



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

SISTEMA DE GERENCIAMENTO DE PECUÁRIA COM ÊNFASE NO MERCADO BOVINO

TATIANE GOMES DE ALVARENGA

ASSIS
2014



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

SISTEMA DE GERENCIAMENTO DE PECUÁRIA COM ÊNFASE NO MERCADO BOVINO

Trabalho apresentado para a obtenção de conclusão do curso de Análise e Desenvolvimento de Sistema do Instituto Municipal de Ensino Superior de Assis - IMESA e a Fundação Educacional do Município de Assis – FEMA.

Orientando: Tatiane Gomes de Alvarenga

Orientador: Prof. Dr. Luiz Carlos Begosso

ASSIS
2014

Alvarenga, Tatiane Gomes de.

A471p Sistema de Gerenciamento de Pecuária com ênfase no mercado Bovino / Tatiane Gomes de Alvarenga. – Assis, 2014. 56f.

Trabalho de Conclusão de Curso (Tecnólogo – Análise e Desenvolvimento de Sistemas)
– Fundação Educacional do Município de Assis.

Orientador: Prof. Dr. Luiz Carlos Begosso

Banca: Prof. Ms. Felipe Alexandre Cardoso Pazinatto

1. Sistema Pecuário. 2. Android. 3. Java Web. I Alvarenga, Tatiane Gomes de. II. Título.
III. Assis – Fundação Educacional do Município de Assis.

**À Deus e aos meus queridos pais
e noivo Igor sem os quais não
teria chegado até aqui.**

AGRADECIMENTOS

À Deus primeiramente, pela minha vida e tudo que dela faz parte.

Aos meus pais, Célia e Jorge que fizeram de tudo para que eu alcance meus ideais e por todo o amor e cuidado ao longo da minha vida.

Ao meu noivo Igor por estar sempre do meu lado, sempre me incentivando a buscar o melhor e me apoiando em todas as minhas escolhas. Por não medir esforços para me ajudar no que fosse preciso e por toda a compreensão nesta etapa de minha vida.

Ao meu orientador Prof. Dr. Luiz Carlos Begosso, pela dedicação, paciência e tempo despendido para orientar minhas dúvidas.

Aos meus amigos Ariane, Kleber, Jauri e Josiane, por toda a ajuda, companheirismo e amizade nesta segunda jornada acadêmica.

À todos professores e colegas do curso pelos três anos de crescimento na minha vida.

E a todas as pessoas que de alguma forma contribuíram para que eu pudesse chegar até aqui.

*“Todas as vitórias ocultam uma abdicação”.
(Simone de Beauvoir)*

RESUMO

Trata-se do desenvolvimento de um sistema de Gerenciamento de Pecuária direcionado ao setor de bovinos com a proposta de contribuir para o controle de atividades, produções e informações geradas pelo setor pecuário. Para tanto, foi implementado um sistema web, responsável pelo gerenciamento de cadastros, visualização de apontamentos realizado no campo e geração de relatórios utilizando a linguagem Java na plataforma Java EE. Também foi desenvolvido na plataforma mobile o aplicativo com a finalidade de realizar apontamentos em campo, usando o Sistema Operacional Android. Para a comunicação entre ambas as plataformas foi implementado um webservice que é executado no servidor de aplicações ApacheTomCat, utilizando a linguagem XML como padronização para a troca de informações.

Palavras-chave: Sistema Pecuário. Android. Java Web

ABSTRACT

This is a project to implement a system of Farming system Management directed to the bovine sector with the proposal to contribute to the control of activities, production and information generated by the livestock sector. Thus, a web system, responsible for managing records and generating reports using the Java EE language was implemented in mobile platform, which has the purpose of making notes in the field, developed on Android. The communication between both platforms webservice using Tomcat was used as standardized XML language for the exchange of information. It was concluded that the project planning and development of diagrams is fundamental to understanding the use cases that make up the system and contributing optimizes the implementation of a system. The use of web server contributes to a better interaction, communication between different platforms, in addition to adding dynamism and security in information exchange.

Keywords: Farming system. Android. Java Web

LISTA DE ILUSTRAÇÕES

Figura 1 - Estrutura do Android.....	19
Figura 2 - EAP do Sistema Agropecuário para Gado.....	21
Figura 3 - Diagrama de caso de uso.....	23
Figura 4 - Ilustra o Caso de Uso- Cadastrar Propriedade.....	24
Figura 5 - Ilustra o Caso de Uso- Cadastrar Usuário.....	25
Figura 6 - Ilustra o Caso de Uso- Cadastrar Cidade.....	26
Figura 7 - Ilustra o Caso de Uso- Cadastrar Local.....	27
Figura 8 - Ilustra o Caso de Uso- Cadastrar Animal.....	29
Figura 9 - Ilustra o Caso de Uso- Gerar Relatório de Transferência.....	30
Figura 10 - Ilustra o Caso de Uso- Gerar Relatório de Situação Atual do Local.....	31
Figura 11 - Ilustra o Caso de Uso- Gerar Relatório de Manejo.....	32
Figura 12 - Ilustra o Caso de Uso- Gerar Relatório de Baixa de Animal.....	32
Figura 13 - Ilustra o Caso de Uso- Gerar Relatório de Morte por Animal.....	33
Figura 14 - Ilustra o Caso de Uso- Gerar Relatório de consumo de Ração por Animal.....	34
Figura 15 - Ilustra o Caso de Uso- Gerar Relatório de Vacinação por Animal.....	34
Figura 16 - Ilustra o Caso de Uso- Gerar Relatório de Quantidade de Animal por Propriedade.....	35
Figura 17 - Ilustra o Caso de Uso- Apontar Manutenção Local.....	36
Figura 18 - Ilustra o Caso de Uso- Apontar Manejo.....	37
Figura 19 - Ilustra o Caso de Uso- Efetuar Transferência do Animal.....	38
Figura 20 - Diagrama de Classe.....	40
Figura 21 – Diagrama Entidade Relacionamento.....	41
Figura 22– Diagrama de Atividades.....	42
Figura 23– Estrutura do sistema.....	43
Figura 24 – Tela Principal do PecBovinos Web.....	44
Figura 25 – Tela de Cadastro Causa Morte.....	45
Figura 26– Tela de Configuração de conexão.....	46
Figura 27 – Tela Principal do PecBovinos Mobile.....	51
Figura 28 – Tela de Apontamentos de Manejo.....	52
Figura 29 – Tela de Apontamento de Vacina por Animal.....	53
Figura 30 – Tela de Apontamento de Alimentação.....	54
Figura 31 – Tela de Apontamento de Baixa de Animais.....	55
Figura 32– Tela de Sincronização de Cadastros.....	56

LISTA DE ABREVIATURAS E SIGLAS

API - Application Programming Interface

HTTP - HypertText Transfer Protocol

HTML - HyperText Markup Language

JIT - Just in Time

JVM - Java Virtual Machine

J2SE - Java 2 Standard Edition

OHA - Open Handset Alliance

SDK - Software Development Kit

SGBD- Sistema de Gerenciamento de Banco de Dados

TI – Tecnologia da Informação

XML – eXtensible Markup Language

SUMÁRIO

1 INTRODUÇÃO.....	11
1.1 Objetivos.....	12
1.2 Justificativas.....	12
1.3 Público.....	13
 2 MÉTODO DE DESENVOLVIMENTO.....	 14
2.1 Metodologia de desenvolvimento do sistema.....	14
2.2 Java EE.....	14
2.3 JBoss.....	16
2.4 Android.....	18
2.5 SQLite.....	19
2.6 PostgreSQL.....	20
 3 ANÁLISE E PROJETO DO SISTEMA.....	 21
3.1 Estrutura Analítica do Projeto (EAP).....	21
3.2 Diagrama de Casos de Uso	22
3.2 Narrativas de Caso de Uso.....	24
3.4 Diagrama de Classe.....	39
3.5 Diagrama E-R.....	41
3.6 Diagrama de Atividades.....	42
3.7 Arquitetura do Sistema.....	43
 4 TELAS DO SISTEMA.....	 44
4.1 Sistema Web.....	44
4.2 Sistema Mobile.....	45
 5 CONCLUSÃO.....	 47
 REFERÊNCIAS.....	 48
 APÊNCICE.....	 50
APÊNDICE A.....	51

1 INTRODUÇÃO

A Pecuária é considerada no Brasil uma das atividades de produção mais importante no que diz respeito à produção de carne bovina e leite, ovos, carne de frango e de suínos. Este setor representa cerca de 25% do PIB Brasileiro além de gerar empregos formais direta ou indiretamente. (TAMASSIA, 2013)

O atual cenário socioeconômico vem impondo significativas mudanças e transformações na cadeia agroindustrial pressionando-a a assumir característica empresariais. Sendo assim as organizações tendem a se tornarem mais competitivas a partir do momento que passam a utilizar tecnologia em seu processo de produção e comercialização. (MACHADO; NANTES, 2011)

A utilização de softwares desenvolvidos especificamente para a área de pecuária caracteriza-se como uma importante ferramenta para controle da produção e comercialização, auxiliando na tomada de decisão e agregando maior qualidade para fins de comércio.

Assim, observa-se a necessidade de se desenvolver um sistema destinado a este mercado econômico visto a tamanha importância que o mesmo representa para o país.

Dentro do mercado da pecuária, a bovinocultura de corte tem se destacado na economia nacional assumindo posição de liderança no mercado mundial de carnes. O Brasil possui o maior rebanho comercial do mundo além de ser o segundo maior produtor mundial de carne bovina. A partir de 2003 passou a ser o primeiro exportador mundial, com destaque tanto no comércio de carnes frescas como no de industrializadas. (EMBRAPA, 2007)

A união dos avanços tecnológicos pode representar ganhos significativos ao produtor visto que o mesmo pode controlar as atividades em campo com o auxílio de dispositivos móveis como celular ou mesmo *tablet*, utilizando tecnologia *mobile*.

Com base neste contexto o presente trabalho propõe o desenvolvimento de um sistema para atuar em negócios da Pecuária com ênfase na área bovina.

Para melhor compreensão, o projeto apresenta os objetivos para o desenvolvimento do sistema, a justificativa bem como o público alvo. Apresenta

também a metodologia que foi utilizada para alcançar o objetivo proposto, a estrutura do desenvolvimento do sistema e o cronograma.

1.2 Objetivos

O presente trabalho tem como Objetivo Geral, desenvolver um software de gestão pecuária de bovinos. Para tanto tem como objetivos específicos implementar em plataforma *web* o controle da criação do animal e manejo, cadastro de animais, locais e propriedades, bem como gerar relatórios gerenciais. Em plataforma *mobile* com comunicação utilizando *webservices*, foi implementados os apontamentos a transferência de animais, manutenção dos locais da propriedade, e manejo.

1.3 Justificativa

No cenário atual torna-se imprescindível a utilização das tecnologias para realizar todo o controle de informações geradas por uma empresa, independente de qual seja o setor em que está inserida.

Segundo Machado e Nantes (2011), o aumento da utilização da tecnologia na agropecuária pode causar impactos positivos, pois substitui a força do trabalho e aumento no controle das atividades. Desta forma reduz a incerteza e riscos, bem como aumenta o ingresso dos mesmos ao mercado, pois facilita o acesso à informação.

Ainda segundo os autores:

A adoção da TI é particularmente importante na pecuária, na qual os sistemas informatizados armazenam e fornecem informações sobre o controle de matrizes, cruzamento de animais, inseminações artificiais, aspectos nutricionais e sanitários de cada animal. O gerenciamento dessas informações pode ser realizado por meio de *softwares* específicos. (Machado; Nantes, 2011, p.557)

É nesta constatação que se justifica o presente trabalho, na contribuição para o controle de atividades, produção e informações geradas pelo setor pecuário.

Na utilização de tecnologias existentes atualmente para a construção de um sistema que auxilie o produtor no manejo do animal, quanto ao trato com aplicações de vacinas e adequação à vigilância sanitária.

1.4 Público Alvo

O presente trabalho tem como público alvo empresas e fazendas pecuárias de pequeno e médio porte que lidam com a criação de gado para venda ou corte.

2 MÉTODO DE DESENVOLVIMENTO

Este capítulo apresenta as tecnologias que foram usadas para desenvolver o sistema de gerenciamento de pecuária bovina.

2.1 Metodologia de desenvolvimento do sistema

A possibilidade de poder constatar todo o conhecimento teórico adquirido colocando em prática através de métodos, justifica a real finalidade da realização de pesquisas. Levando em conta os objetivos propostos, o presente trabalho caracteriza-se como metodologia de pesquisa exploratória.

Neste sentido, com a finalidade de alcançar os objetivos propostos, foi desenvolvido um sistema web em linguagem Java em plataformas Java EE e uma aplicação *mobile* em Android. Para persistência dos dados, foi utilizado o SGBD Postgres SQL para a aplicação web e na plataforma Android, SQLite.

Para a troca de informações entre ambas as plataformas foi utilizado um servidor Web, neste caso a escolha do servidor de aplicação foi o Apache Tomcat.

A fim de padronizar a estrutura das informações de entrega e recebimento, foi utilizada a linguagem XML para a comunicação entre *client* e *server*.

Estas informações são enviadas e recebidas através do protocolo de comunicação HTTP.

2.2 Java EE

A Linguagem Java teve início em 1991 quando a Green, pequeno grupo de projeto da Microsystems, ficou encarregada de criar uma nova geração de computadores portáteis com capacidade de comunicar-se entre si. Para tanto, havia necessidade de criar uma nova plataforma que estivesse de acordo com a ideia de portabilidade em diferentes tipos de aparelhos, optando-se inicialmente pelo uso da linguagem de programação C++, pois além das características, a priori, semelhantes a que se pretendia utilizar, havia também a experiência em desenvolvimento de produtos com esta linguagem por parte da equipe. Contudo, perceberam que a

linguagem C++ não permitia a facilitação da realização do que o grupo pretendia alcançar. (JANDL JUNIOR,1999)

Ainda segundo o mesmo autor, optou-se pela criação de uma nova linguagem de programação que tivesse tudo o que necessitava, além de ser portátil, simples e de fácil programação. Foi então que sobre a coordenação de James Gosling surgiu a linguagem interpretada Oak, batizada com este nome devido a existência de um carvalho, um tipo de árvore que podia ser vista do escritório de Gosling.

Naquela época o mercado de eletrônicos inteligentes para o consumo pessoal não estava se desenvolvendo como visavam, o que a princípio, ameaçou o futuro do projeto. Porém com a explosão da *Word Wide Web* , em meados de 1993, surgiu a oportunidade de utilizar a linguagem agregando interatividade e animações às paginas de Web, impulsionando assim, o projeto. (DEITEL, H; DEITEL, P, 2006)

Devidos os problemas com *copyrights*, a Oak foi rebatizada de Java, nome de uma cafeteria local que o grupo visitara. E foi com a criação de um navegador, apresentado formalmente como *Hotjava* e linguagem Java no *SunWorld'95*, que a linguagem como solução explodiu. Foi então que o Netscape corp. lançou uma nova versão do navegador com a capacidade de efetuar downloads e execução de pequenas aplicações Java. Então, quando a Sun resolve disponibilizar para o grupo de desenvolvedores, a linguagem Java gratuitamente, surge então a Java Developer's kit 1.0 (JDK 1.0), e desde então vem lançando melhorias para o desenvolvimento de aplicações gráficas e distribuídas. (JANDL JUNIOR,1999)

Jandl Junior (1999) define Java como a linguagem de programação mais utilizada e que provocou grande entusiasmo em profissionais da área de desenvolvimento de software. O motivo deste entusiasmo está relacionado ao grande projeto para criação desta linguagem de programação que consiste de uma pesquisa que analisou e transformou não somente em uma linguagem, mas em um ambiente todo de desenvolvimento. Ainda segundo autor acima referido o Java exhibe facilidades pela linguagem orientada a objetos, portabilidade do código produzido, características de segurança que a plataforma oferece e por sua integração com outros ambientes incluindo a Internet.

Um sistema construído em Java é composto por classes que, por sua vez, tem responsabilidades entre si (MOREIRA NETO, 2006).

Segundo Jandl Junior (1999) as principais características que destacam o Java são:

Orientada à Objetos – todos os seus componentes são orientados objetos, desta forma suas características são mecanismos de abstração, encapsulamento e hereditariedade.

Independente de Plataforma – Os programas desenvolvidos em Java são compilados em *bytecodes*, uma forma intermediária de código onde utiliza-se instruções e tipos primitivos com tamanho fixo. O *Java Virtual Machine* é um interpretador de *bytecodes*.

Sem Ponteiros – O Java não permite que os desenvolvedores manipulem o endereço de memória, trabalhando assim com a recuperação automática de memória de objetos não mais referenciados pelo programa.

Performance – Com a *Java Virtual Machine* (JVM) é possível compilar o código nativo durante a execução e assim otimizando, isto é, a maioria das JVMs possui compiladores *Just in Time*(JIT) que compilam os *bytecodes* para código que melhora significativamente a performance de programas Java.

Segurança – a linguagem Java possui mecanismos de segurança que podem evitar qualquer operação no sistema de arquivos da máquina alvo. O mecanismo é flexível para determinar se uma applet é considerada segura.

Permite *Multithreading* – capaz de trabalhar com diferentes rotinas ao mesmo tempo de execução. Estas rotinas são definidas como *thread*.

2.3 Apache Tomcat

Dentre os servidores de aplicações existentes, o Tomcat, caracteriza-se como um robusto servidor para aplicações Java para web, é um software livre e desenvolvido como código aberto dentro do conceituado projeto Apache Jakarta e oficialmente endossado pela Sun como a Implementação de Referência para as tecnologias *Java Servlet* e *JavaServer Pages*.

Segundo Alecrim (2006), o servidor Apache surgiu no *National Center of Supercomputing Applications* por meio de um trabalho de Rob McCool. Ao sair da NCSA, McCool parou de trabalhar no software e então várias pessoas e grupos passaram a adaptar o servidor Web às suas necessidades. No entanto, foram Brian Behlendorf e Cliff Skolnick os principais responsáveis pela retomada do projeto, contando logo em seguida com o apoio de Brandon Long e Beth Frank na qual, mais tarde, juntaram-se ao Apache Group.

Neste ponto surge a Fundação Apache, onde Brian Behlendorf e Cliff Skolnick passaram a controlar as adições de recursos ou correções chamados de *patches* por meio de uma lista de discussão. Dessa lista originou-se um grupo, constituído inicialmente de oito pessoas. Apesar do nome da fundação ter sido baseada a partir de um trocadilho com a expressão em inglês: a *patchy*, os desenvolvedores do servidor, afirmam que o nome é baseado numa tribo americana de nome Apache.

Ainda segundo o autor, o servidor Apache teve sua primeira versão oficial em 1995. Após esta versão, houve várias mudanças no projeto sendo uma das mais importantes, o *Shambhala*, sendo uma arquitetura para o Apache desenvolvida por Robert Thau e que foi incorporada definitivamente ao servidor, tirando de cena a base existente até então. Após uma série de aperfeiçoamentos, testes e de uma documentação nova feita por David Robinson, o Apache 1.0 foi lançado em dezembro de 1995.

Segundo DevMidia (2014) por estar escrito em Java , o mesmo , necessita que a versão Java 2 Standard Edition (J2SE) esteja instalada no mesmo computador onde ele será executado. No entanto, não basta ter a versão *runtime* de Java instalada, pois o Tomcat necessita compilar os programas escritos em Java. O projeto Jakarta da Fundação Apache, do qual o subprojeto Tomcat é o representante, tem como objetivo o desenvolvimento de soluções código aberto baseadas na plataforma Java.

O desenvolvimento de uma típica aplicação web a ser executada pelo Tomcat implica no domínio das seguintes linguagens:

- **Java:** Todos os algoritmos da aplicação devem ser escritos em Java.
- **HTML:** A interface da aplicação é construída na forma de páginas escritas em HTML e visualizadas através do browser.
- **XML:** Qualquer aspecto relacionado à configuração da aplicação deve ser expresso por meio da linguagem XML em um arquivo chamado web.xml. Os dados de configuração podem ser usados tanto pelo Tomcat como pela aplicação.

2.4 Android

Com o crescimento do mercado de celulares, empresas e desenvolvedores buscam por uma plataforma moderna e ágil, para o desenvolvimento de aplicações corporativas que auxiliem em seus negócios e lucros, aplicações móveis que possam estar literalmente conectadas e online, sincronizando informações diretamente de um servidor confiável da empresa. (LECHETA, 2010)

Ainda segundo o Lecheta (2010), a fim de atender este nicho de mercado, a Google desenvolveu o Android o qual consiste em uma nova plataforma de desenvolvimento para aplicações móveis baseada no sistema operacional Linux que, além de já vir com diversas aplicações já instaladas, é um poderoso, ousado e flexível ambiente de desenvolvimento.

O Android foi construído para que os desenvolvedores pudessem criar aplicações móveis permitindo obter total aproveitamento de um dispositivo portátil. É a primeira plataforma open source para dispositivos móveis juntamente com a Open Handset Alliance (OHA) tendo o pacote Software Development Kit (SDK) como kit que contem todas as ferramentas e Application Programming Interface (API) necessárias para o desenvolvimento utilizando a linguagem Java. (PEREIRA, SILVA, 2009)

No Android não há uma diferença entre os aplicativos integrados e os aplicativos criados com o SDK, permitindo, portanto, escrever poderosos aplicativos que acessam os recursos disponíveis dos dispositivos. Neste sentido, o Android é um software que tira proveito de seu kernel Linux e na qual os desenvolvedores escrevem seus aplicativos em Java. A figura1 ilustra a estrutura do sistema Android (ALBLESON *et al.* , 2012):



Figura 1 – Estrutura do Android (Extraído de Albleson et al, 2012).

2.5 SQLite

O Android possui integração com o SQLite, o qual permite que uma aplicação possa ter um ou mais banco de dados. Caracteriza-se com um leve e poderoso banco de dados (LECHETA, 2009)

O SQLite é uma ferramenta simples e eficiente do Android que pode ser utilizada para navegar, armazenar os dados da aplicação em tabelas ou manipular comandos SQL no dispositivo. Funciona como um Sistema Gerenciador de Bancos de Dados que permite criar um arquivo, ler e escrever diretamente sobre o arquivo, o que o caracteriza como próprio servidor. (PEREIRA, SILVA, 2009)

Para Albleson *et al.* (2012, p.147)

O SQLite não possui todos os recursos de produtos cliente/servidor maiores, mas inclui tudo o que você precisa para armazenamento local de dados. Ao mesmo tempo, ele é rápido e relativamente fácil de trabalhar.

2.6 POSTGRESQL

Segundo a Scherer (2007), a implementação do POSTGRES iniciou-se em 1986 pelo professor Michael Stonebraker na Universidade da Califórnia em Berkeley. Primeiro foram definidos os modelos de dados e o sistema de regras do SGBD, logo depois foram surgindo várias versões até chegar a versão operacional em 1987. A quantidade de usuários dobrou durante o ano 1993, gerando uma grande dificuldade na manutenção do código do protótipo e assim a versão de Berkley terminou na versão 4.2.

Ainda segundo o mesmo autor, em 1994, foi adicionado um interpretador da linguagem SQL ao POSTGRES, junto com esta mudança também houve uma mudança em seu nome que passou para Postgres95, nesta versão houve um ganho de 30-50% no critério rapidez comparado com a versão do POSTGRES 4.2. No ano de 1996 percebeu-se que o nome Postgres95 não iria perdurar pelos próximos anos e então foi escolhido o novo nome de PostgreSQL.

De acordo com a Comunidade Brasileira de PostgreSQL (2007), o PostgreSQL é um poderoso software para gerenciamento de bancos de dados, agregando algumas funções dos SGBDs mais avançados, como por exemplo, os softwares proprietários Oracle ou SQL Server. O PostgreSQL é conhecido por sua robustez e extrema segurança e por ser um sistema open-source o qual possui uma comunidade de desenvolvedores do mundo inteiro.

Segundo Gonzaga (2007), a utilização do PostgreSQL tem ganhado cada vez mais espaço no mercado e é utilizado tanto em organizações públicas quanto organizações privadas. Um dos fatores que gerou um aumento significativo de utilização foi a sua inclusão na maioria das distribuições do Linux.

Um dos principais diferenciais do banco é seu sistema de armazenamento de dados, que difere totalmente dos bancos concorrentes. O PostgreSQL possui um sistema de armazenamento baseado em *cluster*, que é um diretório onde todos os arquivos do banco estão armazenados, tanto os arquivos dos dados, quanto os arquivos de configuração do banco.

3 ANÁLISE E PROJETO DO SISTEMA

3.1 Estrutura Analítica do Projeto (EAP)

Para melhor gerenciamento do projeto de criação de um Sistema Agropecuário para Gado, o mesmo foi subdividido hierarquicamente em pacotes e subpacotes de atividades conforme está ilustrado na Figura 2.



Figura 2- EAP do Sistema Agropecuário para Gado.

3.2 Diagrama de Casos de Uso

O Caso de Uso possibilita captar o comportamento pretendido do sistema, fornecer uma maneira de obter um consenso comum entre desenvolvedores e clientes finais e auxiliar na validação da arquitetura podendo verificar a dimensão do sistema a partir de sua evolução durante o desenvolvimento. (BOOCH; RUMBAUGH; JACOBSON, 2006)

Ainda segundo os Autores:

Um caso de uso é uma descrição de um conjunto de sequência de ações, inclusive variantes, que um sistema executa para produzir um resultado de valor observável por um Ator. Graficamente, o caso de uso é representado como uma elipse. (BOOCH; RUMBAUGH; JACOBSON, 2006, p.230)

A Figura 3 apresenta o caso de uso elaborado com base nos objetivos levantados para o sistema proposto, onde se pode observar a interação dos possíveis usuários do sistema, na qual os interiores das elipses contêm o nome de cada caso.

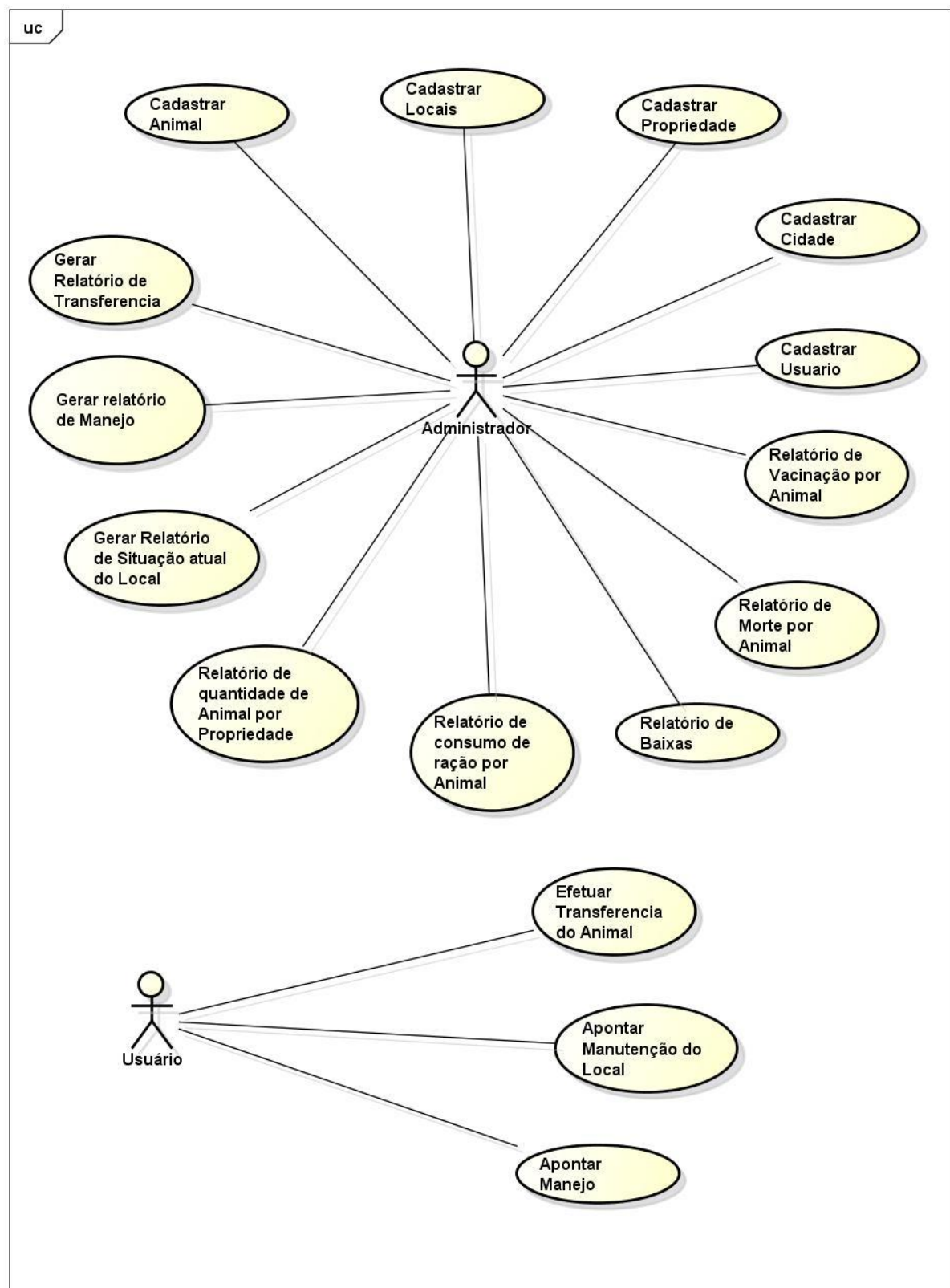


Figura 3- Diagrama de caso de uso

3.2 Narrativas de Caso de Uso

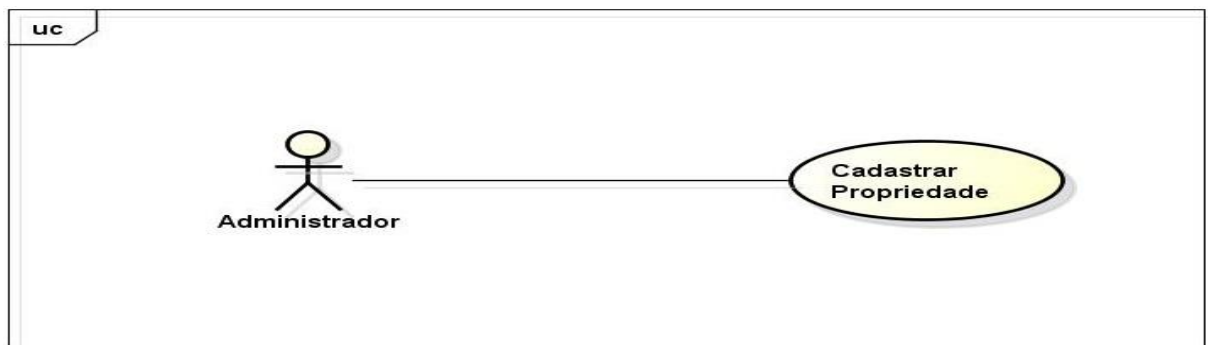


Figura 4 - Ilustra o Caso de Uso- Cadastrar Propriedade

Fluxo Principal

- a) O cliente seleciona a opção Cadastrar no Menu principal e em seguida seleciona a opção Cadastrar Propriedade. **[A1][A2][A3]**
- b) O sistema solicita as informações necessárias para cadastrar a propriedade.
- c) O cliente informa os dados e confirma a operação. **[E1]**
- e) O sistema confirma os dados e apresenta a mensagem de operação concluída.

Fluxo Alternativo

A1- Excluir Propriedade

- a. O usuário seleciona a opção Excluir Propriedade.
- b. O sistema solicita o código da propriedade a ser excluído.
- c. O sistema confirma os dados, exclui a Propriedade e retorna ao passo a do Fluxo Principal.

A2- Alterar Propriedade

- a. O usuário escolhe a opção Alterar Propriedade.
- b. O sistema solicita o código da Propriedade a ser alterado.
- c. O sistema confirma os dados, altera a Propriedade e retorna ao passo a do Fluxo Principal.

A3. Cancelar Operação

- a. O usuário cancela a operação de Cadastrar, podendo ou não informar algum dado.

- b. O sistema retorna ao passo a do Fluxo Principal.

Fluxo de Exceção

E1- Propriedade já cadastrada

- a. O sistema realiza a consulta e verifica que a Propriedade já está cadastrado
- b. O sistema emite a mensagem de Propriedade já cadastrado e retorna ao passo a do Fluxo Principal.

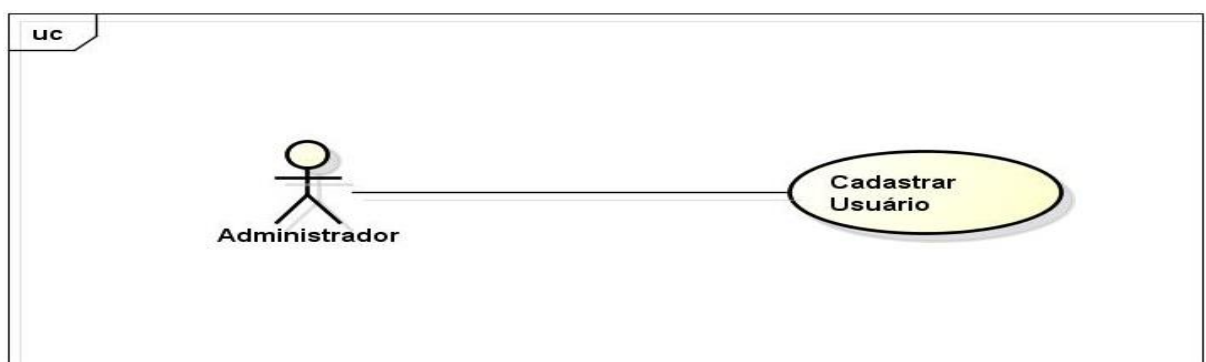


Figura 5 - Ilustra o Caso de Uso- Cadastrar Usuário

Fluxo Principal

- a) O cliente seleciona a opção Cadastrar no Menu principal e em seguida seleciona a opção Cadastrar Usuário. **[A1][A2][A3]**
- b) O sistema solicita as informações necessárias para cadastrar o Usuário.
- c) O cliente informa os dados e confirma a operação. **[E1]**
- e) O sistema confirma os dados e apresenta a mensagem de operação concluída.

Fluxo Alternativo

A1- Excluir Usuário

- a. O usuário seleciona a opção Excluir Usuário.
- b. O sistema solicita o código do Usuário a ser excluído.
- c. O sistema confirma os dados, exclui o Usuário e retorna ao passo a do Fluxo Principal.

A2- Alterar Usuário

- a. O usuário escolhe a opção Alterar Usuário.

- b. O sistema solicita o código do Usuário a ser alterado.
- c. O sistema confirma os dados, altera os dados do Usuário e retorna ao passo a do Fluxo Principal.

A3. Cancelar Operação

- a. O usuário cancela a operação de Cadastrar, podendo ou não informar algum dado.
- b. O sistema retorna ao passo a do Fluxo Principal.

Fluxo de Exceção

E1- Usuário já cadastrada

- a. O sistema realiza a consulta e verifica que o Usuário já está cadastrado
- b. O sistema emite a mensagem de Usuário já cadastrado e retorna ao passo a do Fluxo Principal.

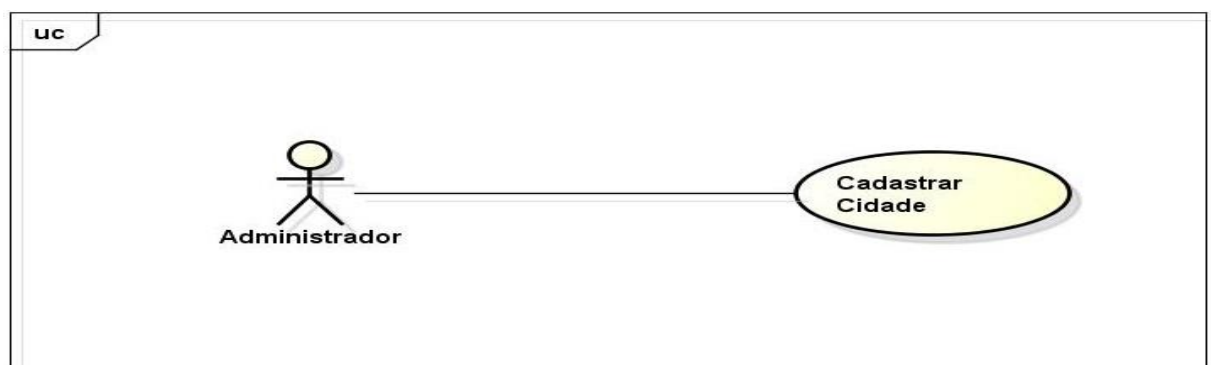


Figura 6 - Ilustra o Caso de Uso- Cadastrar Cidade

Fluxo Principal

- a) O cliente seleciona a opção Cadastrar no Menu principal e em seguida seleciona a opção Cadastrar Cidade. **[A1][A2][A3]**
- b) O sistema solicita as informações necessárias para cadastrar a Cidade.
- c) O cliente informa os dados e confirma a operação. **[E1]**
- e) O sistema confirma os dados e apresenta a mensagem de operação concluída.

Fluxo Alternativo

A1- Excluir Cidade

- a. O usuário seleciona a opção Excluir Cidade.

- b. O sistema solicita o código da Cidade a ser excluído.
- c. O sistema confirma os dados, exclui a Cidade e retorna ao passo a do Fluxo Principal.

A2- Alterar Cidade

- a. O usuário escolhe a opção Alterar Cidade.
- b. O sistema solicita o código da Cidade a ser alterado.
- c. O sistema confirma os dados, altera os dados da Cidade e retorna ao passo a do Fluxo Principal.

A3. Cancelar Operação

- a. O usuário cancela a operação de Cadastrar, podendo ou não informar algum dado.
- b. O sistema retorna ao passo a do Fluxo Principal.

Fluxo de Exceção

E1- Cidade já cadastrada

- a. O sistema realiza a consulta e verifica que a Cidade já está cadastrado
- b. O sistema emite a mensagem de Cidade já cadastrado e retorna ao passo a do Fluxo Principal.

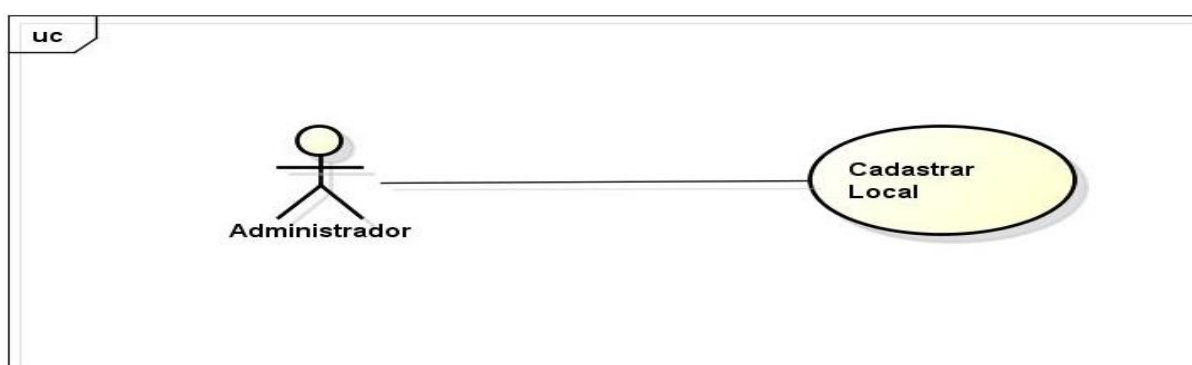


Figura 7 - Ilustra o Caso de Uso- Cadastrar Local

Fluxo Principal

- a) O cliente seleciona a opção Cadastrar no Menu principal e em seguida seleciona a opção Cadastrar Local.[A1][A2][A3]
- b) O sistema solicita as informações necessárias para cadastrar o Local.

- c) O cliente informa os dados e confirma a operação. **[E1]**
- e) O sistema confirma os dados e apresenta a mensagem de operação concluída.

Fluxo Alternativo

A1- Excluir Local

- a. O usuário seleciona a opção Excluir Local.
- b. O sistema solicita o código do Local a ser excluído.
- c. O sistema confirma os dados, exclui o Local e retorna ao passo a do Fluxo Principal.

A2- Alterar Local

- a. O usuário escolhe a opção Alterar Local.
- b. O sistema solicita o código do Local a ser alterado.
- c. O sistema confirma os dados, altera os dados do Local e retorna ao passo a do Fluxo Principal.

A3. Cancelar Operação

- a. O usuário cancela a operação de Cadastrar, podendo ou não informar algum dado.
- b. O sistema retorna ao passo a do Fluxo Principal.

Fluxo de Exceção

E1- Local já cadastrada

- a. O sistema realiza a consulta e verifica que o Local já está cadastrado
- b. O sistema emite a mensagem de Local já cadastrado e retorna ao passo a do Fluxo Principal.

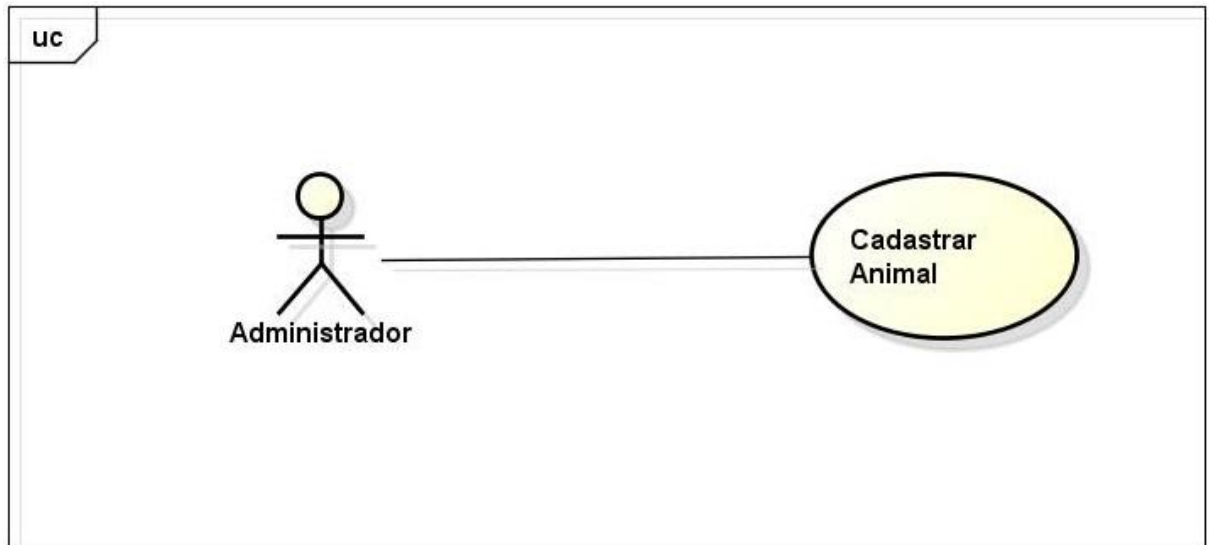


Figura 8 - Ilustra o Caso de Uso- Cadastrar Animal

Fluxo Principal

- a) O cliente seleciona a opção Cadastrar no Menu principal e em seguida seleciona a opção Cadastrar Animal. **[A1][A2][A3]**
- b) O sistema solicita as informações necessárias para cadastrar o Local.
- c) O cliente informa os dados e confirma a operação. **[E1]**
- e) O sistema confirma os dados e apresenta a mensagem de operação concluída.

Fluxo Alternativo

A1- Excluir Animal

- a. O usuário seleciona a opção Excluir Animal.
- b. O sistema solicita o código do Animal a ser excluído.
- c. O sistema confirma os dados, exclui o Animal e retorna ao passo a do Fluxo Principal.

A2- Alterar Animal

- a. O usuário escolhe a opção Alterar Animal.
- b. O sistema solicita o código do Animal a ser alterado.
- c. O sistema confirma os dados, altera os dados do Animal e retorna ao passo a do Fluxo Principal.

A3. Cancelar Operação

- a. O usuário cancela a operação de Cadastrar, podendo ou não informar algum dado.
- b. O sistema retorna ao passo a do Fluxo Principal.

Fluxo de Exceção

E1- Animal já cadastrada

- a. O sistema realiza a consulta e verifica que o Animal já está cadastrado
- b. O sistema emite a mensagem de Animal já cadastrado e retorna ao passo a do Fluxo Principal.

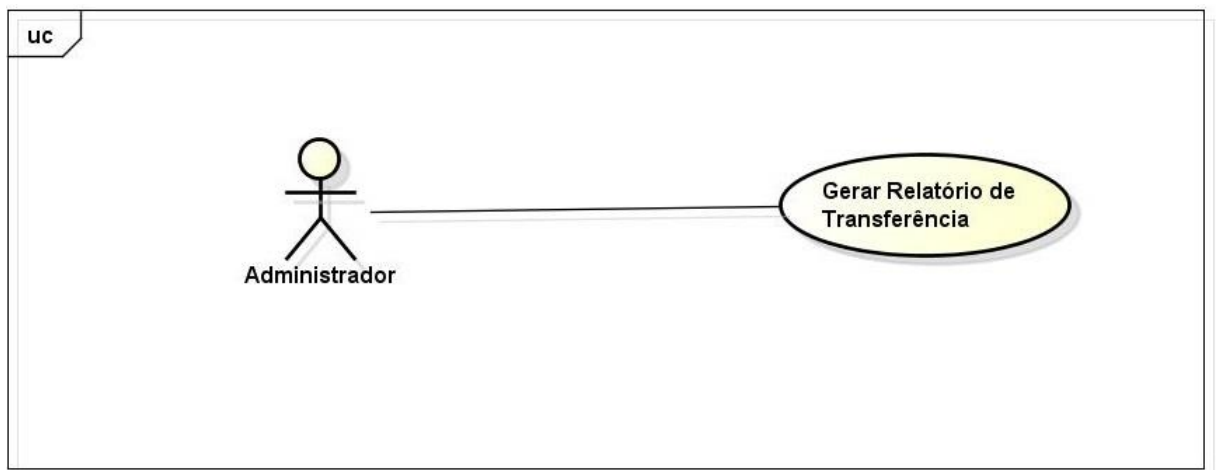


Figura 9 -Ilustra o Caso de Uso- Gerar Relatório de Transferência

Fluxo Principal

- a) O cliente seleciona a opção Relatório no Menu principal e em seguida seleciona a opção Relatório de Transferência.**[A1]**
- b) O sistema solicita informações para gerar o relatório.
- c) O cliente informa os dados e clica e Gerar relatório.
- e) O sistema emite o relatório na tela para o usuário.

Fluxo Alternativo

A1- Cancelar Relatório

- a. O usuário escolhe a opção Cancelar.
- b. O sistema volta ao passo a do Fluxo Principal.

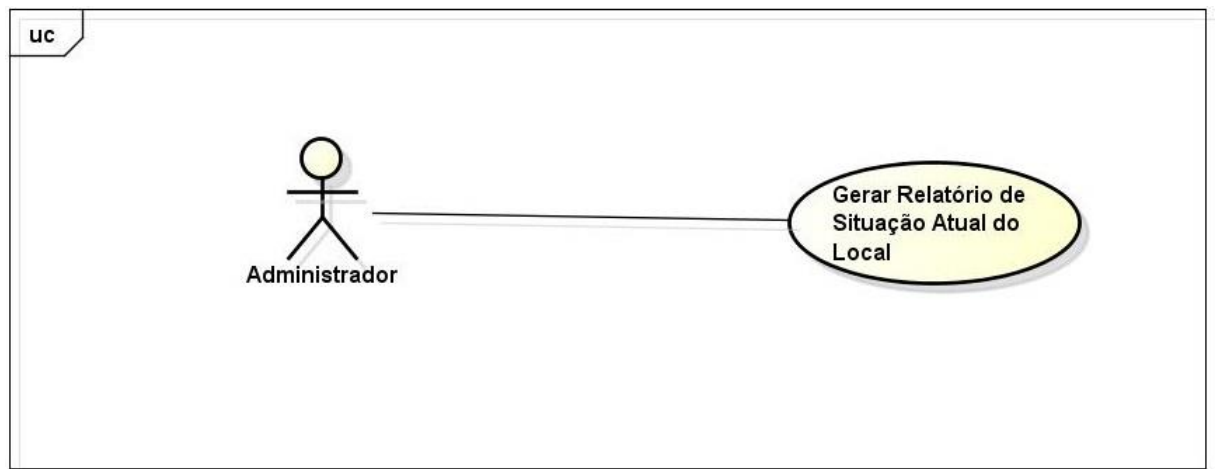


Figura 10 - Ilustra o Caso de Uso- Gerar Relatório de Situação Atual do Local

Fluxo Principal

- a) O cliente seleciona a opção Relatório no Menu principal e em seguida seleciona a opção Relatório de Situação Atual do Local **[A1]**
- b) O sistema solicita informações para gerar o relatório.
- c) O cliente informa os dados e clica e Gerar relatório.
- e) O sistema emite o relatório na tela para o usuário.

Fluxo Alternativo

A1- Cancelar Relatório

- c. O usuário escolhe a opção Cancelar.
- d. O sistema volta ao passo a do Fluxo Principal.

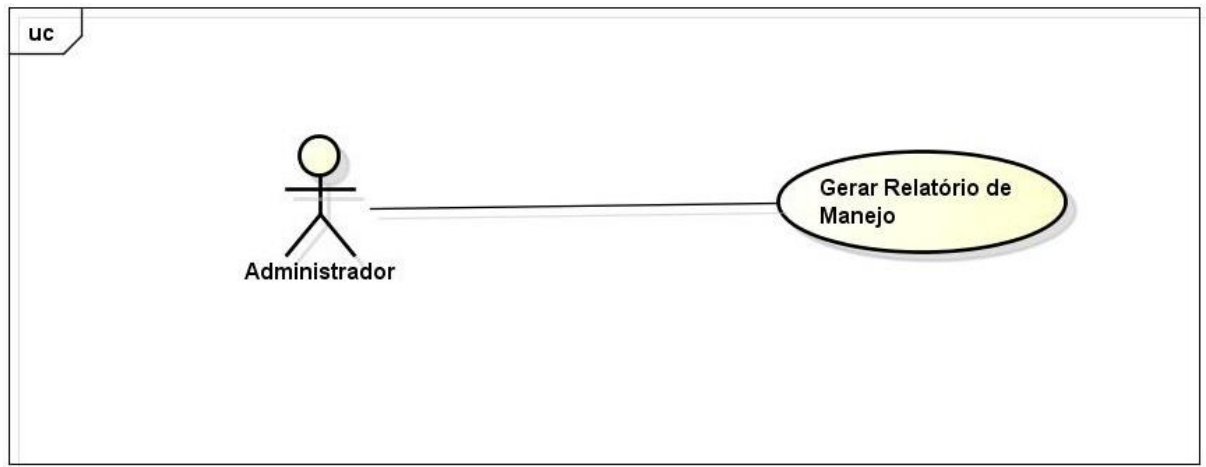


Figura 11 - Ilustra o Caso de Uso- Gerar Relatório de Manejo

Fluxo Principal

- a) O cliente seleciona a opção Relatório no Menu principal e em seguida seleciona a opção Relatório de Manejo **[A1]**
- b) O sistema solicita informações para gerar o relatório.
- c) O cliente informa os dados e clica e Gerar relatório.
- e) O sistema emite o relatório na tela para o usuário.

Fluxo Alternativo

A1- Cancelar Relatório

- e. O usuário escolhe a opção Cancelar.
- f. O sistema volta ao passo a do Fluxo Principal.

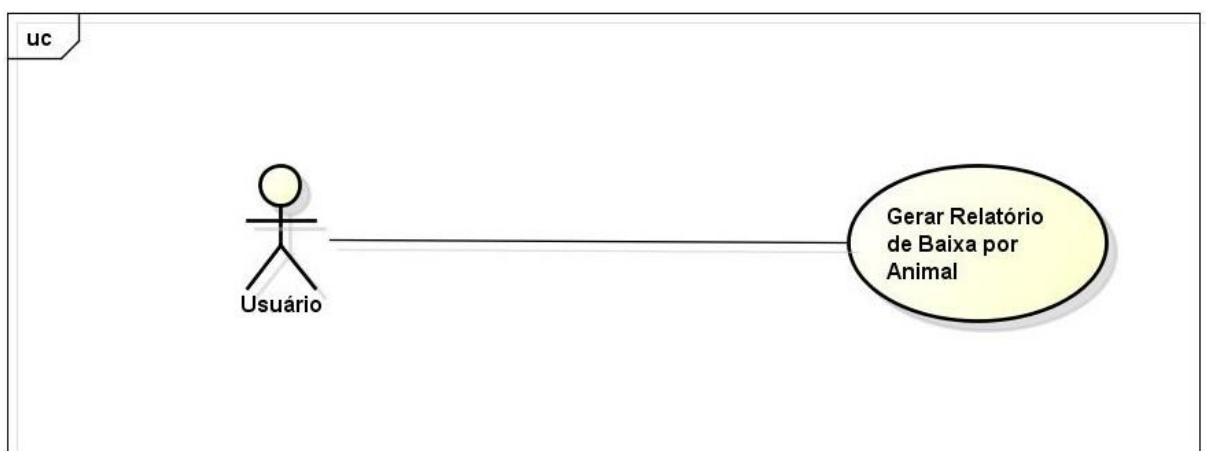


Figura 12 - Ilustra o Caso de Uso- Gerar Relatório de Baixa de Animal

Fluxo Principal

- a) O cliente seleciona a opção Relatório no Menu principal e em seguida seleciona a opção Relatório de Baixa de Animal [A1]
- b) O sistema solicita informações para gerar o relatório.
- c) O cliente informa os dados e clica e Gerar relatório.
- e) O sistema emite o relatório na tela para o usuário.

Fluxo Alternativo

A1- Cancelar Relatório

- a. O usuário escolhe a opção Cancelar.
- b. O sistema volta ao passo a do Fluxo Principal.

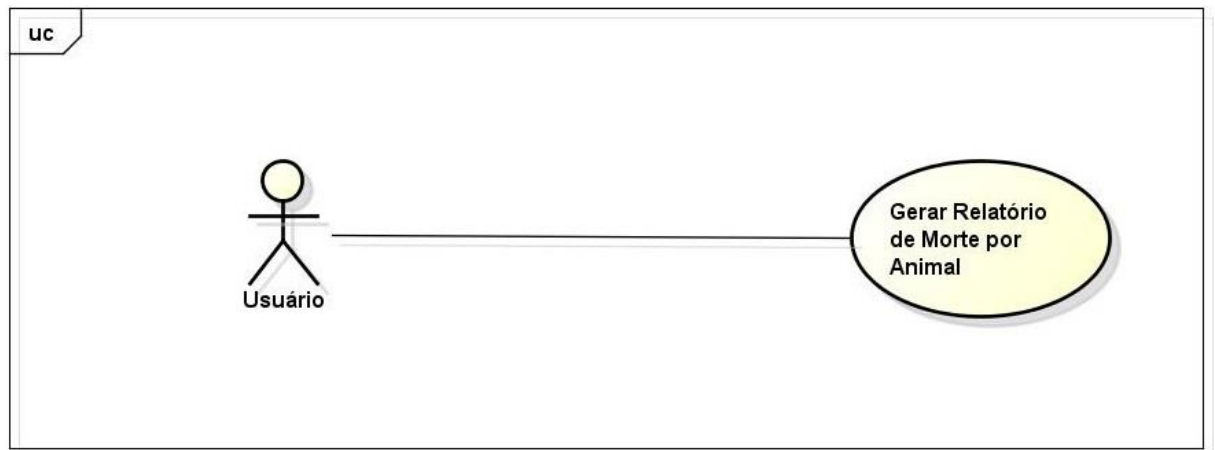


Figura 13- Ilustra o Caso de Uso- Gerar Relatório de Morte por Animal

Fluxo Principal

- a) O cliente seleciona a opção Relatório no Menu principal e em seguida seleciona a opção Relatório de Morte por Animal [A1]
- b) O sistema solicita informações para gerar o relatório.
- c) O cliente informa os dados e clica e Gerar relatório.
- e) O sistema emite o relatório na tela para o usuário.

Fluxo Alternativo

A1- Cancelar Relatório

- a. O usuário escolhe a opção Cancelar.
- b. O sistema volta ao passo a do Fluxo Principal.

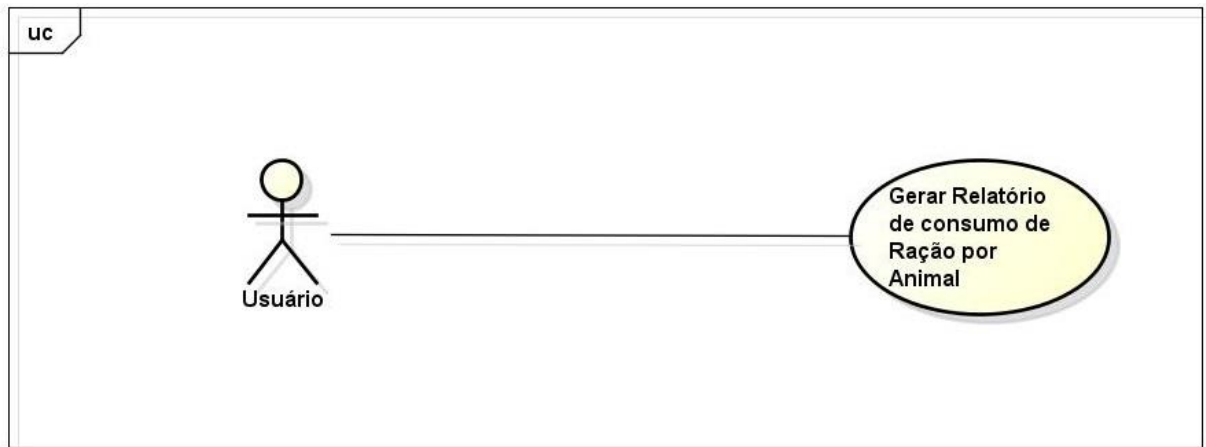


Figura 14- Ilustra o Caso de Uso- Gerar Relatório de consumo de Ração por Animal

Fluxo Principal

- a) O cliente seleciona a opção Relatório no Menu principal e em seguida seleciona a opção Relatório de consumo de Ração por Animal [A1]
- b) O sistema solicita informações para gerar o relatório.
- c) O cliente informa os dados e clica e Gerar relatório.
- e) O sistema emite o relatório na tela para o usuário.

Fluxo Alternativo

A1- Cancelar Relatório

- a. O usuário escolhe a opção Cancelar.
- b. O sistema volta ao passo a do Fluxo Principal.

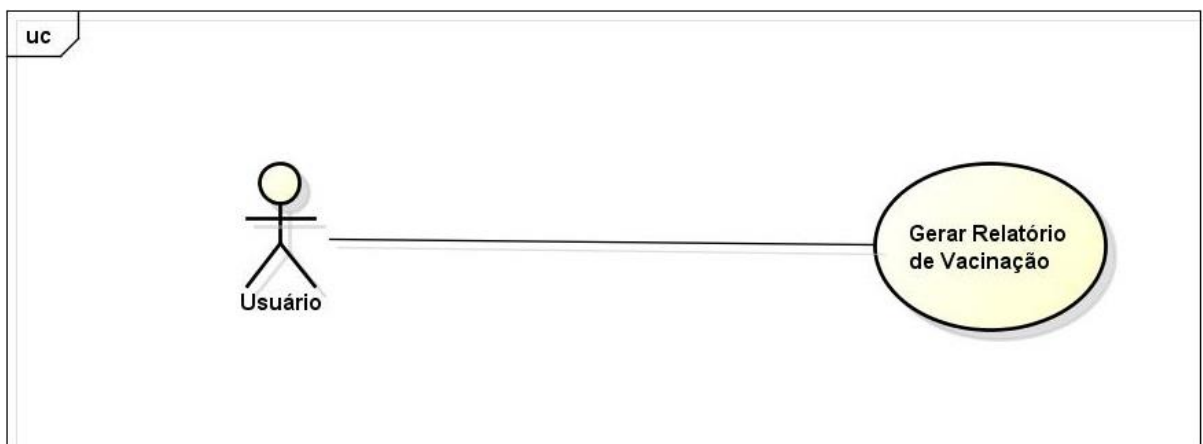


Figura 15- Ilustra o Caso de Uso- Gerar Relatório de Vacinação por Animal

Fluxo Principal

- a) O cliente seleciona a opção Relatório no Menu principal e em seguida seleciona a opção Relatório de Vacinação por Animal [A1]
- b) O sistema solicita informações para gerar o relatório.
- c) O cliente informa os dados e clica e Gerar relatório.
- e) O sistema emite o relatório na tela para o usuário.

Fluxo Alternativo

A1- Cancelar Relatório

- a. O usuário escolhe a opção Cancelar.
- b. O sistema volta ao passo a do Fluxo Principal.

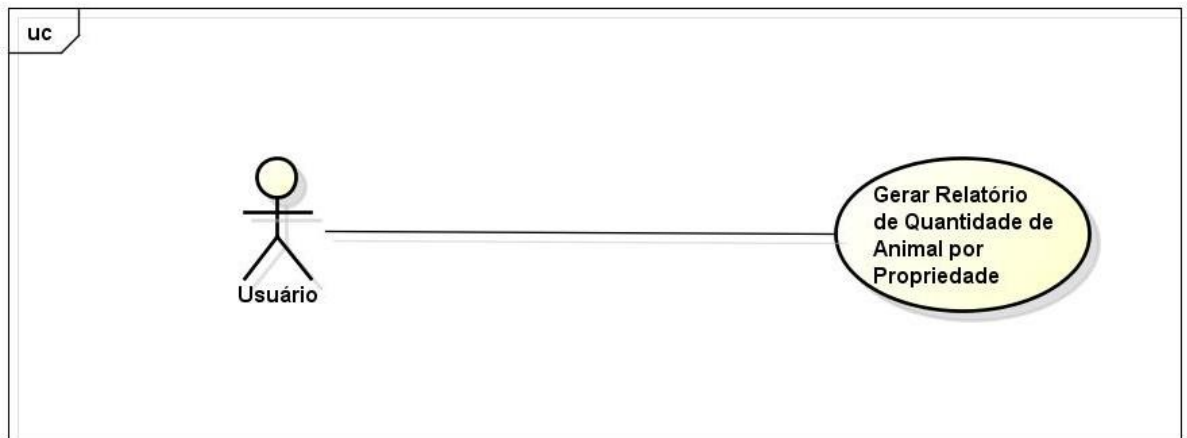


Figura 16- Ilustra o Caso de Uso- Gerar Relatório de Quantidade de Animal por Propriedade

Fluxo Principal

- a) O cliente seleciona a opção Relatório no Menu principal e em seguida seleciona a opção Relatório de Quantidade de Animal por Propriedade [A1]
- b) O sistema solicita informações para gerar o relatório.
- c) O cliente informa os dados e clica e Gerar relatório.
- e) O sistema emite o relatório na tela para o usuário.

Fluxo Alternativo

A1- Cancelar Relatório

- a. O usuário escolhe a opção Cancelar.
- b. O sistema volta ao passo a do Fluxo Principal.

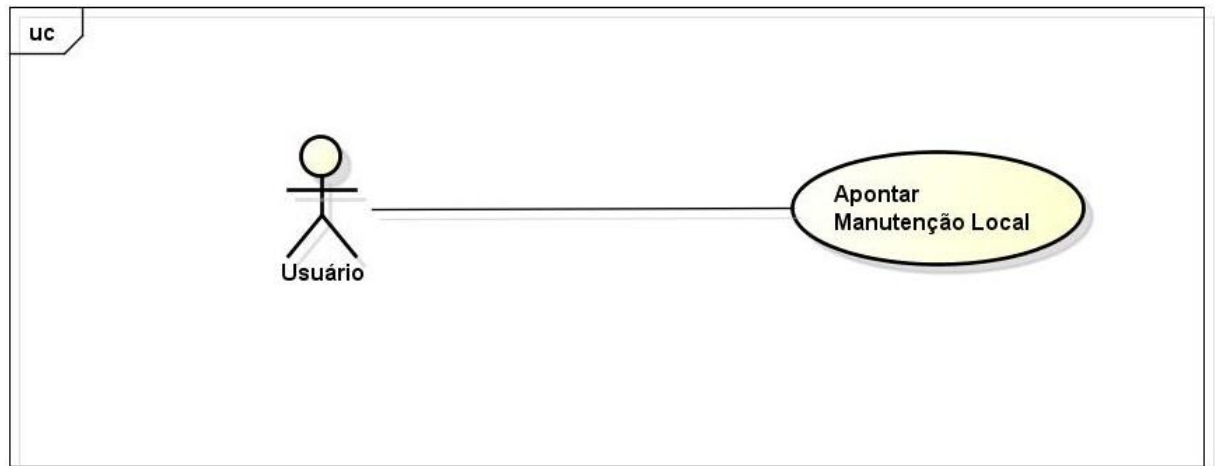


Figura 17- Ilustra o Caso de Uso- Apontar Manutenção Local

Fluxo Principal

- a) O cliente seleciona a opção Apontar Local. **[A1]**
- b) O sistema solicita dados necessários para consultar o Local.
- c) O cliente informa os dados e confirma a operação. **[E1]**
- e) O sistema verifica os dados e solicita qual manutenção será realizada na área **[A2]**
 - f) O cliente informa os dados.
 - g) O sistema solicita que seja informada a situação atual do Local e o motivo do apontamento.
 - h) O cliente informa os dados e confirma.
 - i) O sistema solicita a data do apontamento e data estimada para próxima manutenção.
 - j) O cliente informa os dados e confirma.
 - k) O caso de uso é encerrado.

Fluxo Alternativo

A1- Cancelar Operação

- a. O usuário cancela a operação de Cadastrar, podendo ou não informar algum dado.
- b. O sistema retorna ao passo a do Fluxo Principal.

A2- Informar tipo de Gramínea

- a. O sistema oferece a oportunidade de informar o tipo de Gramínea plantada no local e a quantidade.
- b. O sistema retornar ao passo g do Fluxo Principal.

Fluxo de Exceção

E1- Local não cadastrado

- a. O sistema realiza a consulta e verifica que o Loca não está cadastrado
- b. O sistema retorna ao passo b do Fluxo Principal.

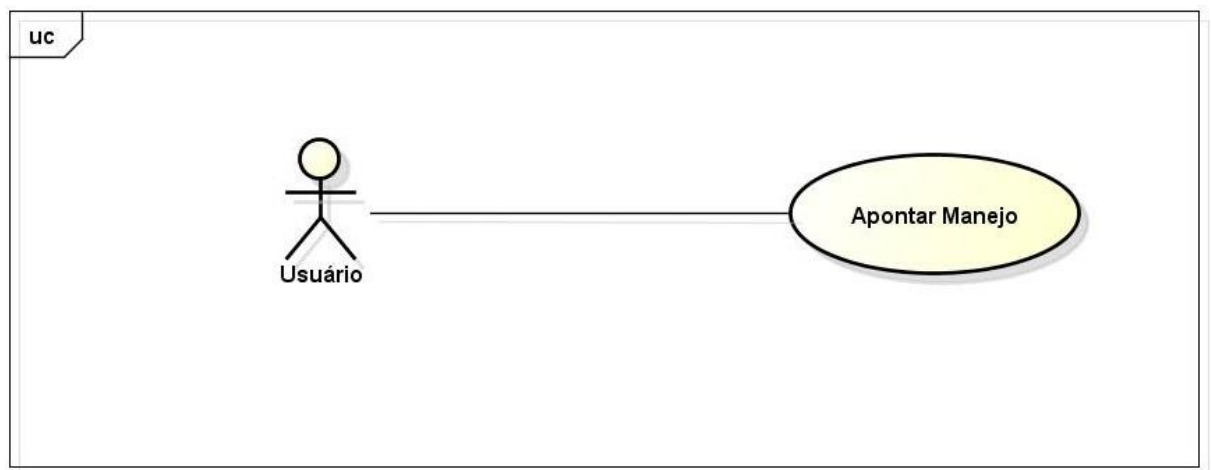


Figura 18 - Ilustra o Caso de Uso- Apontar Manejo

Fluxo Principal

- a) O cliente seleciona a opção Apontar Manejo. **[A1]**
- b) O sistema solicita dados necessários para consultar o Animal.
- c) O cliente informa os dados e confirma a operação. **[E1]**
- e) O sistema verifica os dados e solicita o tipo de manejo que será apontado
- f) O cliente informa os dados.
- g) O sistema habilita os campos e solicita as informações necessárias para o tipo de manejo selecionado.
- h) O cliente informa os dados e confirma.
- i) O sistema solicita a data do apontamento.
- j) O cliente informa os dados e confirma.
- k) O caso de uso é encerrado.

Fluxo Alternativo

A1- Cancelar Operação

- a. O usuário cancela a operação de Cadastrar, podendo ou não informar algum dado.
- b. O sistema retorna ao passo a do Fluxo Principal.

Fluxo de Exceção

E1- Animal não cadastrado

- a. O sistema realiza a consulta e verifica que o Animal não está cadastrado
- b. O sistema retorna ao passo b do Fluxo Principal.

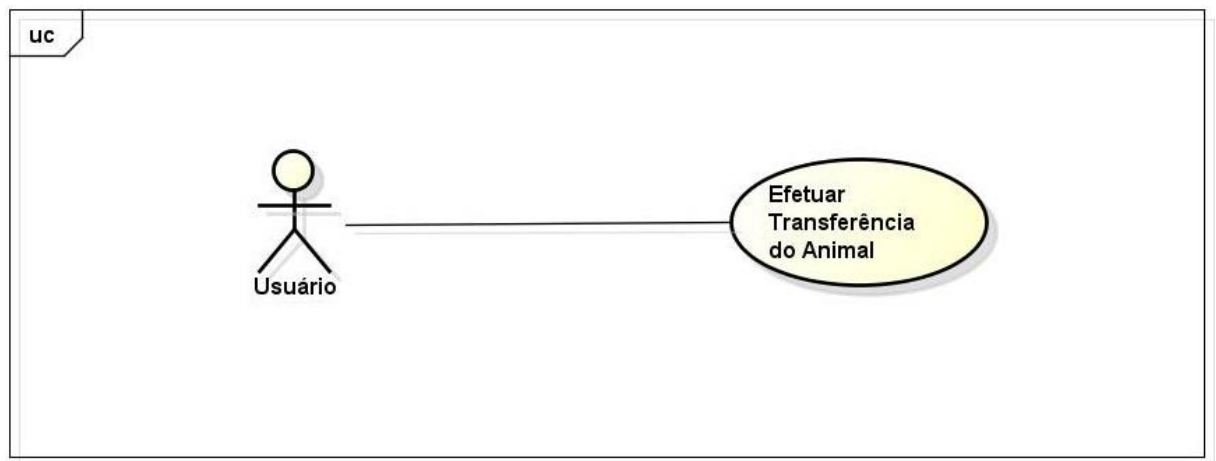


Figura 19 - Ilustra o Caso de Uso- Efetuar Transferência do Animal

Fluxo Principal

- a) O cliente seleciona a opção Transferência do Animal. **[A1]**
- b) O sistema solicita dados necessários para consultar o Animal e lote
- c) O cliente informa os dados e confirma a operação. **[E1][E2]**
- e) O sistema verifica os dados e solicita o tipo de Transferência.
- f) O cliente informa o tipo de transferência.
- g) O sistema habilita os campos e solicita as informações necessárias para o tipo de Transferência.
- h) O cliente informa os dados e confirma. **[E3][E4]**
- i) O sistema solicita a data e hora da Transferência
- j) O cliente informa os dados e confirma.

k) O caso de uso é encerrado.

Fluxo Alternativo

A1- Cancelar Operação

- a. O usuário cancela a operação de Cadastrar, podendo ou não informar algum dado.
- b. O sistema retorna ao passo a do Fluxo Principal.

Fluxo de Exceção

E1- Animal não cadastrado

- a. O sistema realiza a consulta e verifica que o Animal não está cadastrado
- b. O sistema retorna ao passo b do Fluxo Principal.

E2- Lote não cadastrado

- a. O sistema realiza a consulta e verifica que o Lote não está cadastrado
- b. O sistema retorna ao passo b do Fluxo Principal.

E3- Local não cadastrado

- a. O sistema realiza a consulta e verifica que o Local não está cadastrado
- b. O sistema retorna ao passo g do Fluxo Principal.

E4 - Propriedade não cadastrado

- a. O sistema realiza a consulta e verifica que o Propriedade não está cadastrado
- b. O sistema retorna ao passo g do Fluxo Principal.

3.4 Diagrama de classe

Os Diagramas de Classes são os principais diagramas de uma modelagem de sistema orientado a objeto, além disso, são fundamentais na construção de sistemas executáveis e oferece uma visão dos serviços que o sistema deverá oferecer aos usuários finais. Apresentam um relacionamento entre o conjunto de classes, colaborações e interfaces sendo representados por diagramas interligados. (BOOCH; RUMBAUGH; JACOBSON, 2006)

A figura 15 ilustra a estrutura do Sistema Agropecuário proposto.

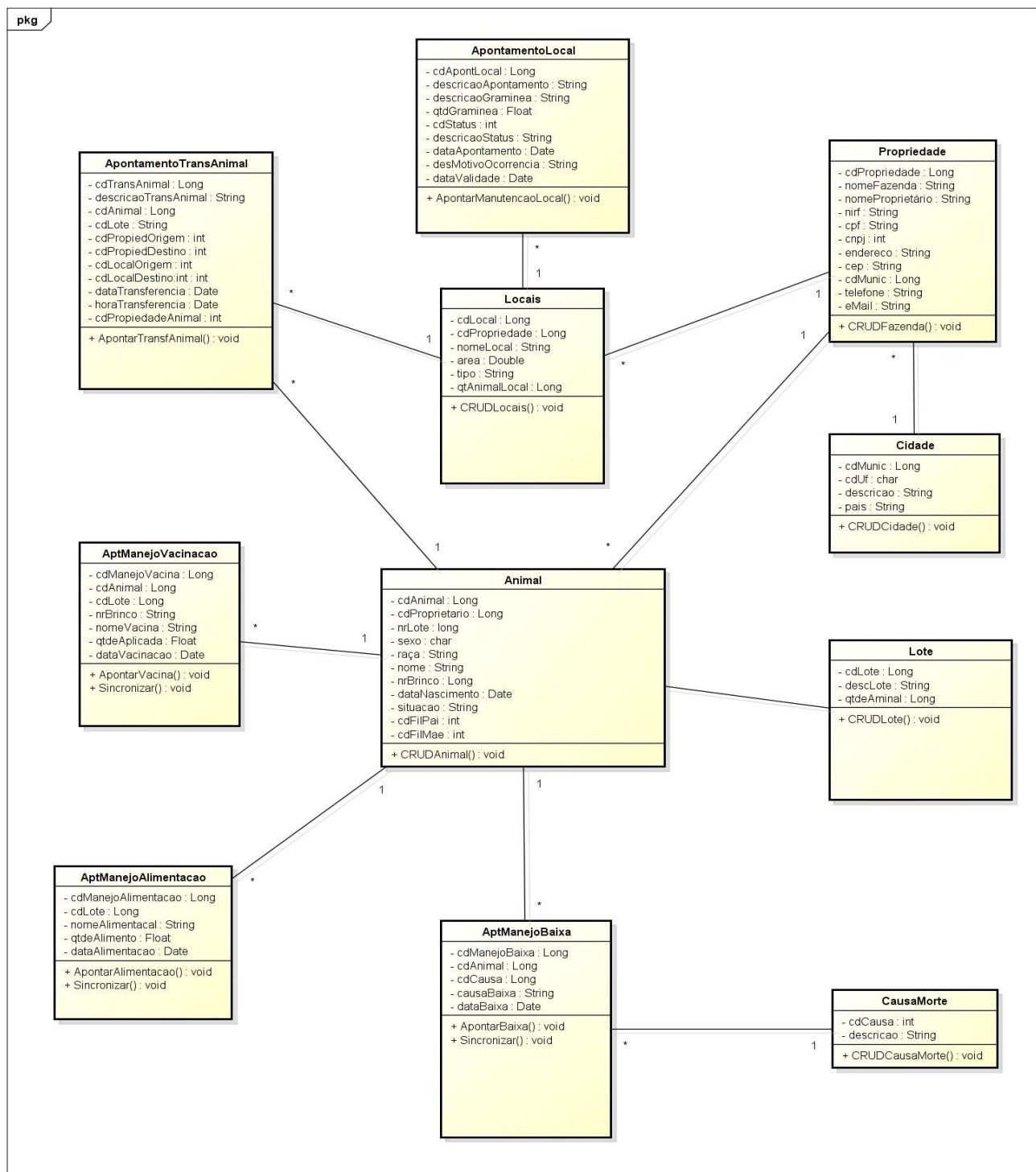


Figura 20- Diagrama de Classe

3.5 Diagrama E-R

O Diagrama Entidade Relacionamento, ilustrado na Figura 16, descreve as entidades e seus relacionamentos estáticos entre si. Fundamentado em um conceito de relação, pode-se pensar, de maneira simples, em uma relação entre tabelas, compostas de linhas e colunas na qual cada relação possui um nome. (BEZERRA, 2011)

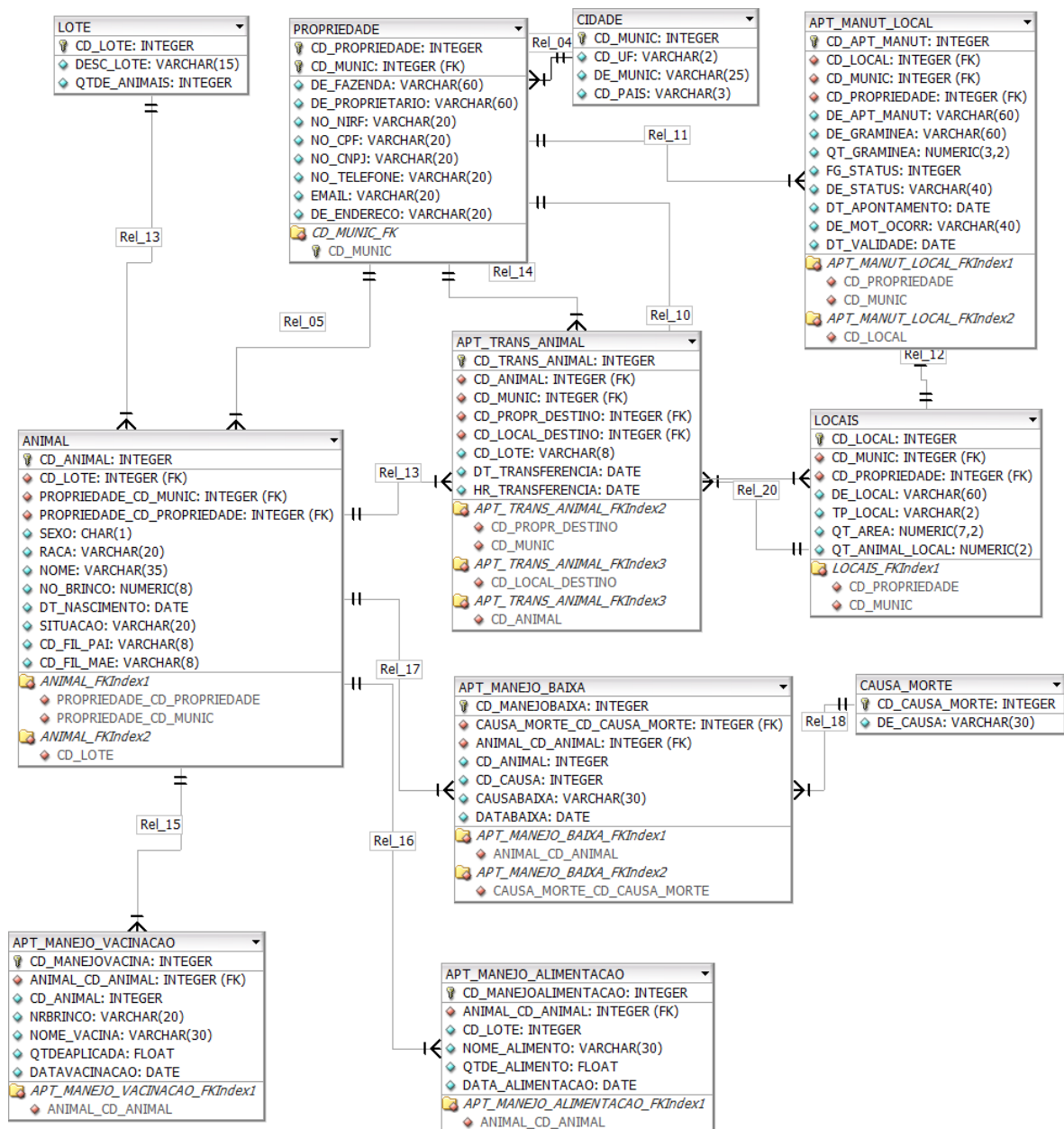


Figura 21– Diagrama Entidade Relacionamento.

3.6 Diagrama de Atividade

O fluxo de controle entre as atividades empregadas para fazer a modelagem dos aspectos dinâmicos do sistema está ilustrado na Figura 22.

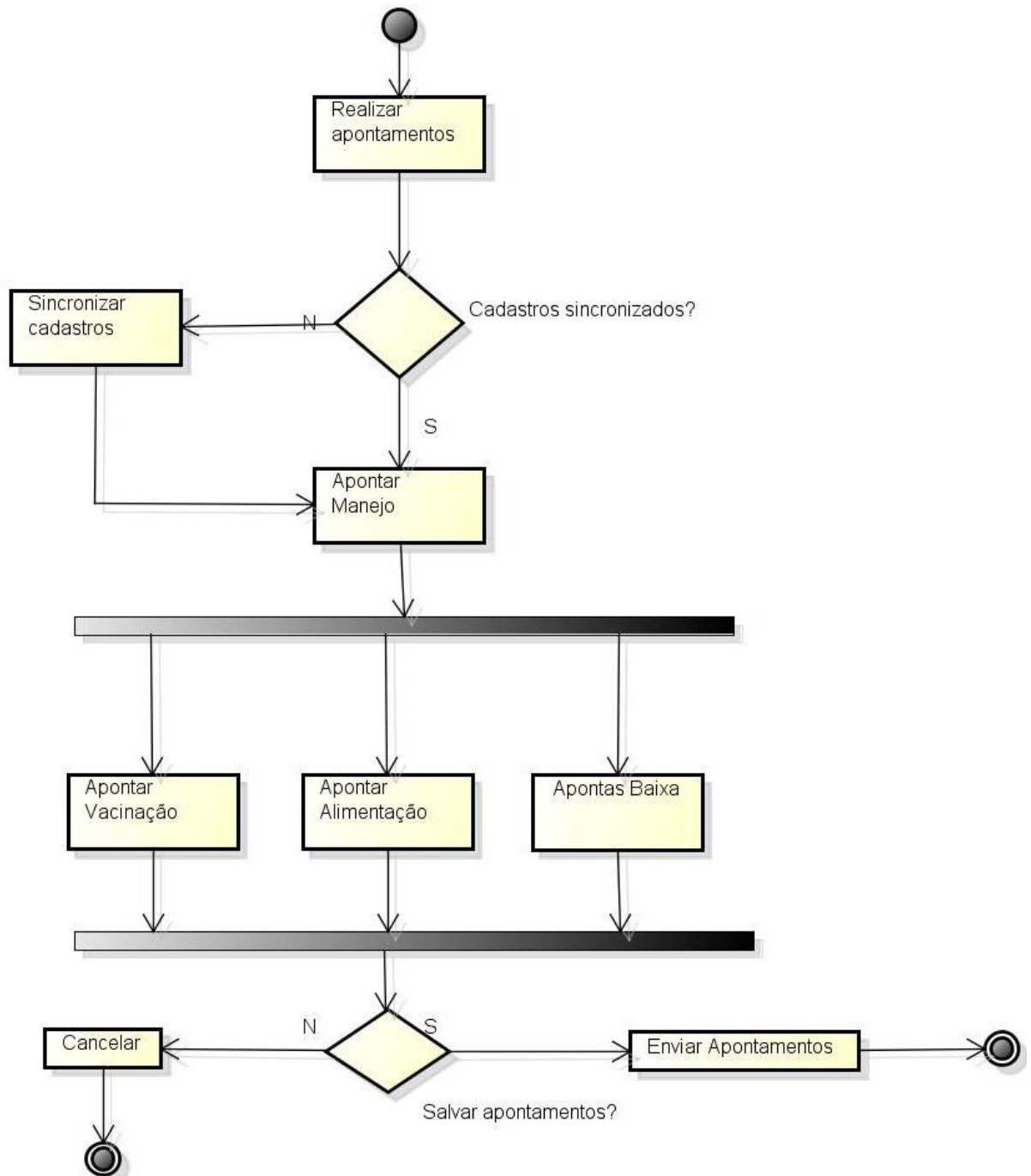


Figura 22– Diagrama de Atividades.

3.7 Arquitetura do Sistema

O aplicativo móvel comunica-se com o servidor de aplicação por meio do protocolo HTTP e utiliza a linguagem XML para padronização de dados enviados ou recebidos. Todos os dados recebidos do servidor ou apontados no próprio dispositivo móvel são armazenados em seu banco de dados local.

O cliente Web também realiza a comunicação com o servidor da aplicação por meio de um protocolo HTTP que por sua vez, realiza o acesso ao banco de dados.

Esta estrutura pode ser melhor compreendida de acordo com a Figura 23.

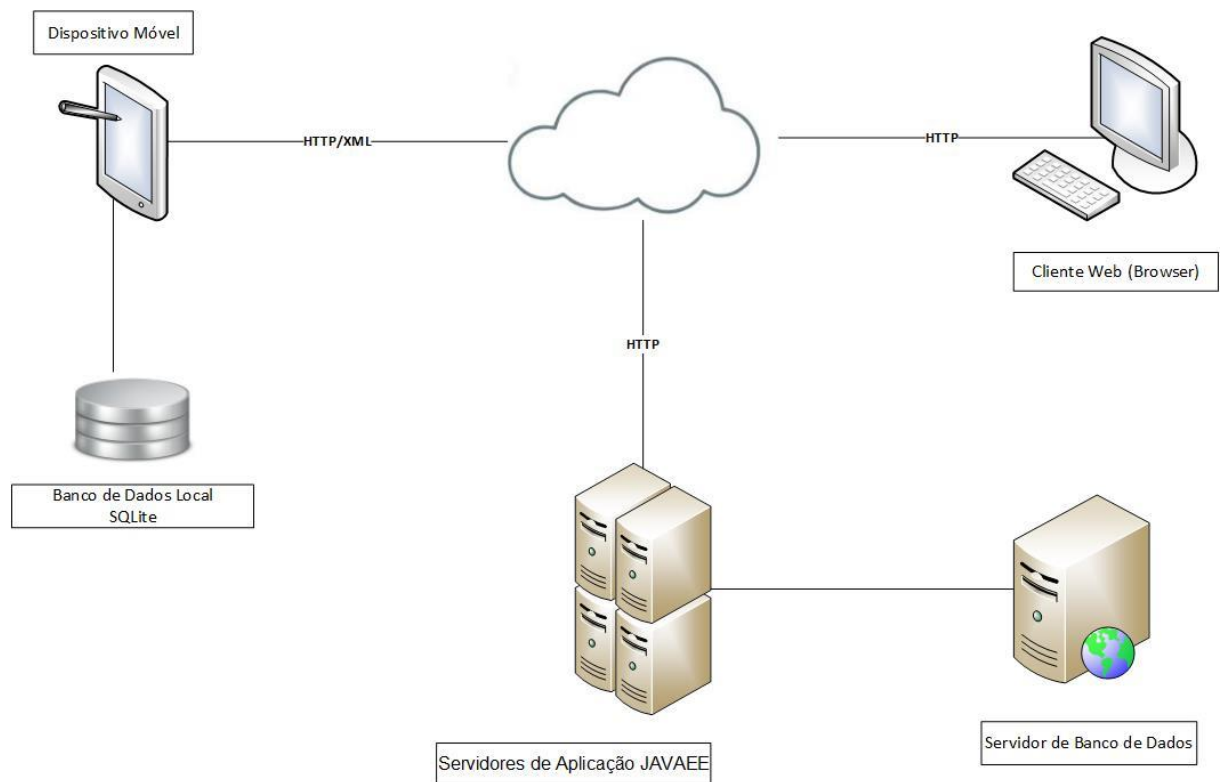


Figura 23– Estrutura do sistema.

4 TELAS DO SISTEMA

Este capítulo apresenta as principais telas que compõem o sistema Web e Mobile com uma breve explicação sobre suas funcionalidades.

4.1 Sistema Web

O Sistema Web possui os seguintes Menus:

- **Cadastros:** destinado ao cadastro de Animais, Causa Morte, Cidades, Locais, Lotes e Propriedades;
- **Movimentações:** este menu possibilita visualizar todos os apontamentos de manejo realizados em campo após a sincronização e também a possibilidade de lançar apontamentos em caráter de contingência;
- **Relatórios:** disponibiliza relatórios com informações de Manejo.

A Figura 24 mostra a interface principal do sistema, na qual é possível visualizar as opções do menu de Cadastro.

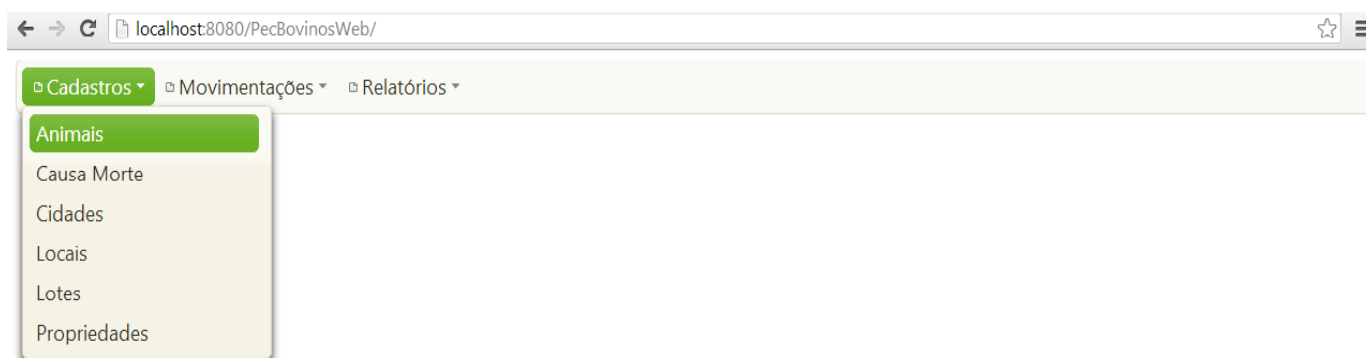


Figura 24 – Tela Principal do PecBovinos Web.

A Figura 25 ilustra a tela de um dos cadastros do Sistema Web, neste caso o cadastro de Causa Morte.

Cadastro de Causa Morte

Código

Descrição Causa

Salvar **Voltar**

Listagem de Causas

Código ↕	Descrição ↕	Opções
1	Envenenamento	Alterar Remover
2	Causas Indefinidas	Alterar Remover
3	Eletrocussão por raio	Alterar Remover
4	Verminose	Alterar Remover
5	Acidente	Alterar Remover

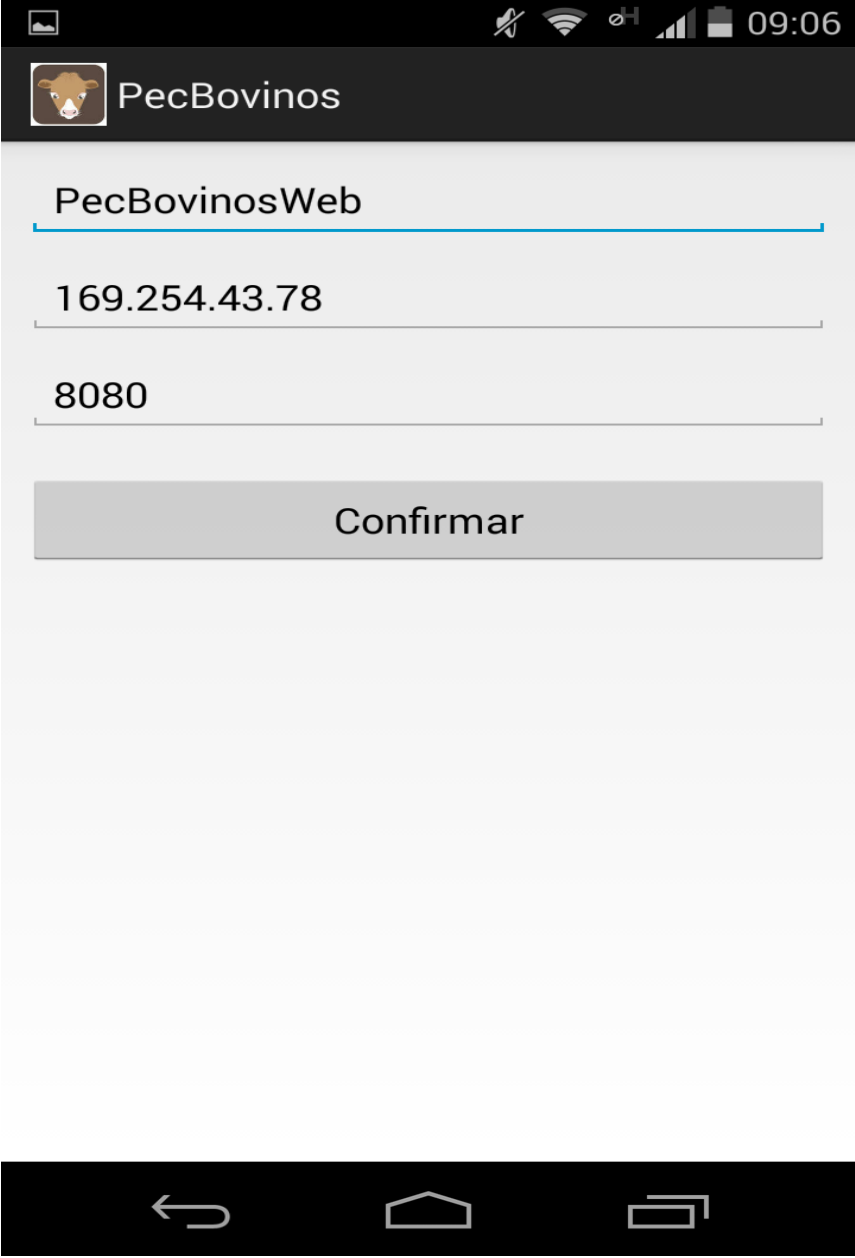
Figura 25– Tela de Cadastro Causa Morte.

4.2 Sistema Mobile

O Sistema Mobile é composto por:

- **Manejo de Animais:** Este menu direciona aos apontamentos de Vacinação, na qual é possível realizar o apontamento da Vacinação por Lote e individualmente; Apontamento de Alimentação e Apontamento da Baixa do animal; (Figura 20)
- **Sincronização:** realiza o sincronismo dos cadastros de Animais, Causa Mortes, Cidades, Lotes e Propriedades;
 - **Configuração:** Neste menu o usuário informa as configurações necessárias para poder realizar o sincronismo dos cadastros. É necessário

informar o nome do sistema, ip da rede e porta de comunicação, como mostra na figura 26.



The screenshot shows a mobile application interface for 'PecBovinos'. At the top, there is a status bar with icons for signal, Wi-Fi, and battery, along with the time '09:06'. Below this is a header bar with a cow icon and the text 'PecBovinos'. The main content area has three text input fields: the first contains 'PecBovinosWeb', the second contains '169.254.43.78', and the third contains '8080'. Below these fields is a large grey button labeled 'Confirmar'. At the bottom of the screen is a black navigation bar with three white icons: a back arrow, a home icon, and a recent apps icon.

Figura 26– Tela de Configurações de conexão.

As demais telas do sistema como apontamentos de manejo e sincronização de dados encontram-se no Apêndice A.

5 CONCLUSÃO

O setor da agropecuária causa um impacto significativo no cenário econômico do país, por este motivo há a necessidade de inserir novas tecnologias neste mercado a fim de aumentar lucros e agregar maior qualidade ao produto final.

O sistema desenvolvido neste trabalho oferece ao produtor de gado a possibilidade de realizar os apontamentos em campo podendo assim controlar aspectos nutricionais, sanitários e de baixas de animal.

As tecnologias adotadas permitem com que estes apontamentos realizados em um ambiente sem sinal de internet, possam ser armazenados no banco de dados do próprio aplicativo móvel, podendo ser sincronizados posteriormente.

A utilização do servidor web contribuiu para uma melhor interação, comunicação entre as diferentes plataformas, além de agregar dinamismo e segurança nas trocas de informações, por não haver interação humana.

As informações coletadas em campo poderão ser utilizadas para uma análise do produtor e possível venda do animal, além de poder controlar custos de alimentação e vacinação.

Também foi possível constatar que o planejamento do projeto e desenvolvimento de diagramas é fundamental para entendimento dos casos de uso que compõe o sistema contribuindo e otimizando para a implementação do mesmo.

Constatou-se que a utilização deste sistema para apontar as atividades de campo de um produtor de bovinos garante a segurança da informação, controle de animais e criação dos mesmos.

Como trabalhos futuros pode-se destacar o desenvolvimento dos apontamentos de transferência de animal para outros locais da fazenda e controle de manutenção de locais. Em caso de problemas com o dispositivo móvel, pode-se também desenvolver um formulário como solução de contingência.

REFERÊNCIAS

ALBRESON *et. al.* **Android em Ação**. Rio de Janeiro: Elsevier, 2012.

ALECRIM, E. **Conhecendo o Servidor Apache**. Disponível em: <<http://www.infowester.com/servapach.php>> Acesso em: jul.2014

BEZERRA, E. **Princípio de Analise e Projeto de sistemas com UML**. Rio de Janeiro: Elsevier, 2011.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: Guia do Usuário**. Rio de Janeiro: Elsevier, 2006.

COMUNIDADE BRASILEIRA DE POSTGRESQL. **Sobre o PostgreSQL**. Disponível em:< <http://www.postgresql.org.br/sobre>> Acesso em: Fev. 2014.

DEITEL, H. M.; DEITEL, P.J. **Java: Como programar**. 8.ed. São Paulo: Person Prentice Hall, 2010.

DevMedia. **Conheça o Apache Tomcat**. Disponível em: <<http://www.devmedia.com.br/conheca-o-apache-tomcat/4546>> Acesso em: jul. 2014.

EMBRAPA. **Boas práticas agropecuárias: bovinos de corte**. Campo Grande, MS: Embrapa Gado de Corte, 2007. Disponível em: <http://www.jbs.com.br/_doc/bpa_jbs_embrapa.pdf> Acesso em: out.2013.

GONÇALVES, Edson. **Desenvolvendo aplicações web com jsp, servlets, javaserver faces, hibernate, ejb3 persistence e AJAX**. Editora Ciência Moderna: Rio de Janeiro, 2007.

JANDL JÚNIOR, P. **Introdução ao Java**. Disponível em: < <http://pt.scribd.com/doc/13989005/Apostila-JAVA-Em-Portugues>>. Acesso em: fev. 2014.

LECHETA, R.R. **Google Android: aprenda a criar aplicações para dispositivos móveis com o Android SDK**. São Paulo: Novatec Editora, 2010.

MACHADO, J. G. C. F.; NANTES, J. F. D. Adoção da tecnologia da informação em organizações rurais: o caso da pecuária de corte. **Gestão e Produção**. São Carlos, v.18, n.3, 2011. Disponível em: < http://www.scielo.br/scielo.php?pid=S0104-530X2011000300009&script=sci_arttext> Acesso em: out. 2013.

MOREIRA NETO, O. **Entendendo e dominando o Java para Internet:** desenvolva aplicativos java para Web!. São Paulo: Digerati Books, 2006.

PEREIRA, L. C. O.; SILVA, M. L. **Android para desenvolvedores.** Rio de Janeiro: Braspot, 2009. Disponível em: <<http://books.google.com.br/books?hl=pt-BR&lr=&id=8u9wJowXfdUC&oi=fnd&pg=PA1&dq=+android++aplica%C3%A7%C3%A3o&ots=LSli422sn1&sig=mJspl0wVyzXwgWCxNbAxikqzIBg#v=onepage&q=android%20%20aplica%C3%A7%C3%A3o&f=false>> Acesso em: fev. 2014.

SCHERER, A. P. Z. **PostgreSQL:** instalando e conhecendo seus recursos. Disponível em: <<http://www.faculadadedombosco.edu.br/downloadAnexo.php?type=artigo&id=9>>. Acesso em: fev. 2014

TAMASSIA, L. F. **Tecnologias para uma pecuária sustentável.** Disponível em: <http://www.agrodebate.com.br/_conteudo/2013/09/artigos/10376-tecnologias-para-uma-pecuaria-sustentavel.html> Acesso em: out. 2013.

APÊNDICE

APÊNDICE A- Telas do Sistema



Figura 27– Tela Principal do PecBovinos Mobile

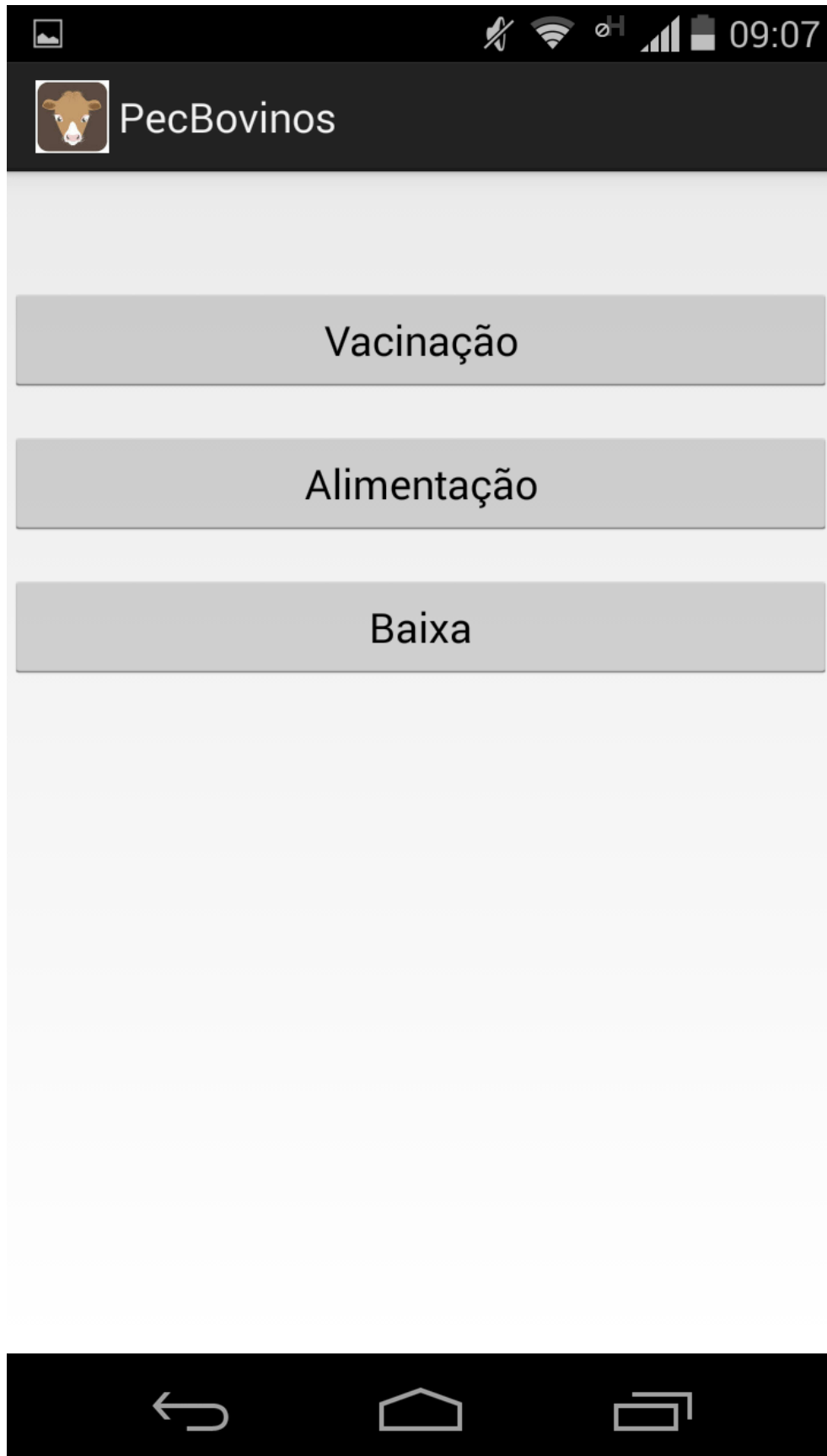


Figura 28 – Tela de Apontamentos de Manejo

The screenshot shows the PecBovinos mobile application interface. At the top, there is a status bar with various icons and the time 09:08. Below the status bar is a dark header with a cow icon and the text "PecBovinos". The main content area is light gray and contains five labeled input fields: "Código", "Animal", "Nome da Vacina", "Quantidade Aplicada", and "Data da Vacinação". Each field has a corresponding input line. The "Animal" field has a magnifying glass icon to its right. The "Data da Vacinação" field has a calendar icon to its right. At the bottom of the form is a large gray button labeled "Salvar". The bottom of the screen features a black navigation bar with three white icons: a back arrow, a home house icon, and a recent apps icon.

Código

Animal

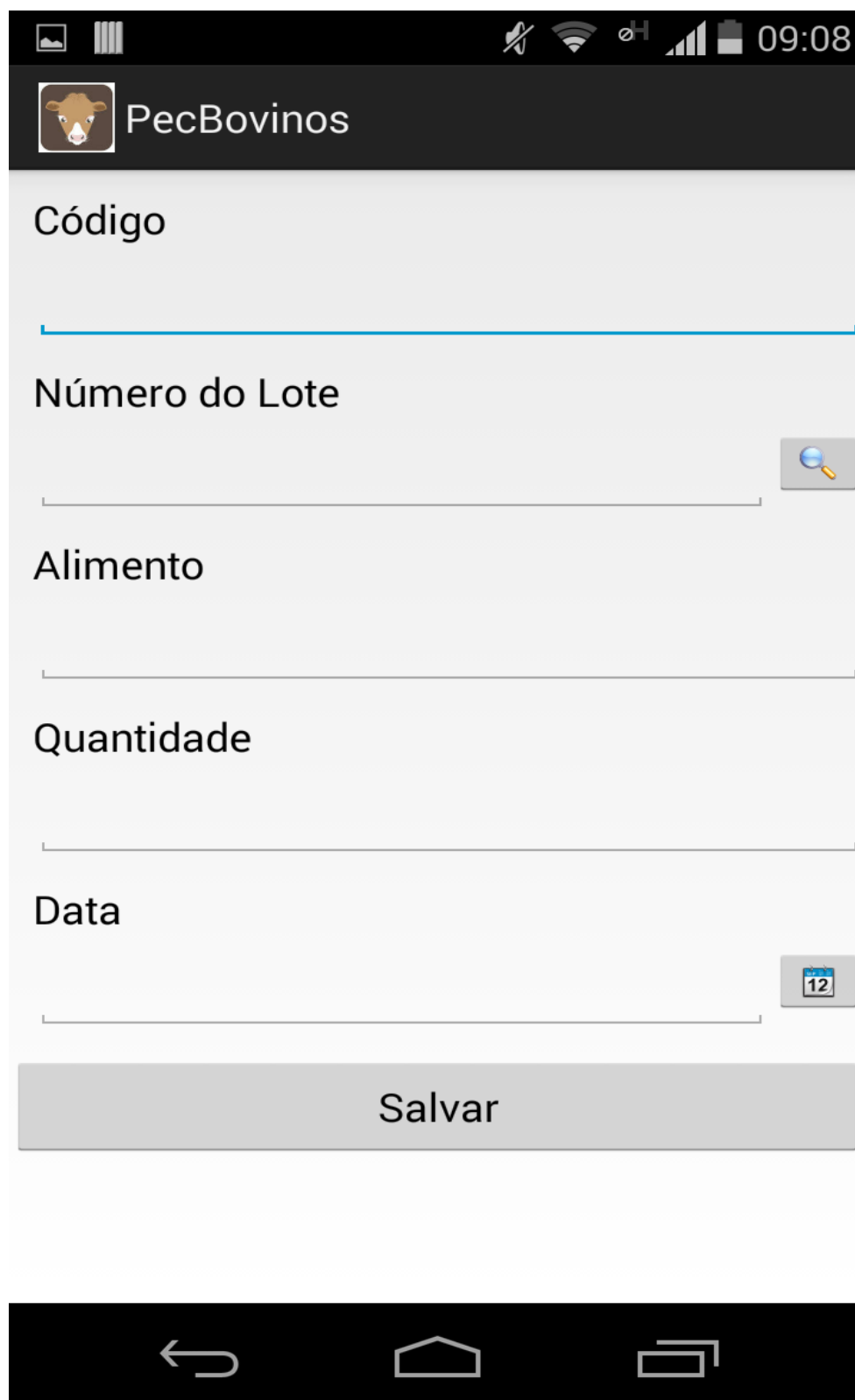
Nome da Vacina

Quantidade Aplicada

Data da Vacinação

Salvar

Figura 29 – Tela de Apontamento de Vacina por Animal



The screenshot shows the PecBovinos application interface. At the top, there is a status bar with various icons and the time 09:08. Below the status bar is a header with a cow icon and the text "PecBovinos". The main form consists of several input fields with labels: "Código", "Número do Lote", "Alimento", "Quantidade", and "Data". Each field has a corresponding input line. The "Número do Lote" field has a magnifying glass icon to its right. The "Data" field has a calendar icon to its right, showing the number 12. At the bottom of the form is a large grey button labeled "Salvar". The bottom of the screen features a black navigation bar with three white icons: a back arrow, a house icon, and a list icon.

Código

Número do Lote

Alimento

Quantidade

Data

Salvar

Figura 30 – Tela de Apontamento de Alimentação

 PecBovinos

Código

Animal 

Motivo da Baixa 

Causa Morte 

Data 

Salvar

Figura 31 – Tela de Apontamento de Baixa de Animais



Figura 32 – Tela de Sincronização de Cadastros