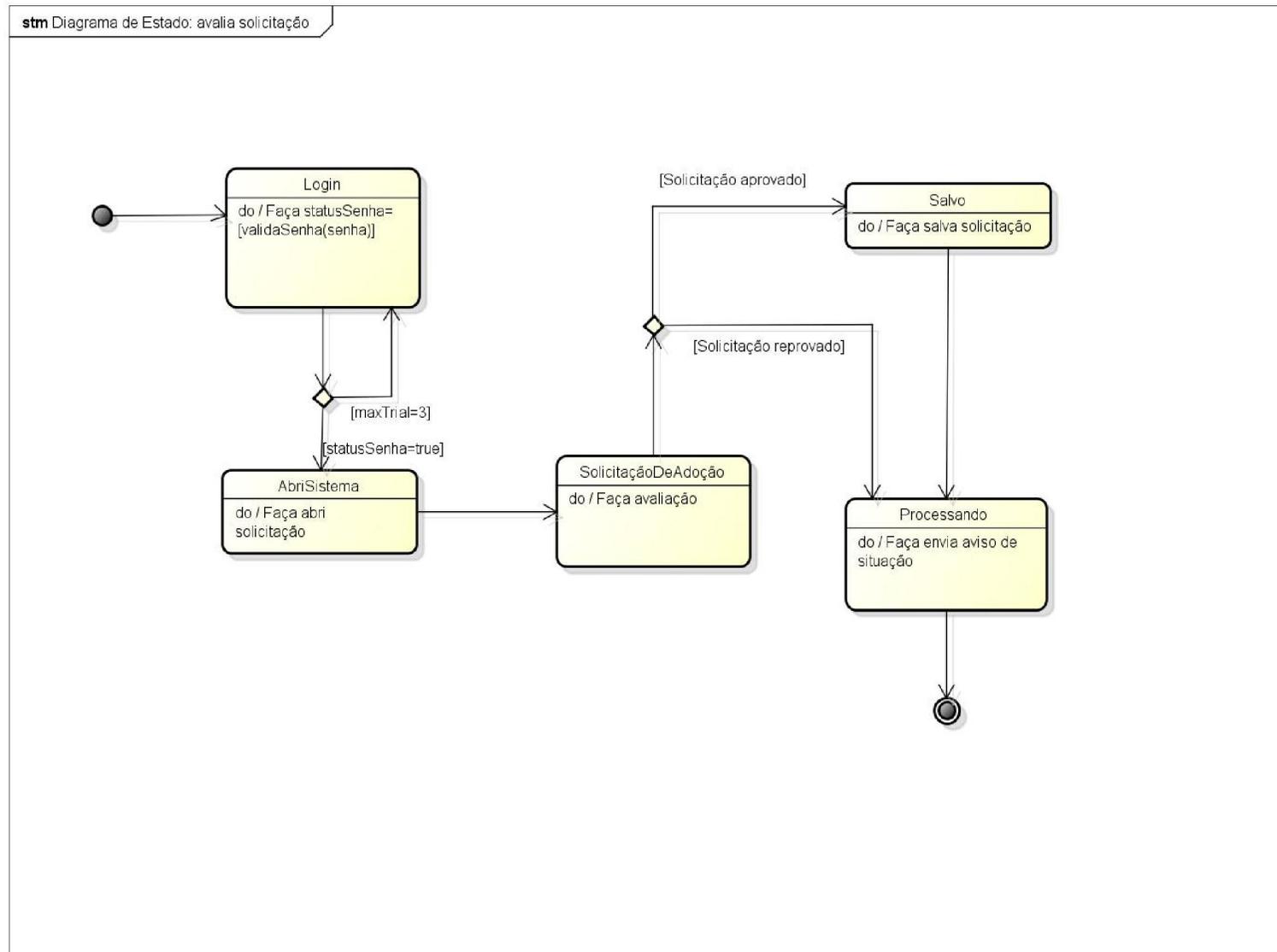


Diagrama de Estado: Avalia Solicitação

APÊNDICE H

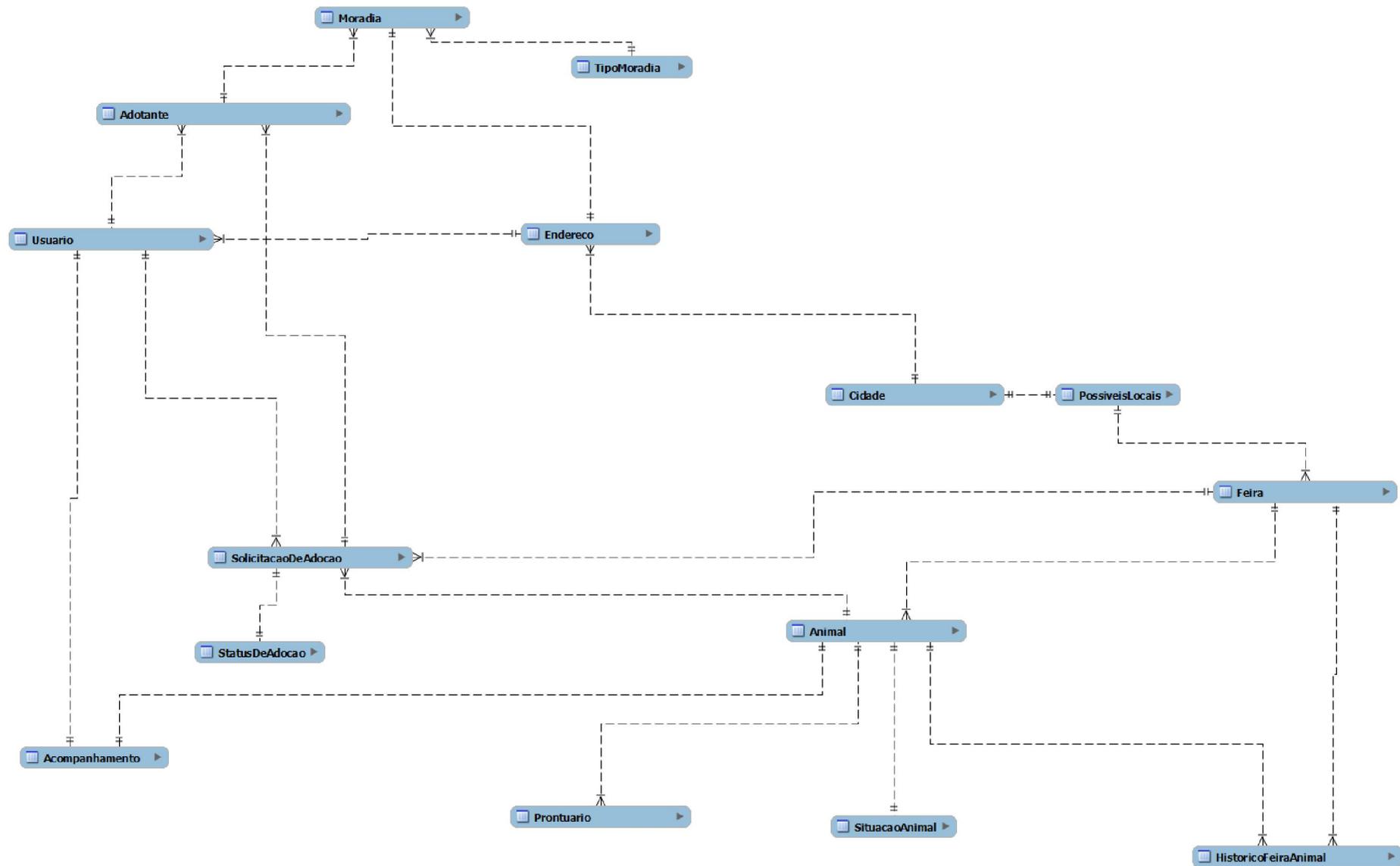


DIAGRAMA ENTIDADE RELACIONAMENTO SIMPLIFICADO (DER)

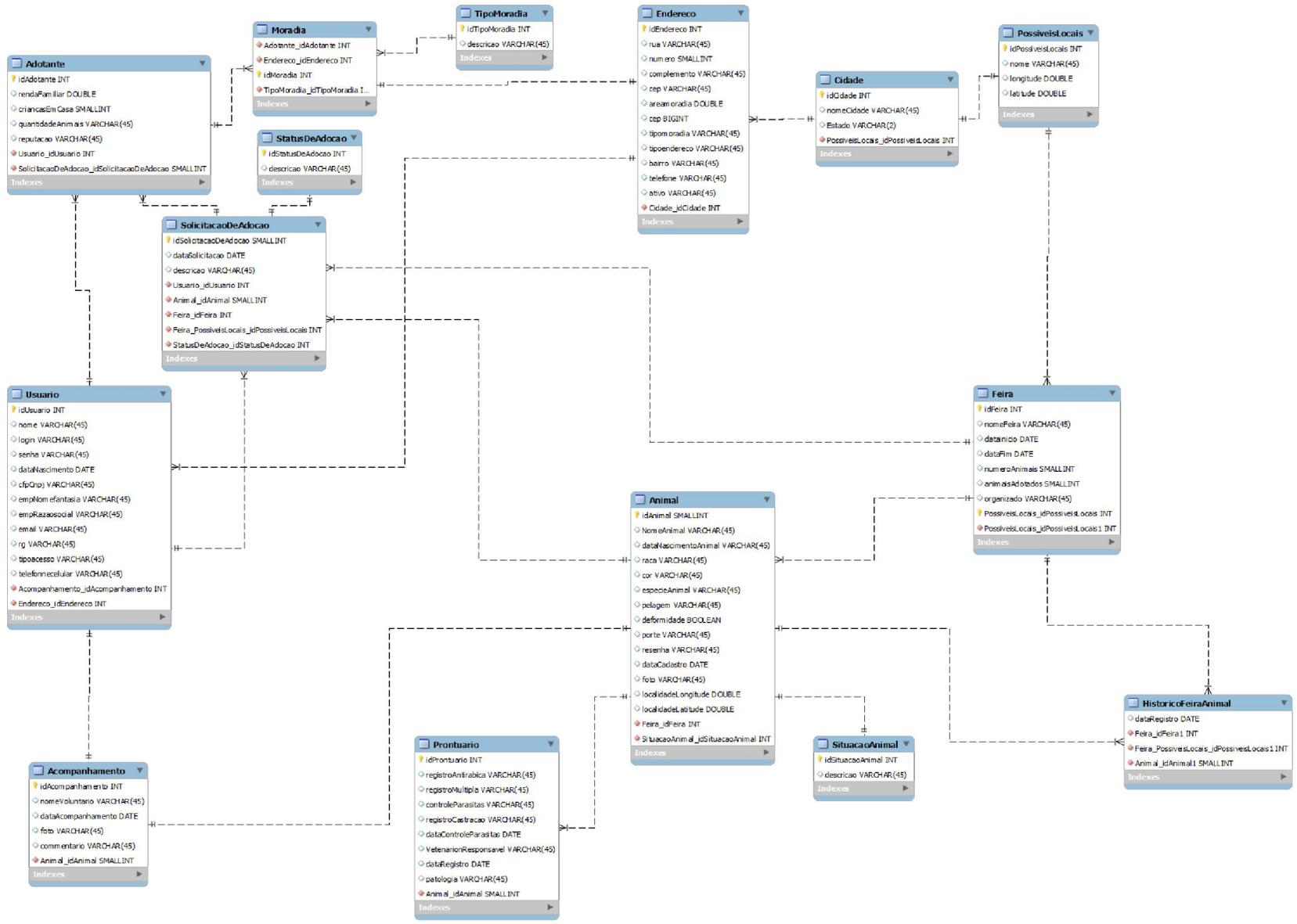


Diagrama de Entidade Relacionamento (DER)

APÊNDICE I

Dicionário de Dados Tabela: acompanhamento (serve para guardar dados de acompanhamento de animais)

Colunas

Nome	Tipo de Dado	Não Null?	Chave Primária?	Default	Comentário
idacompanhamento	integer	Sim	Sim	nextval('acompanhamento_id acompanhamento_seq'::regcl ass)	Campo para armazenar id de acompanhamento
idusuario_usuario	integer	Sim	Não		Campo para relacionar com tabela usuário
idsolicitacaoadocao_solicitacaoadocao	integer	Sim	Não		Campo para relacionar com tabela solicitação
statusacompanhamento	character(15)	Não	Não		Campo para armazenar status de acompanhamento
comentario	text	Não	Não		Campo para armazenar comentários
imagem	character(100)	Não	Não		Campo para guardar links de imagens

Restrições

Nome	Tipo	Definição	Comentário
idacompanhamento	Chave Primária	(idacompanhamento)	Campo para armazenar id de acompanhamento
solicitacaoadocao_fk	Chave Estrangeira	(idsolicitacaoadocao_solicitacaoadocao) REFERENCES solicitacaoadocao (idsolicitacaoadocao) MATCH FULL ON UPDATE CASCADE ON DELETE RESTRICT	Campo para relacionar com tabela usuário
usuario_fk	Chave Estrangeira	(idusuario_usuario) REFERENCES usuario (idusuario) MATCH FULL ON UPDATE CASCADE ON DELETE RESTRICT	Campo para relacionar com tabela solicitação

Dicionário de Dados Tabela: adotante (armazena os dados do adotante)

Colunas

Nome	Tipo de Dados	Não Null?	Chave Primária?	Default	Comentário
idadotante	integer	Sim	Sim	nextval('adotante_idadotante_seq'::regclass)	Campo para guardar id de tabela adotante
rendafamiliar	double precision	Não	Não		Serve para armazenar valor em moeda de renda familiar
criancasemcasa	smallint	Não	Não		Campo para armazenar número de crianças em casa

quantidadeanimais	smallint	Não	Não		Utiliza para salvar número de animais em casa
reputacao	smallint	Não	Não		Serve para gerar reputação de adotante
idusuario_usuario	integer	Sim	Não		Campo para relacionar com a tabela usuário

Restrições

Nome	Tipo	Definição	Comentário
idadotante	Chave Primária	(idadotante)	Campo para guardar id de tabela adotante
usuario_fk	Chave Primária	(idusuario_usuario) REFERENCES usuario (idusuario) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE	Campo para relaciona com a tabela usuário

Dicionário de Dados Tabela: animal (armazena dados de animais)

Colunas

Nome	Tipo de Dado	Não Null?	Chave Primária?	Default	Comentário
Idanimal	integer	Sim	Sim	nextval('animal_idanimal_seq'::regclass)	Campo para salvar id de tabela animal
nome	character(45)	Não	Não		Campo para guardar nome de animal
Datanascimento	date	Não	Não		Campo para armazenar data de

					nascimento de animal
Raca	character(45)	Não	Não		Serve para armazenar raça de animal
Cor	character(25)	Não	Não		Utilizar para salvar cor de animal
Espécie	character(45)	Não	Não		Campo para salvar espécie de animal
Pelagem	character(45)	Não	Não		Campo para salvar pelagem de animal
Deformidade	boolean	Não	Não		Utiliza para armazenar deformidade de animal
Porte	character(10)	Não	Não		Serve para salvar tipo de porte de animal
Resenha	text	Não	Não		Campo para salvar tipo de resenha de animal
Datacadastro	date	Não	Não		Campo data para armazenar data de cadastro.
Foto	character(100)	Não	Não		Serve para salvar links de imagens de animal
Localidadelongitude	double precision	Não	Não		Utilizar para salvar coordenadas
Localidadelatitute	double precision	Não	Não		Utilizar para salvar coordenadas
idsituacao_situacao	integer	Não	Não		Campo para relacionar com a tabela situação
idfeira_feira	integer	Não	Não		Campo para relacionar com a tabela feira
Codigoanimal	character(40)	Não	Não		Serve para gerar código de animal
Localidadelongitude	double precision	Não	Não		Utilizar para salvar coordenadas
Localidadelatitute	double precision	Não	Não		Utilizar para salvar coordenadas

Restrições

Nome	Tipo	Definição	Comentário
idanimal	Chave Primária	(idanimal)	Campo único para identificar animal
feira_fk	Chave Estrangeira	(idfeira_feira) REFERENCES feira (idfeira) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE	Utilizar para relacionar com a tabela feira
situacao_fk	Chave Estrangeira	(idsituacao_situacao) REFERENCES situacao (idsituacao) MATCH FULL ON UPDATE CASCADE ON DELETE RESTRICT	Campo para relacionar com a tabela situação
codigoanimal	Unique	(codigoanimal)	Campo único para gerar código animal

Dicionário de Dados Tabela: cidade (armazena os nomes da cidade do adotante)

Colunas

Nome	Tipo de Dados	Não Null?	Chave Primária?	Default	Comentário
Idcidade	integer	Sim	Sim	nextval('cidade_idcidade_seq':: regclass)	Campo identificador de tabela cidade
Nome	character(45)	Não	Não		Utilizar para armazenar nome de cidade

Estado	character(2)	Não	Não		Serve para armazenar UF de estado
--------	--------------	-----	-----	--	-----------------------------------

Restrições

Nome	Tipo	Definição	Comentário
idcidade	Chave Primária	(idcidade)	Funciona para identificar cidade

Dicionário de Dados Tabela: endereço(armazena os dados de endereço do usuário)

Colunas

Nome	Tipo de Dados	Não Null?	Chave Primária?	Default	Comentário
Endereço	integer	Sim	Sim	nextval('endereco_idendereco_seq'::regclass)	Campo para identificar tabela endereço
Rua	character(45)	Não	Não		Campo armazenar nome da rua
Numero	smallint	Não	Não		Serve para armazenar número de casa
Complemento	character(75)	Não	Não		Campo para guardar complemento de endereço
Cep	bigint	Não	Não		Funciona para armazenar código postal de endereço
Área moradia	double precision	Não	Não		Campo para lista área de moradia

Tipoendereco	character(6)	Não	Não		Funciona para diferenciar tipo de endereço
Bairro	character(45)	Não	Não		Campo para armazenar nome de bairro.
Telefone	character(45)	Não	Não		Campo para salvar número de telefone fixo
Ativo	boolean	Não	Não		Serve para adicionar outro endereço
idusuario_usuario	integer	Sim	Não		Funciona para relacionar com a tabela usuário
idcidade_cidade	integer	Sim	Não		Campo para <i>linkar</i> com a tabela cidade

Restrições

Nome	Tipo	Definição	Comentário
idendereco	Chave Primária	(idendereco)	Identificador de tabela endereço
cidade_fk	Chave Estrangeira	(idcidade_cidade) REFERENCES cidade (idcidade) MATCH FULL ON UPDATE CASCADE ON DELETE RESTRICT	Chave estrangeira de tabela cidade serve para relacionar com a tabela endereço
usuario_fk	Chave Estrangeira	(idusuario_usuario) REFERENCES usuario (idusuario) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE	Chave estrangeira de tabela usuário serve para relacionar os dados entre tabela endereço e usuário.

Dicionário de Dados Tabela: feira (tabela para armazenar dados de feiras)

Colunas

Nome	Tipo de Dados	Não Null?	Chave Primária?	Default	Comentário
idfeira	integer	Sim	Sim	nextval('feira_idfeira_seq'::regclass)	Campo identificador de tabela feira
nomefeira	character(45)	Não	Não		Funciona para armazenar nome de feira.
datainicio	timestamp without time zone	Não	Não		Armazenar data de inicio de feira
dataencerramento	timestamp without time zone	Não	Não		Armazenar data de encerramento de feira.
numeroanimais	smallint	Não	Não		Armazenar numero de animais levados a feira.
idpossiveislocais_possiveislocais	integer	Não	Não		Campo para relacionar com a tabela possíveis locais.
longitude	double precision	Não	Não		Armazenar longitude de local de feira
latitude	double precision	Não	Não		Armazenar latitude de local de feira.
idcidade_cidade	integer	Não	Não		Campo para relacionar com a tabela cidade.
ativo	boolean	Não	Não		Serve para ativar ou desativar uma feira

organizadorresponsavel	character(50)	Não	Não		Armazenar nome de organizador de feira
------------------------	---------------	-----	-----	--	--

Restrições

Nome	Tipo	Definição	Comentário
idfeira	Chave Primária	(idfeira)	Chave Primária para identificar dados de colunas de tabela feira
cidade_fk	Chave Estrangeira	(idcidade_cidade) REFERENCES cidade (idcidade) MATCH FULL ON UPDATE CASCADE ON DELETE RESTRICT	Chave estrangeira serve para relacionar os dados de tabela cidade com uma feira.

Dicionário de Dados Tabela: historicofeiraanimal (tabela para armazenar histórico de feira)

Colunas

Nome	Tipo de Dados	Não Null?	Chave Primária?	Default	Comentário
idhistoricofeiraanimal	integer	Sim	Sim	nextval('historicofeiraanimal_idhis	Campo identificador de tabela

				toricofeiraanimal_seq)::regclass)	
idfeira_feira	integer	Não	Não		Campo para cruzar os dados de tabela feira
idanimal_animal	integer	Não	Não		Campo para cruzar os dados de tabela animal com esta tabela
dataregistro	timestamp without time zone	Não	Não		Armazenar data de registro de feria

Restrições

Nome	Tipo	Definição	Comentário
idhistoricofeiraanimal	Chave Primária	(idhistoricofeiraanimal)	Chave para identificar dados de colunas de tabela historicofeiraanimal
animal_fk	Chave Estrangeira	(idanimal_animal) REFERENCES animal (idanimal) MATCH FULL ON UPDATE CASCADE ON DELETE RESTRICT	Chave estrangeira utiliza para cruzar os dados com esta tabela
feira_fk	Chave Estrangeira	(idfeira_feira) REFERENCES feira (idfeira) MATCH FULL ON UPDATE CASCADE ON DELETE RESTRICT	Chave estrangeira serve para relacionar com os dados de tabela feira

Dicionário de Dados Tabela: moradia(tabela para armazenar dados de moradia de adotante)

Colunas

Nome	Tipo de Dados	Não Null?	Chave Primária?	Default	Comentário
------	---------------	-----------	-----------------	---------	------------

Idmoradia	integer	Sim	Sim	nextval('moradia_idmoradia_seq'::regclass)	Campo identificador da tabela
idadotante_adotante	integer	Sim	Não		Campo para cruzar os dados de tabela adotante com tabela moradia
idtipomoradia_tipomoradia	integer	Não	Não		Campo para cruzar os dados de tabela tipomoradia com tabela moradia
idendereco_endereco	integer	Não	Não		Campo para cruzar os dados de tabela endereço com tabela moradia

Restrições

Nome	Tipo	Definição	Comentário
idmoradia	Chave Primária	(idmoradia)	Chave identificador de tabela moradia
adotante_fk	Chave Estrangeira	(idadotante_adotante) REFERENCES adotante (idadotante) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE	Chave estrangeira para cruzar os dados de tabela adotante com esta tabela
endereco_fk	Chave Estrangeira	(idendereco_endereco) REFERENCES endereco (idendereco) MATCH FULL ON UPDATE CASCADE ON DELETE CASCADE	Chave para cruzar os dados de tabela endereço com tabela moradia
tipomoradia_fk	Chave Estrangeira	(idtipomoradia_tipomoradia) REFERENCES tipomoradia (idtipomoradia) MATCH FULL ON UPDATE CASCADE ON DELETE RESTRICT	Campo para cruzar os dados de tabela tipomoradia com tabela moradia
moradia_uq1	Unique	(idendereco_endereco)	Chave única para diferenciar tipo de moradia.

Dicionário de Dados Tabela: possiveislocais (serve para armazenar dados de possíveis locais para feira)

Colunas

Nome	Tipo de Dados	Não Null?	Chave Primária?	Default	Comentário
idpossiveislocais	integer	Sim	Sim	nextval('possiveislocais_idpossiveislocais_seq'::regclass)	Campo para identificador de tabela
Nome	character(45)	Não	Não		Armazenar nome de possíveis locais para feira.
Longitude	double precision	Não	Não		Armazenar longitude de tabela possíveis locais para feira.
Latitude	double precision	Não	Não		Armazenar latitude de tabela possíveis locais para feira.
idcidade_cidade	integer	Não	Não		Campo para cruzar dados de tabela cidade com a tabela possíveis locais.

Restrições

Nome	Tipo	Definição	Comentário
idpossiveislocais	Chave Primária	(idpossiveislocais)	Chave identificador de tabela

Dicionário de Dados Tabela: prontuário (tabela para guardar os dados de prontuário de animais)

Colunas

Nome	Tipo de Dados	Não Null?	Chave Primária?	Default	Comentário
idprontuario	integer	Sim	Sim	nextval('prontuario_idprontuario_seq':regclass)	Campo identificador de tabela
registroantirabica	character(100)	Não	Não		Armazenar os registros de antirábicas.
registromultipla	character(100)	Não	Não		Armazenar os registros de múltiplas.
registrocastracao	character(100)	Não	Não		Armazenar os registros de castração.
controleparasitas	character(100)	Não	Não		Armazenar os registros de controle de parasitas.
datacontroleparasitas	date	Não	Não		Armazenar data de controle de parasitas
veterinarioresponsavelcontroleparasitas	character(45)	Não	Não		Armazenar o nome de médico responsável.
dataregistro	date	Não	Não		Guardar as datas de registro.
patologia	character(100)	Não	Não		Armazenar os registros patologia.
idanimal_animal	integer	Sim	Não		Campo para cruzar dados de tabela animal com esta tabela

Restrições

Nome	Tipo	Definição	Comentário
idprontuario	Chave Primária	(idprontuario)	Campo identificador de tabela
animal_fk	Chave Estrangeira	(idanimal_animal) REFERENCES animal (idanimal) MATCH FULL ON UPDATE CASCADE ON DELETE RESTRICT	Chave para cruzar os dados de tabela animal com tabela prontuário.

Dicionário de Dados Tabela: situação (tabela para armazenar dados de situação de animal)

Colunas

Nome	Tipo de Dados	Não Null?	Chave Primária?	Default	Comentário
idsituacao	integer	Sim	Sim	nextval('situacao_idsituacao_seq'::regclass)	Campo identificador de tabela
Descrição	character(25)	Não	Não		Armazenar descrição de animais

Restrições

Nome	Tipo	Definição	Comentário
idsituacao	Chave Primária	(idsituacao)	Chave identificador de tabela

Dicionário de Dados Tabela: solicitacaoadocao (tabela para armazenar dados de solicitação de adoção)

Colunas

Nome	Tipo de Dados	Não Null?	Chave Primária?	Default	Comentário
idsolicitacaoadocao	integer	Sim	Sim	nextval('solicitacaoadocao_idsolicitacaoadocao_seq'::regclass)	Identificador de tabela
datasolicitacaoadocao	date	Não	Não		Campo para armazenar data de solicitação de adoção
descricao	text	Não	Não		Armazena descrição de solicitação de adoção
idadotante_adotante	integer	Não	Não		Campo para cruzar dados de tabela adotante e esta tabela
idanimal_animal	integer	Não	Não		Campo para cruzar dados de tabela animal e esta tabela.
idusuario_usuario	integer	Não	Não		Campo para cruzar dados de tabela usuário e esta tabela.
idfeira_feira	integer	Não	Não		Campo para cruzar dados de tabela feira e esta tabela.
idstatus_status	integer	Não	Não		Campo para cruzar dados de tabela status e

					esta tabela.
dataalteracaostatus	timestamp without time zone	Não	Não		Campo para armazenar datas de alteração de status.
indicativocontratogerado	boolean	Não	Não		Campo para identificar a geração de contrato.
datavalidadecomparecimento	timestamp without time zone	Não	Não		Armazenar data de retirada de animal.
idadminvisualizacao	integer	Não	Não		Campo para identificar visualização de administrador.
validadevisualizacao	timestamp without time zone	Não	Não		Armazenar data de visualização de admin.

Restrições

Nome	Tipo	Definição	Comentário
idsolicitacaoadocao	Chave Primária	(idsolicitacaoadocao)	Campo identificador de tabela
adotante_fk	Chave Estrangeira	(idadotante_adotante) REFERENCES adotante (idadotante) MATCH FULL ON UPDATE CASCADE ON DELETE RESTRICT	Serve para cruzar os dados de tabela adotante com esta tabela
animal_fk	Chave Estrangeira	(idanimal_animal) REFERENCES animal	Serve para cruzar os dados de tabela

		(idanimal) MATCH FULL ON UPDATE CASCADE ON DELETE RESTRICT	animal com esta tabela
feira_fk	Chave Estrangeira	(idfeira_feira) REFERENCES feira (idfeira) MATCH FULL ON UPDATE CASCADE ON DELETE SET NULL	Serve para cruzar os dados de tabela feira com esta tabela
status_fk	Chave Estrangeira	(idstatus_status) REFERENCES status (idstatus) MATCH FULL ON UPDATE CASCADE ON DELETE SET NULL	Serve para cruzar os dados de tabela status com esta tabela
usuario_fk	Foreign key	(idusuario_usuario) REFERENCES usuario (idusuario) MATCH FULL ON UPDATE CASCADE ON DELETE RESTRICT	Serve para cruzar os dados de tabela usuário com esta tabela
solicitacaoadocao_uq	Unique	(idadotante_adotante)	Chave único para identificar adotante
solicitacaoadocao_uq1	Unique	(idanimal_animal)	Chave único para identificar animal
solicitacaoadocao_uq2	Unique	(idusuario_usuario)	Chave único para identificar usuário
solicitacaoadocao_uq3	Unique	(idfeira_feira)	Chave único para identificar feira

Dicionário de Dados Tabela: status (tabela para armazenar status de animais)

Colunas

Nome	Tipo de Dados	Não Null?	Chave Primária?	Default	Comentário
Idstatus	integer	Sim	Sim	nextval('status_idstatus_seq' ::regclass)	Campo identificador de tabela status
Descrição	character(25)	Não	Não		Armazenar descrição de status de animais

Restrições

Nome	Tipo	Definição	Comentário
idstatus	Chave Primária	(idstatus)	Chave única para identificar status de animais

Dicionário de Dados Tabela: tipomoradia (tabela para guardar dados de tipo de moradia)

Colunas

Nome	Tipo de Dados	Não Null?	Chave Primária?	Default	Comentário
Idtipomoradia	integer	Sim	Sim	nextval('tipomoradia_idtipomoradia_seq'::regclass)	Identificador de tabela tipo moradia
Descrição	character(25)	Não	Não		Serve para armazenar descrição de tipo de moradia.

Restrições

Nome	Tipo	Definição	Comentário
idtipomoradia	Chave Primária	(idtipomoradia)	Chave identificador de tabela

Dicionário de Dados Tabela: usuário (tabela para armazenar dados de usuário)

Colunas

Nome	Tipo de Dados	Não Null?	Chave Primária?	Default	Comentário
idusuario	integer	Sim	Sim	nextval('usuario_idusuario_seq'::regclass)	Campo para armazenar código identificador de tabela
nome	character(50)	Não	Não		Campo para armazenar nome de usuário
login	character(50)	Não	Não		Campo para armazenar <i>login</i> de usuário
senha	character(100)	Não	Não		Campo para armazenar senha criptografado de usuário
datanascimento	date	Não	Não		Armazena data de nascimento de usuário.
cpfcnpj	character(14)	Não	Não		Serve para armazenar CPF ou CNPJ de usuário
emp_nomefantasia	character(75)	Não	Não		Campo para armazenar nome fantasia de empresa
emp_razaosocial	character(75)	Não	Não		Campo para armazenar nome de razão social de usuário
email	character(100)	Não	Não		Campo para armazenar <i>e-mail</i> de usuário
rg	character(15)	Não	Não		Serve para guardar RG de usuário

tipoacesso	character(3)	Não	Não		Campo para armazenar id de tipo de acesso de usuário
telefonecelular	character(15)	Não	Não		Campo para armazenar número de telefone
tipopessoa	character(2)	Não	Não		Armazenar identificador de tipo de pessoa

Restrições

Nome	Tipo	Definição	Comentário
idusuario	Chave Primária	(idusuario)	Restrições de tabela usuário
cpfcnpj	Unique	(cpfcnpj)	Chave única para identificar usuário
login	Unique	(login)	Chave única para autenticar login

APÊNDICE J

The image shows a wireframe of a web browser window. The browser's address bar contains a home icon, the text 'Viralata', and navigation links: 'Cadastre-se', 'Duvidas', 'Contato', and 'Login'. The main content area features a circular logo placeholder labeled 'Logo'. Below the logo is the title 'Viralata: Sistema para Gerenciamento de Ação de Animais'. A central login form is enclosed in a rectangular box and contains the following elements: a label 'Login' above the form, a 'Login:' label next to a text input field containing the text 'login', a 'Senha:' label next to a text input field containing the text 'senha', a 'Login' button, a blue link 'Recupera senha.', and a blue link 'Novo usuário? Realizar Cadastro'. At the bottom of the browser window, a footer bar contains the text 'Viralata 2014 todos os direitos reservados'.

PROTÓTIPO TELA LOGIN

The image shows a wireframe of a web browser window. The browser's address bar contains a home icon, the text 'Viralata', and navigation links: 'Cadastre-se', 'Duvidas', 'Contato', and 'Login'. The main content area features a circular placeholder labeled 'Logo'. Below the logo, the title 'Viralata: Sistema para Gerenciamento de Ação de Animais' is displayed. A sub-header reads 'Para realizar cadastro selecion tipo pessoa'. Two radio button options are provided: 'Pessoa Física' and 'Pessoa Jurídica'. At the bottom of the page, a footer box contains the text 'Viralata 2014 todos os direitos reservados'.

PROTÓTIPO TELA SELECIONA TIPO CADASTRO

[Virelata](#) [Cadastre-se](#) [Dúvidas](#) [Contato](#) [Login](#)

Cadastro do Adotante, favor preencher os campos com seu dados

Dados Pessoais

Nome:

Nome Completo

Data Nascimento:

dd/mm/aaaa

CPF/CNPJ:

999-999-999-99

RG:

99 999 999

Email:

email

Celular

celular

Endereço Atual

CEP:

999-99-999

Rua:

Bairro

Bairro

Cidade:

cidade

UF

UF

Numero

numero

Complemento

complemento

Telefone Fixo

telefone fixo

Tipo Moradia

seleciona tipo

Endereço Adicional

CEP:

999-99-999

Rua:

Bairro

Bairro

Cidade:

cidade

UF

UF

Numero

numero

Complemento

complemento

Telefone Fixo

telefone fixo

Tipo Moradia

seleciona tipo

Dados Socioeconomicos

Renda Familia

renda

Numero de filhos

filhos

Quantidade Animais

animais em casa

Usuário

Login

login

Senha

senha

Confirma Senha

confirma senha

Virelata 2014 todos os direitos reservados

Virelata Cadastre-se Dúvidas Contato Login

Cadastro do Adquirente Pessoa Jurídica, favor preencher os campos com seu dados

Dados Pessoais

Nome:

Nome Completo

Razão Social:

Razão social

Nome Fantasia

Nome fantasia

Data Nascimento:

dd/mm/aaaa

CNP.J:

999.999-999.99

RG

99 999.999

Email:

email

Celular

celular

Endereço Atual

CEP:

999.99.999

Rua:

Rua

Bairro

bairro

Cidade:

cidade

UF

UF

Numero

numero

Complemento

complemento

Telefone fixo

telefone fixo

Tipo Moradia

seleciona tipo

Área Moradia em metros

Endereço Adicional

CEP:

999.99.999

Rua:

Rua

Bairro

bairro

Cidade:

cidade

UF

UF

Numero

numero

Complemento

complemento

Telefone fixo

telefone fixo

Tipo Moradia

seleciona tipo

Área Moradia em metros

Dados Socioeconomicos

Renda Familiar

Renda

Numero de filhos

filhos

Quantidade Animais

animais em casa

Usuário

Login

login

Senha

senha

Confirma Senha

confirma senha

Virelata 2014 todos os direitos reservados

Area Adotante Perfil Minha Adoções Busca Animais Duvidas Contato Logout

Logo

Viralata: Sistema para Gerenciamento de Ação de Animais

Lista da minha solicitações

Codigo	Nome Animal	Data	Situação

Viralata 2014 todos os direitos reservados

Home Área Adotante Perfil Minha Adoções Busca Animais Dúvidas Contato Logout

Logo

Viraldia: Sistema para Gerenciamento de Ação de Animais

Dados Cadastrados

Nome:

Nome Completo

Data Nascimento:

dd/mm/aaaa

CPF/CNP J:

qqq-qqq-qqq-qq

RG

qq qq qq qq

CPF/CNP J:

qqq-qqq-qqq-qq

Endereço

CEP:

qqq-qq-qq-qqq

Rua:

Rua

Bairro

bairro

Cidade:

cidade

UF

UF

Numero

numero

Complemento

complemento

Telefone Fixo

telefone fixo

Área Moradia em metros

Tipo Moradia

tipo moradia

area moradia

Login

login

Senha

senha

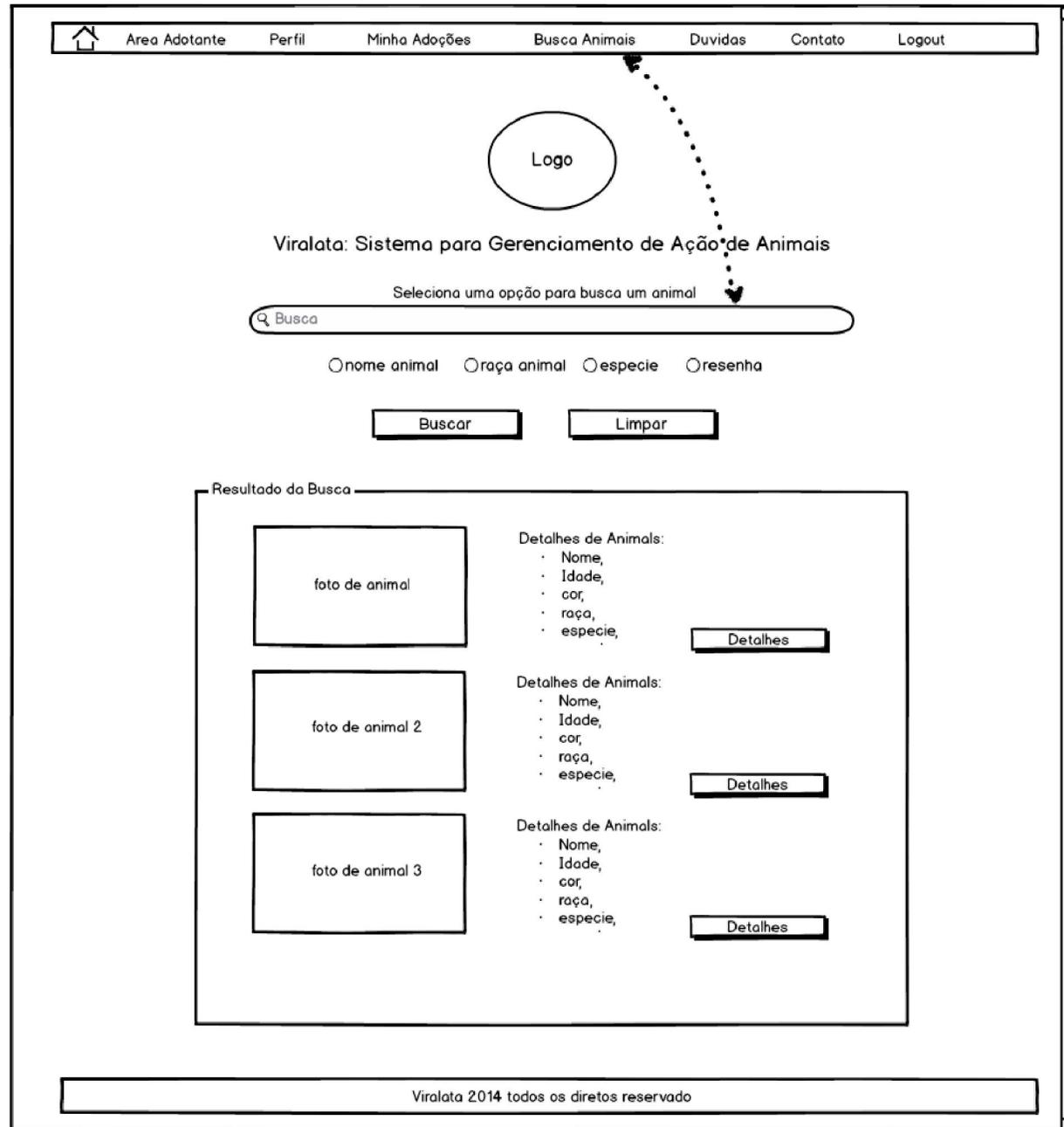
Confirma Senha

confirma senha

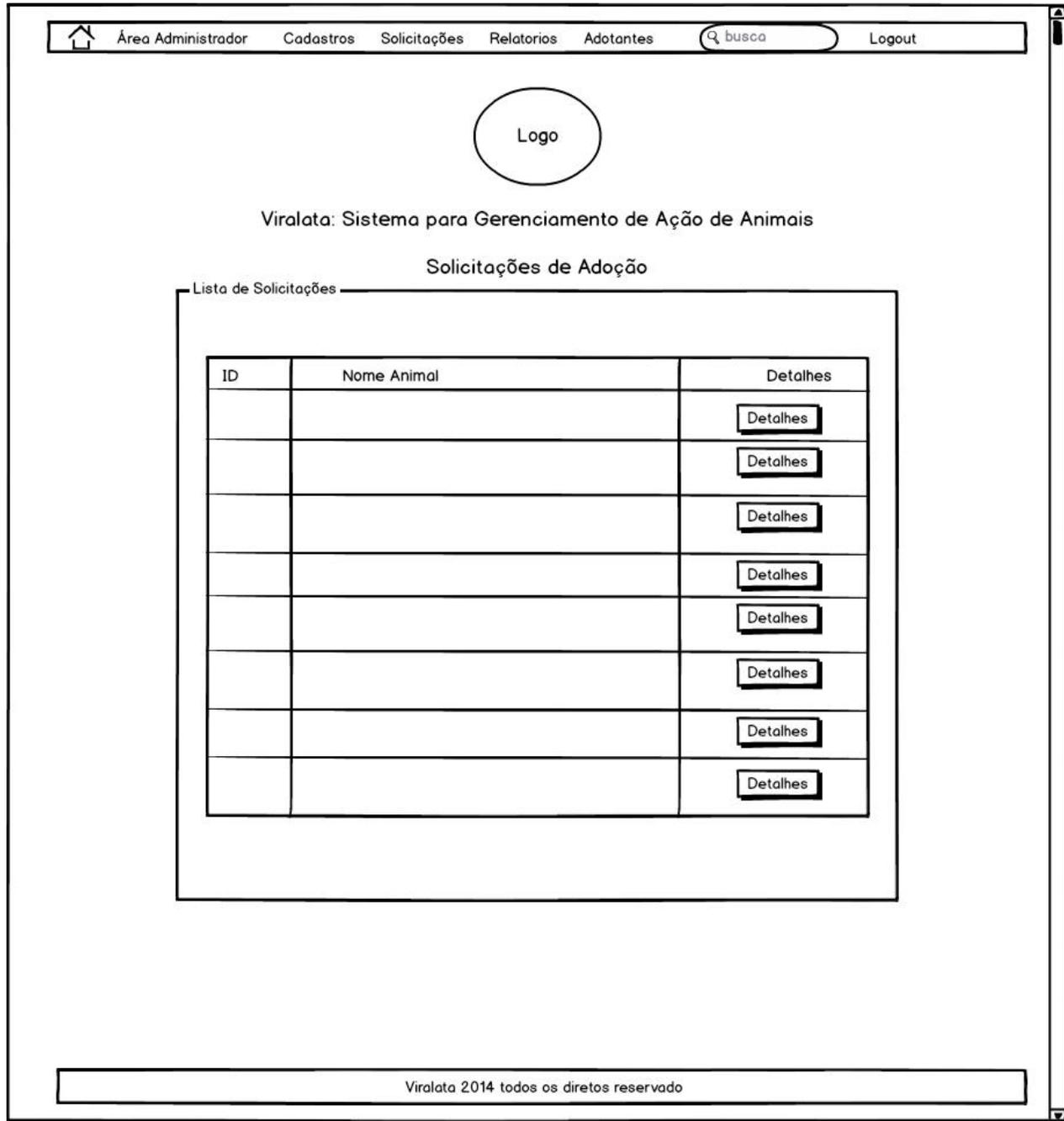
Atualizar Perfil

Viraldia 2014 todos os direitos reservados

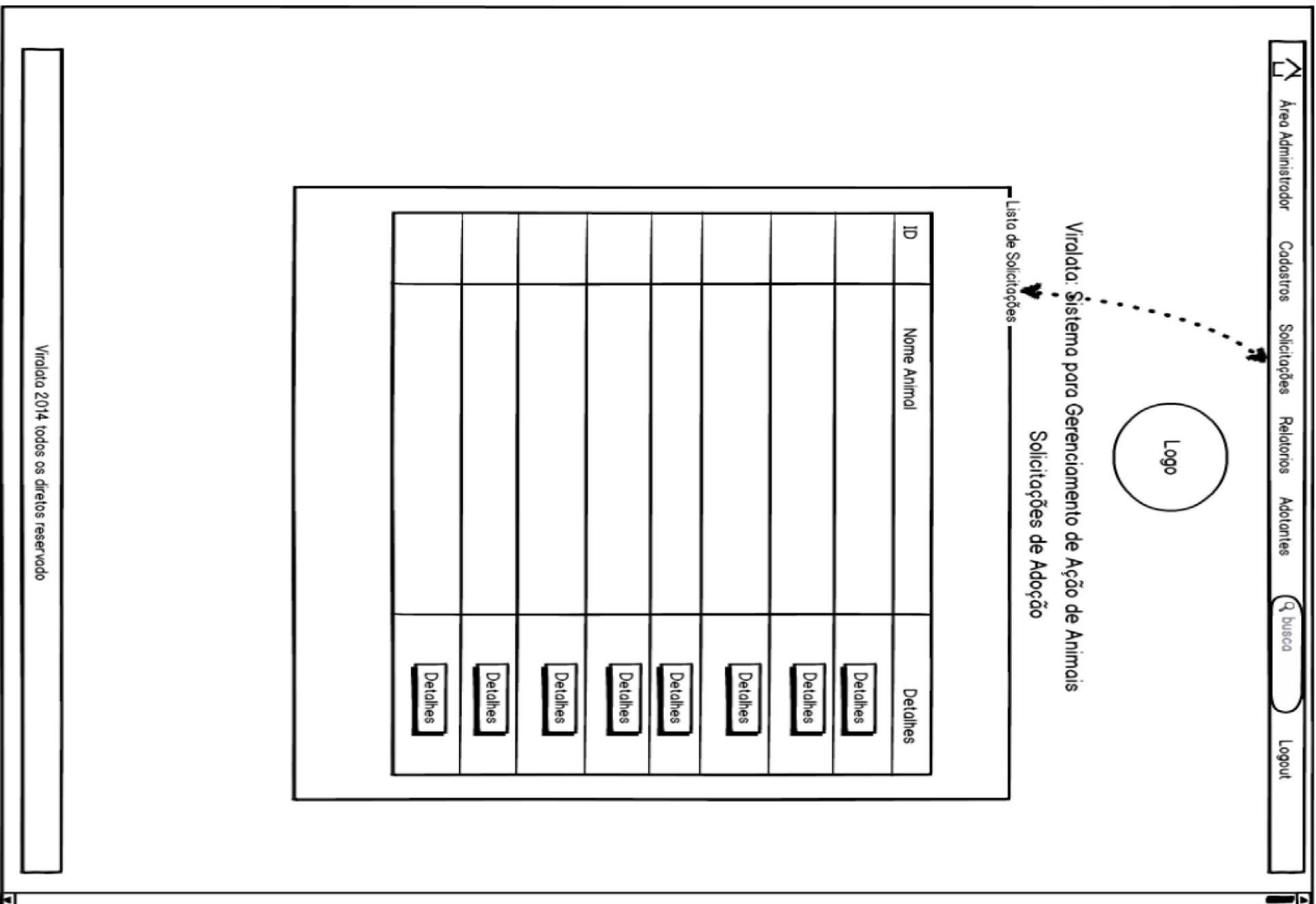
PROTÓTIPO TELA PERFIL ADOTANTE



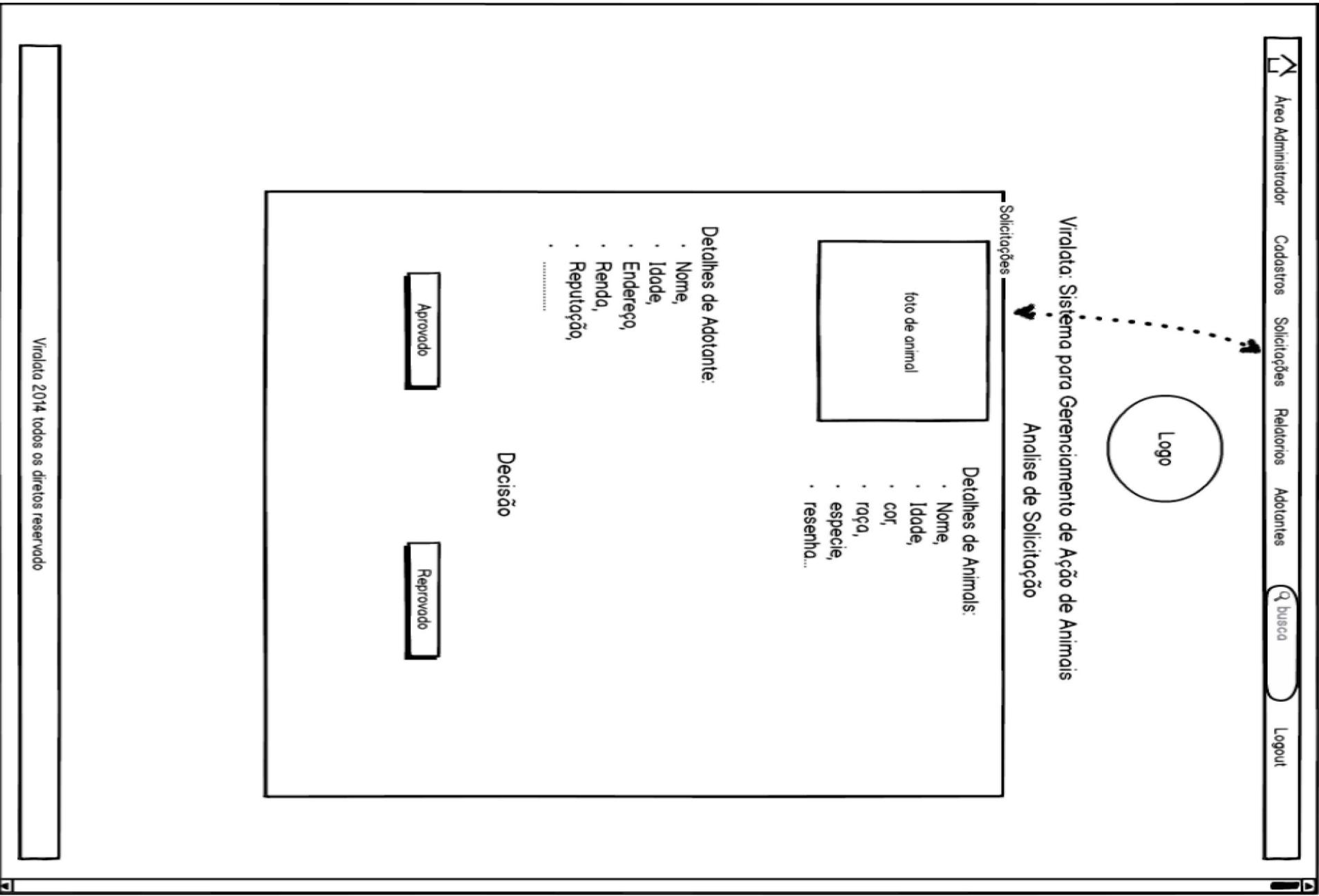
PROTÓTIPO TELA BUSCA ANIMAL ADOTANTE



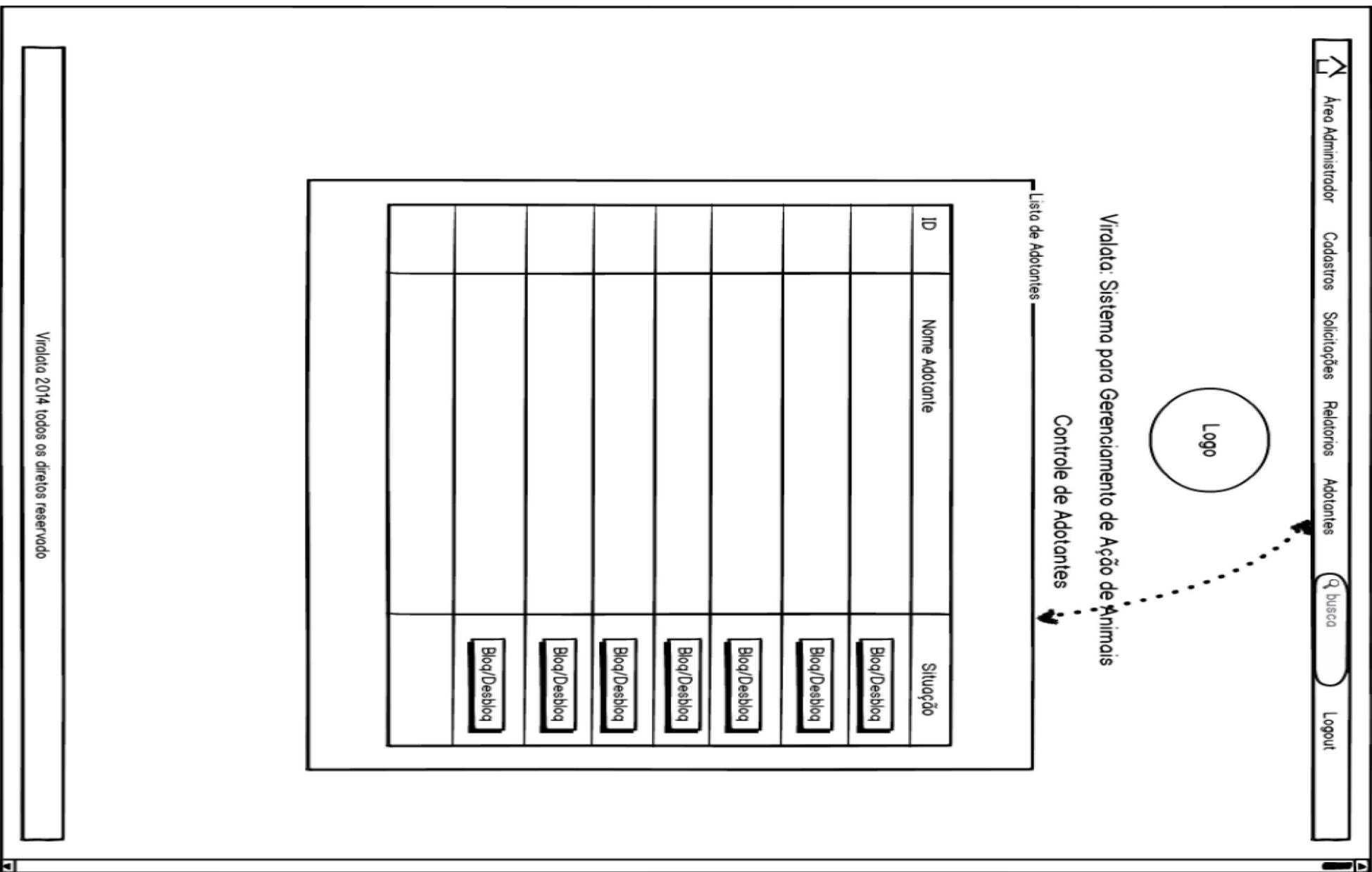
PROTÓTIPO TELA HOME ADMINISTRADOR



PROTÓTIPO TELA SOLICITAÇÕES DE ADOÇÕES



PROTÓTIPO TELA ANÁLISE DE SOLICITAÇÕES DE ADOÇÕES



PROTÓTIPO TELA BLOQUEIA OU DESBLOQUEIA ADOTANTE

Area Administrador Cadastros Solicitações Relatórios Adotantes Logout

Sub-Menu Cadastro de Animais

Logo

Viralata: Sistema para Gerenciamento de Ação de Animais

Cadastro de Animais

Dados de Animal

Nome Animal:

Idade:

Raça:

Especie: ▼

Cor:

Tamanho:

Resenha:

Localidade:

Pelagem:

Porte:

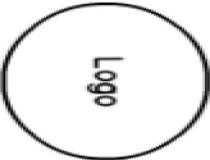
Deformidade: ▼

Preview

Viralata 2014 todos os direitos reservados

PROTÓTIPO TELA CADASTRO DE ANIMAIS

Área Administrador Cadastros Solicitações Relatórios Adotantes busca Logout



Logo

Viralata: Sistema para Gerenciamento de Ação de Animais

Cadastro de Locais

Dados de Locais

Nome Local:

Endereço:

Longitude:

Latitude:

Viralata 2014 todos os direitos reservados

PROTÓTIPO TELA CADASTRO DE POSSÍVEIS LOCAIS

Área Administrador Cadastros Solicitações Relatorios Adotantes Q busca Logout

Sub-Menu Cadastro de Feiras

Logo

Viralata: Sistema para Gerenciamento de Ação de Animais

Cadastro de Feiras

Dados de Feiras

Local

Nome Feira:

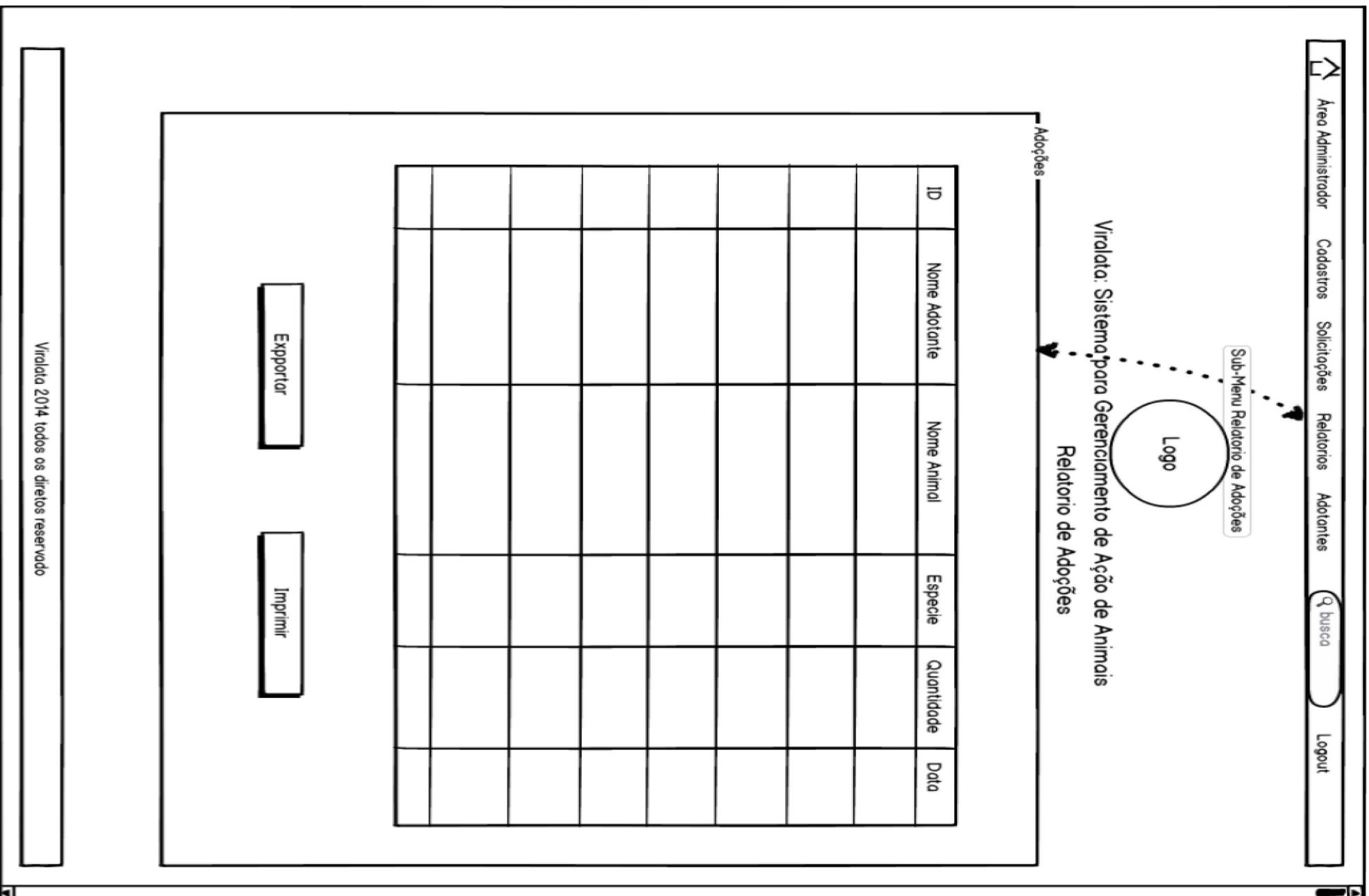
Numero Animais

Numero de Animais Adotados

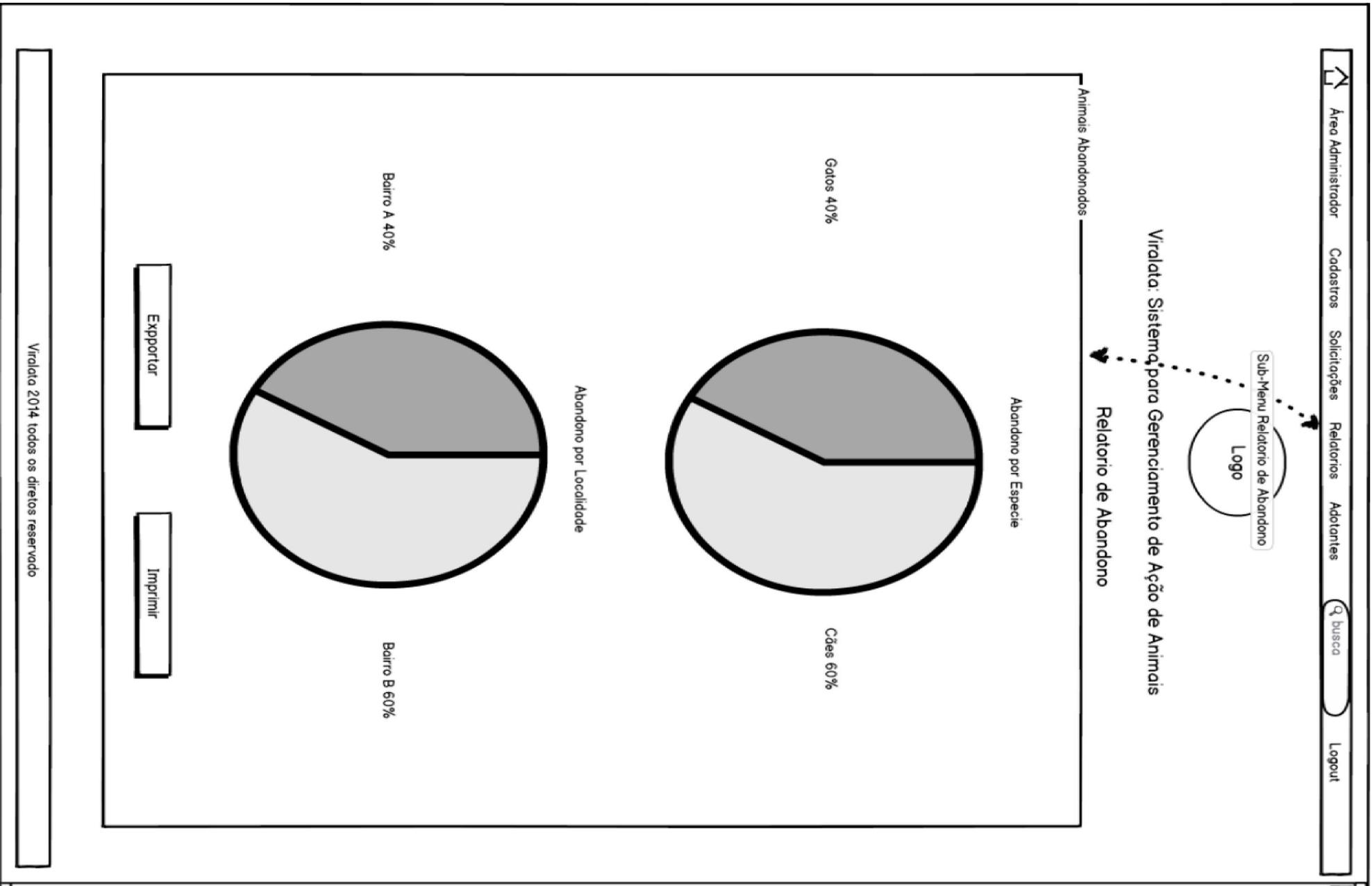
Organizadora

Viralata 2014 todos os direitos reservados

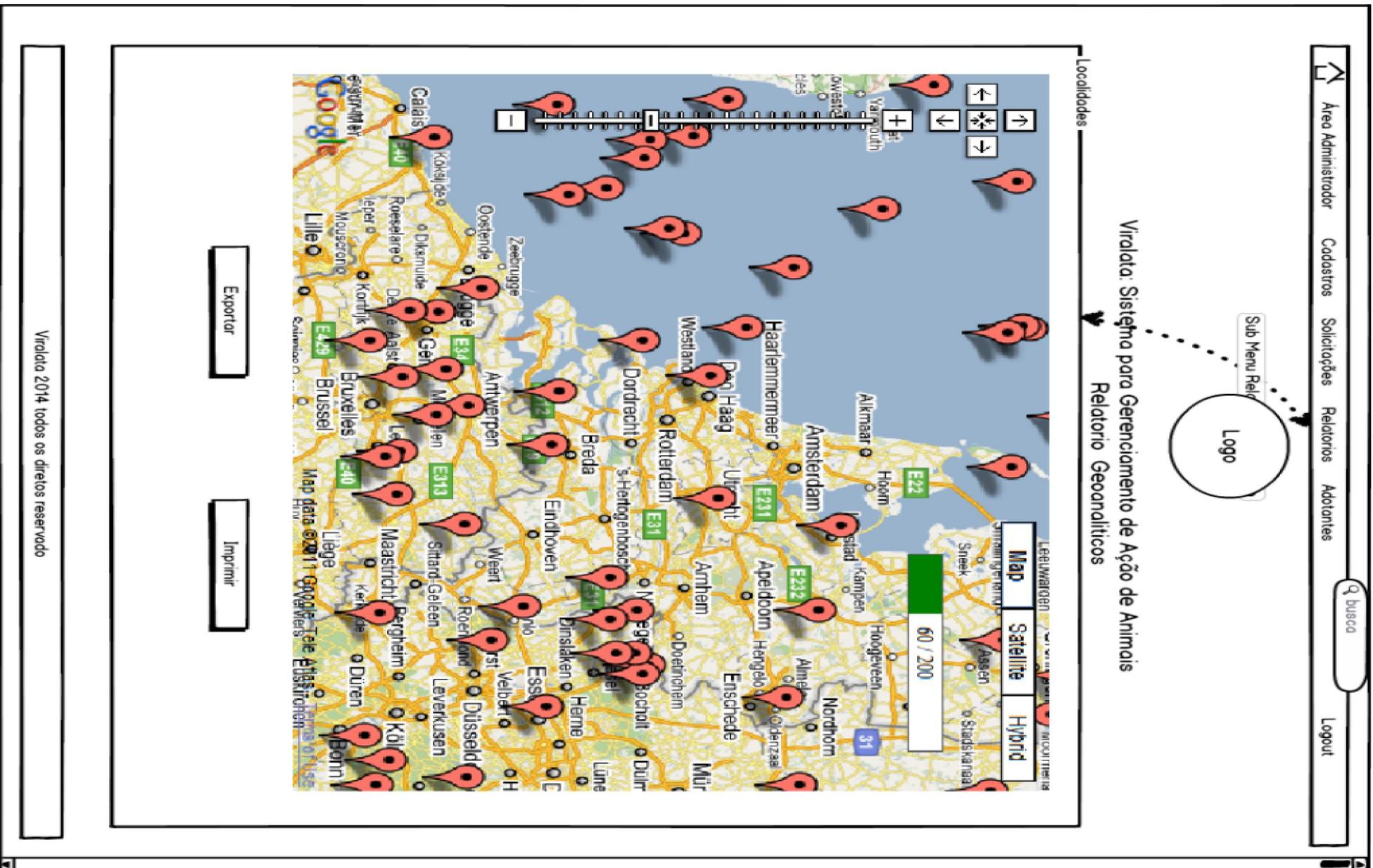
PROTÓTIPO TELA CADASTRO DE FEIRAS



PROTÓTIPO TELA RELATÓRIO DE ADOÇÕES



PROTÓTIPO TELA RELATÓRIO DE ABANDONO



PROTÓTIPO TELA RELATÓRIO GEOANALÍTICO