

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

JEAN MARCOS FUSINATTO

SOFTWARE PARA GERENCIAMENTO DE GADO LEITEIRO

TRABALHO DE CONCLUSÃO DE CURSO

**PATO BRANCO
2018**

JEAN MARCOS FUSINATTO

SOFTWARE PARA GERENCIAMENTO DE GADO LEITEIRO

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Profa. Beatriz Terezinha Borsoi

**PATO BRANCO
2018**



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Pato Branco
Departamento Acadêmico de Informática
Curso de Tecnologia em Análise e Desenvolvimento
de Sistemas



TERMO DE APROVAÇÃO
TRABALHO DE CONCLUSÃO DE CURSO

SOFTWARE PARA GERENCIAMENTO DE GADO LEITEIRO

POR

JEAN MARCOS FUSINATTO

Este trabalho de conclusão de curso foi apresentado no dia 26 de junho de 2018, como requisito parcial para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, pela Universidade Tecnológica Federal do Paraná. O acadêmico foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Banca examinadora:

Profª Drª Beatriz Terezinha Borsoi
Orientador

Profª MSc Andreia Scariot Beulke

Prof. MSc.Vinicius Pegorini

Prof. Dr. Edilson Pontarolo
Coordenador do Curso de Tecnologia em
Análise e Desenvolvimento de Sistemas

Profª Drª Beatriz Terezinha Borsoi
Responsável pela Atividade de Trabalho de
Conclusão de Curso

A Folha de Aprovação assinada encontra-se na Coordenação do Curso.

RESUMO

FUSINATTO, Jean Marcos. Software para gerenciamento de gado leiteiro. 2018. 63f. Monografia (Trabalho de Conclusão de Curso 1) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2018.

A pecuária leiteira é uma das principais atividades agrícolas em termos de relevância do Brasil e, juntamente com a produção de grãos, compõem um dos eixos de sustentabilidade do país. A atividade de produção leiteira é realizada desde por pequenos produtores com a ordenha manual até os grandes produtores com sistemas mecanizados (há processos de ordenha, embora poucos, que são completamente automatizados). Quanto ao rebanho há os que ocupam extensivas áreas de pastagens e há os mantidos em confinamento. O desenvolvimento de tecnologias aplicadas à agropecuária tem sido expressivo, tanto em termos de automação como de sistemas de informação para o gerenciamento e o controle da produção. Neste texto é apresentada a modelagem e a implementação de um sistema *desktop* para o gerenciamento de gado leiteiro em uma pequena propriedade que realiza a ordenha mecanizada. O objetivo do sistema é prover um controle para as receitas e as despesas, da produção e das intervenções realizadas nos animais, fornecendo relatórios que possibilitem ao gestor uma visão ampla e realista do negócio. O sistema foi implementado para ambiente *desktop* utilizando a linguagem Java.

Palavras-chave: Sistema para gerenciamento de gado leiteiro. Informatização de pequenas propriedades. TI no agronegócio.

ABSTRACT

FUSINATTO, Jean Marcos. Software to management of dairy cattle. 2018. 63f. Monografia (Trabalho de Conclusão de Curso 1) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2018.

Dairy farming is one of the main relevant agricultural activities in Brazil and, together with the production of grains, make up one of the axes of country's sustainability. Milk production is mainly carried out by small producers with manual milking up, but there are large producers with mechanized systems, though there are very few that are completely automated. As for the cattle, there are those that occupy extensive areas of pasture and there are those kept in confinement. The development of technologies applied to agriculture has been expressive, both in terms of automation and information systems for the management and control of production. In this text is reported about the software developed to management dairy cattle in a small property that performs mechanized milking. The goal of the system is to provide the control of incomes and expenses, production and interventions performed on animals, providing reports that allow the manager have a broad and realistic view of the business. The system is desktop software develop in Java language.

Keywords: Software to management of dairy cattle. Information of small agricultural properties. TI in agribusiness.

LISTA DE FIGURAS

Figura 1 – Tecnologias da informação no agronegócio.....	15
Figura 2 – Agricultura de precisão (AP).....	17
Figura 3 – Zootecnia de precisão.....	18
Figura 4 – Zootecnia de precisão e rastreabilidade.....	19
Figura 5 – m-Business na empresa agrícola.....	20
Figura 6 – Diagrama de casos de uso.....	27
Figura 7 – Diagrama de entidades e relacionamentos	31
Figura 8 – Tela de autenticação ao sistema.....	39
Figura 9 – Tela principal.....	39
Figura 10 – Pesquisa de pessoas.....	40
Figura 11 – Cadastro de pessoas.....	41
Figura 12 – Mensagem de campo obrigatório.....	42
Figura 13 – Mensagem de campo inválido	42
Figura 14 – Mensagem de sucesso na gravação.....	43
Figura 15 – Solicitação de confirmação para de exclusão	44
Figura 16 – Registro de ordenha.....	45
Figura 17 – Registro de coleta do leite	45
Figura 18 – Registro de análise do leite.....	46
Figura 19 – Filtros para geração do relatório de produção	46
Figura 20 – Relatório de produção de leite	47
Figura 21 - Relatório de qualidade do leite.....	48

LISTA DE QUADROS

Quadro 1 – Tecnologias e ferramentas utilizadas.....	22
Quadro 2 – Requisitos funcionais.....	25
Quadro 3 – Requisitos não funcionais	26
Quadro 4 – Operação incluir dos casos de uso de cadastro	28
Quadro 5 – Operação alterar dos casos de uso de cadastro.....	28
Quadro 6 – Operação excluir dos casos de uso de cadastro.....	29
Quadro 7 – Operação consulta dos casos de uso de cadastro.....	29
Quadro 8 – Realizar controle de estoque	29
Quadro 9 – Inclusão de dados de ordenha	30
Quadro 10 – Inclusão de dados de análises de leite.....	30
Quadro 11 – Campos da tabela Usuario.....	32
Quadro 12 – Campos da tabela Pessoa	32
Quadro 13 – Campos da tabela TipoPessoa.....	32
Quadro 14 – Campos da tabela Animal	33
Quadro 15 – Campos da tabela Rebanho	33
Quadro 16 – Campos da tabela Ordenha.....	33
Quadro 17 – Campos da tabela VacinaAnimal	34
Quadro 18 – Campos da tabela Produto.....	34
Quadro 19 – Campos da tabela Marca	34
Quadro 20 – Campos da tabela TipoProduto	35
Quadro 21 – Campos da tabela Estoque	35
Quadro 22 – Campos da tabela Receita	35
Quadro 23 – Campos da tabela TiposOperacao	36
Quadro 24 – Campos da tabela Despesa	36
Quadro 25 – Campos da tabela Tanque	36
Quadro 26 – Campos da tabela ColetaLeite.....	37
Quadro 27 – Campos da tabela AnaliseLeite	37
Quadro 27 – Campos da tabela Raca	37
Quadro 28 – Campos da tabela ComposicaoRacial.....	38
Quadro 29 – Campos da tabela Pelagem.....	38
Quadro 30 – Campos da tabela Cidade.....	38
Quadro 31 – Campos da tabela Estado	38

LISTA DE SIGLAS

CRUD	<i>Create, Read, Update e Delete</i>
DAO	<i>Data Access Object</i>
FAO	<i>Food and Agriculture Organization</i>
GPS	Sistemas de Posicionamento Global
IBGE	Instituto Brasileiro de Geografia e Estatística
IDE	<i>Integrated Development Environment</i>
JDBC	<i>Java Database Connectivity</i>
JPA	<i>Java Persistence API</i>
MVC	<i>Model-View-Controller</i>
RF	Rádio Frequência
RFID	<i>Radio Frequency Identification</i>
SIG	Sistemas de Informações Geográficas

SUMÁRIO

1 INTRODUÇÃO.....	10
1.1 CONSIDERAÇÕES INICIAIS	10
1.2 OBJETIVOS.....	12
1.2.1 Objetivo Geral.....	12
1.2.2 Objetivos Específicos	12
1.3 JUSTIFICATIVA.....	13
1.4 ORGANIZAÇÃO DO TEXTO	13
2 REFERENCIAL TEÓRICO	14
2.1 APLICAÇÕES DA TI NO AGRONEGÓCIO BRASILEIRO	14
2.2.1 Tecnologias de administração e gestão	15
2.2.2 Tecnologias de controle, monitoramento e robótica.....	16
2.2.3 Tecnologias de Telecomunicações e Internet	19
3 MATERIAIS E MÉTODO	22
3.1 MATERIAIS	22
3.2 MÉTODO.....	22
4 RESULTADO	24
4.1 ESCOPO DO SISTEMA	24
4.2 MODELAGEM DO SISTEMA	24
4.3 APRESENTAÇÃO DO SISTEMA	39
4.4 IMPLEMENTAÇÃO DO SISTEMA.....	49
5 CONSIDERAÇÕES FINAIS.....	62
REFERÊNCIAS.....	63

1 INTRODUÇÃO

Este capítulo apresenta as considerações iniciais, os objetivos e a justificativa. O capítulo é finalizado com a apresentação dos capítulos.

1.1 CONSIDERAÇÕES INICIAIS

A produção de leite é uma atividade vinculada à pecuária que está presente em grande parte do país e está em ascensão tanto no Brasil quanto no mundo. As Tabelas 1 a 3 apresentam dados que sustentam essa afirmação.

A Tabela 1 apresenta dados da produção mundial de leite das diversas espécies animais que fornecem leite para consumo humano, seja para consumo do produto *in natura* ou industrializado nos mais diversos produtos derivados (como queijos e iogurtes) ou como matéria-prima no preparo de alimentos industrializados ou de produção caseira. Os dados para 2011 apresentados na Tabela 1 são provenientes *Food and Agriculture Organization* (FAO)/Faostat e foram elaborados pela Embrapa Gado de Leite. Os dados de 2011 são estimados.

Tabela 1 - Produção mundial de leite de diferentes espécies - 2010/2011

Espécie	Volume de produção (toneladas)		% do total
	2010	2011*	
Vaca	599.615.097	610.247.100	82,9
Búfala	92.514.917	95.439.057	13
Cabra	16.646.618	17.231.269	2,3
Ovelha	10.025.106	10.333.863	1,4
Camela	2.178.269	2.354.133	0,3
TOTAL	720.980.007	735.505.664	

Fonte: Empresa ... (2014, p. 1).

A tendência de crescimento da produção leiteira é sustentada por meio dos dados apresentados na Tabela 2 que se referem ao Brasil no período de 2005 a 2010. Os dados de 2011, desta Tabela 2, também são estimados. Os dados da Tabela 2 são atribuídos à fonte citada ao Instituto Brasileiro de Geografia e Estatística (IBGE)/Pesquisa da Pecuária Nacional e elaborados pela Embrapa Gado de Leite. A Tabela 2 torna evidente o crescimento da produção no país. Um volume de mais de 32 milhões de litros para 2011, indicam a necessidade de suporte de tecnologias que automatizem o processo de ordenha e de sistemas computacionais para a gestão da produção e das suas receitas e despesas. Para esse volume de

produção é necessário o suporte de aplicativos computacionais que auxiliem no processo de tomada de decisão.

Tabela 2 - Produção de leite, vacas ordenhadas e produtividade animal no Brasil – 2005/2010

Ano	Vacas Ordenhadas (mil cabeças)	Produtividade (litros/vaca/ano)	Volume produzido (milhões de litros)
2005	20.820	1.183	24.621
2006	20.943	1.213	25.398
2007	21.122	1.237	26.134
2008	21.599	1.277	27.585
2009	22.435	1.297	29.105
2010	22.925	1.340	30.715
2011	23.508	1.374	32.296

Fonte: Milkpoint (2013, p. 1).

A Tabela 3 apresenta dados da produção brasileira em suas cinco regiões. Essa tabela mostra o aumento do volume produzido e o crescimento e a redução nas diversas regiões do país. Esses dados compreendem um panorama de quase quatro décadas.

Tabela 3 - Produção brasileira de leite por região

Ano	Norte (%)	Nordeste (%)	Sudeste (%)	Sul (%)	Centro-Oeste (%)	Produção anual (bilhões de litros)
1974	1	13	54	23	9	7,1
1980	1	14	51	23	11	11,2
1990	4	14	48	23	12	14,5
2000	5	11	43	25	16	19,8
2010	6	13	36	31	14	30,7
2011	5	13	35	32	15	32,1

Fonte: Maia et al. (2013, p. 375).

A MilkPoint indicava em 2013 que a produção de leite no Brasil aumentaria 5% em 2014, conforme projeção do IBGE. Elevando, assim, a produção no país para 36,75 bilhões de litros em 2014. Essa produção em 2013 foi de 35 bilhões de litros (MILKPOINT, 2014).

Tendo em vista o crescimento da produção de leite no país foi verificada uma oportunidade de mercado para criação de um software para gerenciamento de dados produção de gado leiteiro. Há que se ressaltar que existentes aplicativos e sistemas computacionais em versões pagas e até gratuitas para esse gerenciamento. A própria Embrapa Gado de Leite oferece um sistema de gestão informatizada de sistemas de produção de leite, denominado GisLeite (GISLEITE, 2014). Contudo, embora existam aplicativos computacionais para o

gerenciamento do gado leiteiro, o interesse por implementar um sistema para esse gerenciamento decorre do acesso do autor desta proposta a uma propriedade rural de produção de leite que não possui esse tipo de controle. Essa propriedade será utilizada para o levantamento dos requisitos e para uso experimental do sistema.

O software proposto como resultado deste trabalho foi desenvolvido para a plataforma *desktop* e fornecerá informações para o gerenciamento e acompanhamento das atividades rotineiras de ordenha e análise realizadas em amostra de leite, controle de receitas, despesas e estoque. Além disso, o aplicativo fornecerá dados que podem auxiliar os produtores na tomada de decisões sobre a produção de leite, avaliação de sanidade dos animais e o controle financeiro do rebanho.

1.2 OBJETIVOS

A seguir são apresentados o objetivo geral e os objetivos específicos deste trabalho.

1.2.1 Objetivo Geral

Desenvolver um sistema que visa facilitar o armazenamento e o controle de dados relacionados a gado leiteiro.

1.2.2 Objetivos Específicos

Com o software desenvolvido os objetivos específicos são:

- Fornecer as informações necessárias para um melhor acompanhamento do produtor e auxiliar na tomada de decisão sobre o rebanho.
- Acompanhar dados de produção leiteira e de análises realizadas com as amostras de leite visando identificar tratamentos e medicações que possam ser necessários.
- Armazenar e disponibilizar dados sobre produção média e individual dos animais ordenhados e as interações medicamentosas realizadas.

1.3 JUSTIFICATIVA

Os volumes de produção anual apresentados nas Tabelas 1 a 3 justificam a necessidade de aplicativos computacionais que auxiliem na gestão e no controle da produção leiteira. Além das atividades inerentes à ordenha, a agropecuária de leite envolve uma série de outras tarefas que necessitam de controle.

Considerando os custos financeiros envolvidos na produção agropecuária de leite torna-se indispensável o auxílio por meio de sistemas que permitam armazenar dados que possam efetivamente auxiliar o produtor na tomada de decisão. Assim, os investimentos estarão respaldados por dados históricos que juntamente com outras informações do mercado (como valores de insumos e do leite) permitem analisar a viabilidade de investimentos e o retorno sobre investimento.

O sistema desenvolvido é para ambiente *desktop* porque o mesmo será utilizado por cada produtor individualmente e por que a Internet nem sempre faz parte da realidade dos pequenos produtores.

1.4 ORGANIZAÇÃO DO TEXTO

O restante deste texto está organizado em capítulos. No Capítulo 2 é apresentado o referencial teórico, centrado em tecnologia da informação e o seu uso como ferramenta de suporte ao negócio em empresas de pequeno porte. No Capítulo 3 são apresentadas as tecnologias e as ferramentas utilizadas para a modelagem e o desenvolvimento do sistema. No Capítulo 4 está o resultado do desenvolvimento que é a modelagem e a implementação do sistema. O texto é finalizado com as considerações finais, no Capítulo 5, seguido da listagem das referências citadas no texto.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial teórico do trabalho voltado para o uso de tecnologias de informação e comunicação nas atividades agrícolas seja para a produção de alimentos de origem vegetal, seja para a produção de origem animal.

2.1 APLICAÇÕES DA TI NO AGRONEGÓCIO BRASILEIRO

O desenvolvimento de tecnologias de informação e comunicação está em ritmo acelerado. Estudos da Forrester indicam que o mercado brasileiro nesse segmento deverá crescer 11,6% (STEFANINI, 2014). O setor de agronegócio também promoveu ambiente propício ao desenvolvimento e uso de tecnologias, embora de uma forma mais lenta e menos intensiva que os setores de comércio, prestação de serviços e indústria (EMBRAPA, 2010; OFFICE OF TECHNOLOGY ASSESSMENT, 1992; PAGLIS, 2010; MURAKAMI; SARAIVA, 2005; SILVA et al., 2002; ZAMBALDE, 2000).

Dois fatores contribuíram para impulsionar o processo de adoção, uso e desenvolvimento das diversas aplicações das tecnologias da informação no agronegócio (ARRAES, 1993, p. 2): a necessidade das organizações para que esse segmento se tornasse mais eficaz nas atividades direta e indiretamente relacionadas aos seus processos produtivos; o interesse das organizações de tecnologia da informação (software, hardware, automação e telecomunicações) pelo setor no sentido de expandir seus mercados.

As aplicações das tecnologias nas atividades agrícolas e de produção animal podem ser classificadas de diferentes formas, conforme as características que se deseja identificar e aplicar, considerando as reais necessidades do mercado e podem ser sintetizadas em três grandes grupos (AGROSOFT, 2000; ARRAES, 1993; CARRASCAL, 1985; EMBRAPA, 2010; FARMSOFT, 1994; OFFICE OF TECHNOLOGY ASSESSMENT, 1992; MURAKAMI; SARAIVA, 2005; SILVA et al., 2002; ZAMBALDE, 2000): a) administração e gestão, b) controle, monitoramento e robótica; e c) telecomunicações e internet. A Figura 1 apresenta um esboço ilustrativo desses três grupos de tecnologias da informação aplicadas ao agronegócio.

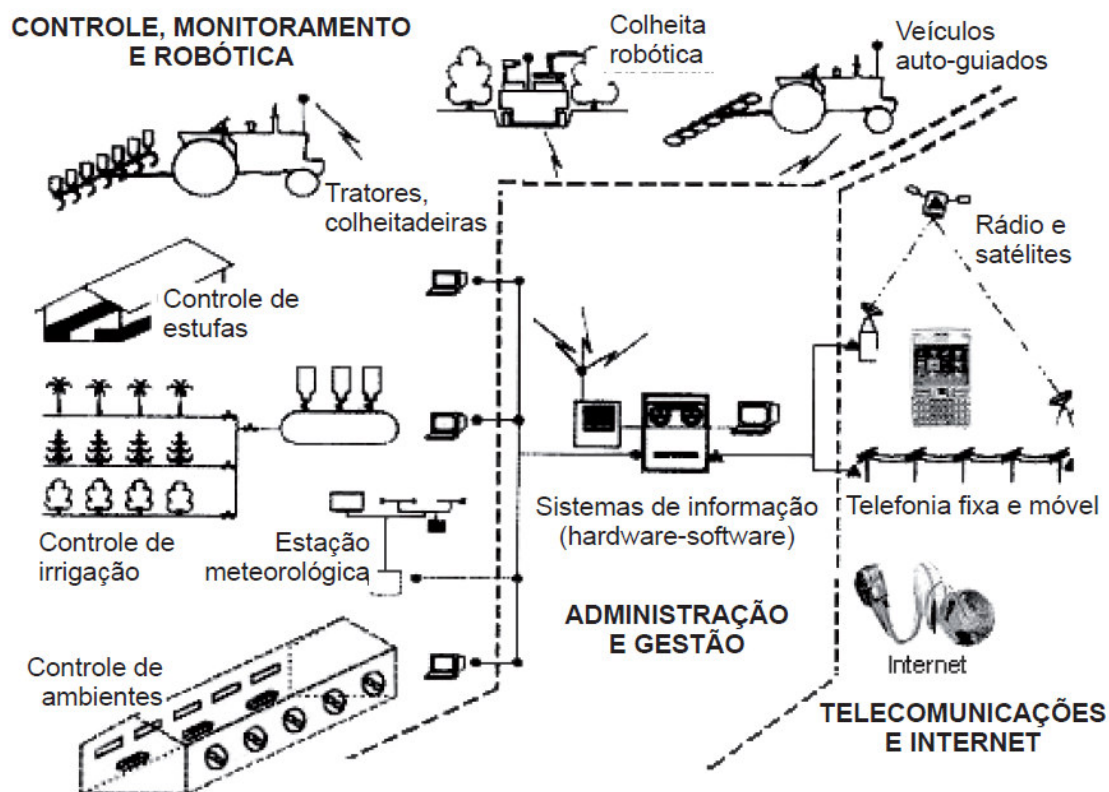


Figura 1 – Tecnologias da informação no agronegócio

Fonte: Arraes (1993), adaptado por Mendes, Oliveira e Santos (2011, p. 43).

A seguir esses grupos são explanados como subseções desta seção.

2.2.1 Tecnologias de administração e gestão

Como tecnologias de administração e gestão encontram-se as aplicações de caráter genérico denominados aplicativos de automação de escritórios como planilhas eletrônicas de cálculo, os sistemas gerenciadores de banco de dados, os processadores de texto e os geradores de apresentação. Também pertencem a esse grupo os aplicativos de objetivo gerencial e específico, que são utilizados para apoio às atividades administrativas e de acompanhamento e gestão de atividades produtivas. Nesse grupo estão os sistemas contábeis e financeiros, de controle de estoques, de gestão de recursos humanos, de *marketing* e comercialização de produtos e serviços; manutenção de veículos e equipamentos, transporte e logística, planejamento e gerenciamento de plantio e colheita; sistemas de suporte a decisão, pesquisa operacional, modelagem e otimização de produção, sistemas de gestão e controle de rebanhos (ZAMBALDE et al., 2011, p. 43).

Entre esses aplicativos estão os utilizados pelos produtores, os destinados às cooperativas, os armazéns e a agroindústrias. Esses aplicativos envolvem todos os níveis de gestão (operacional, gerencial ou tático e estratégico) e os mais diversos setores organizacionais (produção, financeiro, recursos humanos, comercialização, logística).

2.2.2 Tecnologias de controle, monitoramento e robótica

As tecnologias de controle, monitoramento e robótica incluem sensores, controladores, monitores e atuadores aplicados à agricultura e zootecnia de precisão e aos processos de gestão agroindustriais (ZAMBALDE et al, 2011).

Os sistemas de sensoriamento e monitoramento possuem as mais diversas aplicações nas atividades agrícolas e pecuárias, como, por exemplo, (CASTRO NETO; PINTO; COELHO, 2005; JESUS; ZAMBALDE, 1998; LOPES, 2010; MURAKAMI; SARAIVA, 2005): controle e monitoramento de tratores, implementos e colhedeiças; irrigação e drenagem; controle e monitoramento de ambiente; sistemas de autômatos (robôs) de ordenha e limpeza de ambientes e equipamentos; sistemas de rastreabilidade, tecnologias de identificação eletrônica e pesagem de animais; sistemas de informações geográficas e de posicionamento; limpeza, seleção, embalagem, manuseio e armazenamento de grãos; sistemas de controle de processos de secagem e secadores.

O uso de tecnologias da informação e comunicação que denominam agricultura de precisão permitem reduzir os custos, aumentar a produção, ajustar as entradas às necessidades do solo e das culturas, aumentar os rendimentos e reduzir os impactos ambientais (CASTRO NETO; PINTO; COELHO, 2005). Para esses autores o conceito de Agricultura de Precisão está, normalmente, associado à utilização de equipamento de alta tecnologia e software para realizar avaliação e monitoramento.

Na agricultura de precisão são utilizadas tecnologias da informação e comunicação para avaliar e/ou monitorar as condições em determinada área de solo, referenciar geograficamente essa área e implantar um processo específico de dosagem de fatores de produção (água, sementes, defensivos, adubação etc.). As principais tecnologias envolvidas são o Sistema de Posicionamento Global (GPS), os Sistemas de Informações Geográficas (SIG) e os sensores e atuadores eletrônicos (ZAMBALDE et al., 2011, CASTRO NETO; PINTO; COELHO, 2005).

Os objetivos da agricultura de precisão são: uso otimizado de insumos agrícolas; minimização de impactos ambientais; e maximização da qualidade, produtividade e do retorno financeiro (ZAMBALDE et al., 2011; ARVUS, 2014).

A Figura 2 apresenta as etapas abrangidas pela agricultura de precisão na preparação do solo, no plantio, no acompanhamento da lavoura e na colheita.



Figura 2 – Agricultura de precisão (AP)

Fonte: Arvus Tecnologia (2014, p. 1).

De acordo com o representado na Figura 2, as tecnologias de informação e comunicação estão presentes em todas as fases do ciclo de vida da produção agrícola. Além das atividades representadas nessa figura, a tecnologia é utilizada no melhoramento de sementes, defensivos e adubos. E na pós-colheita estão as tecnologias para limpeza, seleção, embalagem, armazenamento, transporte e beneficiamento dos grãos. Além dessas há as relacionadas à logística e os sistemas de apoio à decisão voltados para oferta, demanda e análise de mercados.

Além da agricultura as tecnologias de informação e comunicação aplicam precisão às atividades relacionadas à produção animal, denominada, zootecnia de precisão. Essa é uma prática realizada com o uso de tecnologia da informação para avaliar e/ou monitorar animais, processos e comportamentos em produção animal (ZAMBALDE et al., 2011, p. 44). Nas diversas fases e atividades relacionadas à produção animal podem ser empregadas tecnologias, como, por exemplo: identificação e rastreabilidade, monitoramento crescimento, produção de leite, carne, ovos, gestão da alimentação, controle de ciclo reprodutivo e medicamentos.

Nas Figuras 3 e 4 são apresentados, respectivamente, exemplos de práticas de zootecnia de precisão e de rastreabilidade.

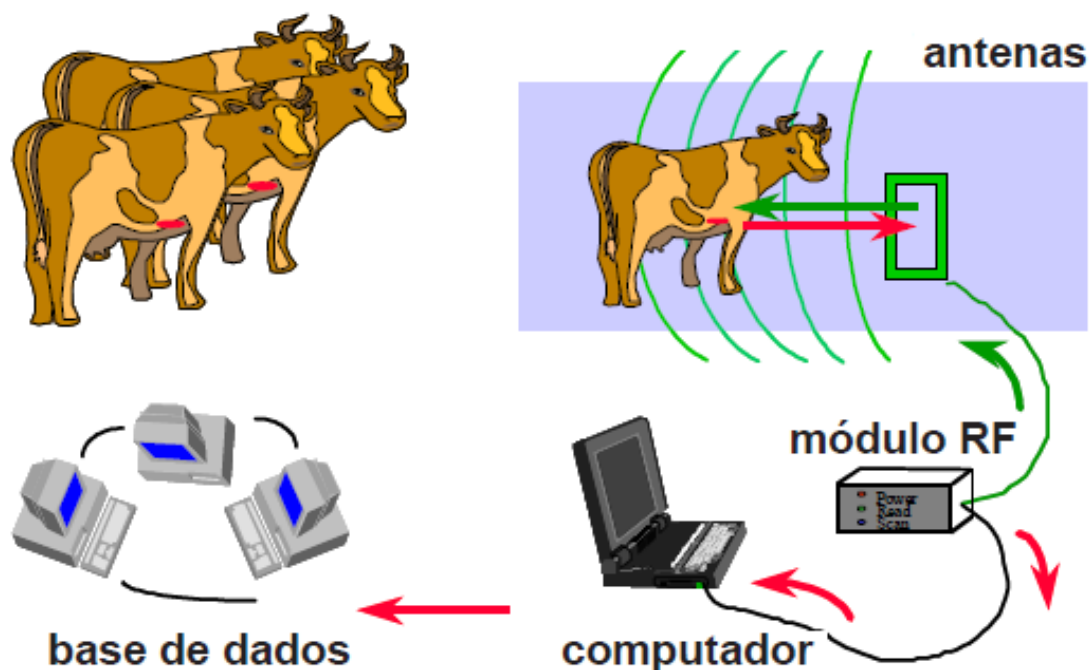


Figura 3 – Zootecnia de precisão

Fonte: Agriiência (2009) citado por Zambalde et al. (2011, p. 45).

Na Figura 3 está representada o monitoramento por meio de sistema por Rádio Frequência (RF). Essa forma de monitoramento é interessante pela possibilidade de transmissão sem fio. Uma etiqueta (*tag*) RFID (*Radio Frequency Identification*) pode ser colocada no animal e transmitir informações sobre o mesmo.

Na Figura 4 é exemplificada a rastreabilidade, que também pode ocorrer por meio de dispositivos equipados com rádio frequência e por meio de sistema de posicionamento global manter atualizados dados sobre localização do animal, por exemplo.



Figura 4 – Zootecnia de precisão e rastreabilidade

Fonte: Geers et al. (1993) citado por Furlaneto e Manzano (2010, p. 1).

2.2.3 Tecnologias de Telecomunicações e Internet

No grupo das tecnologias de telecomunicações e internet encontram-se as possibilidades de interação e integração de dados e a difusão de informação e conhecimento (ZAMBALDE et al., 2011, p. 44). Para Castro Neto, Pinto e Coelho (2005, p. 5), a utilização das telecomunicações, redes de contatos e fóruns de discussão poderão contribuir para o aumento da vantagem competitiva face à crescente concorrência.

Algumas das possíveis aplicações das tecnologias de telecomunicações e internet são (ZAMBALDE et al., 2011): criação e disponibilização de repositórios de informações sobre recursos naturais, dados climáticos, oferta e demanda de produtos, dentre outros; capacitação como cursos e treinamento, acesso a resultados pesquisa; sistemas de comércio *online* para marketing e comércio eletrônico em suas diversas formas; e aplicações móveis para agricultura e zootecnia de precisão.

A Figura 5 apresenta um esboço do uso de Tecnologias de Telecomunicações e Internet na propriedade agrícola.

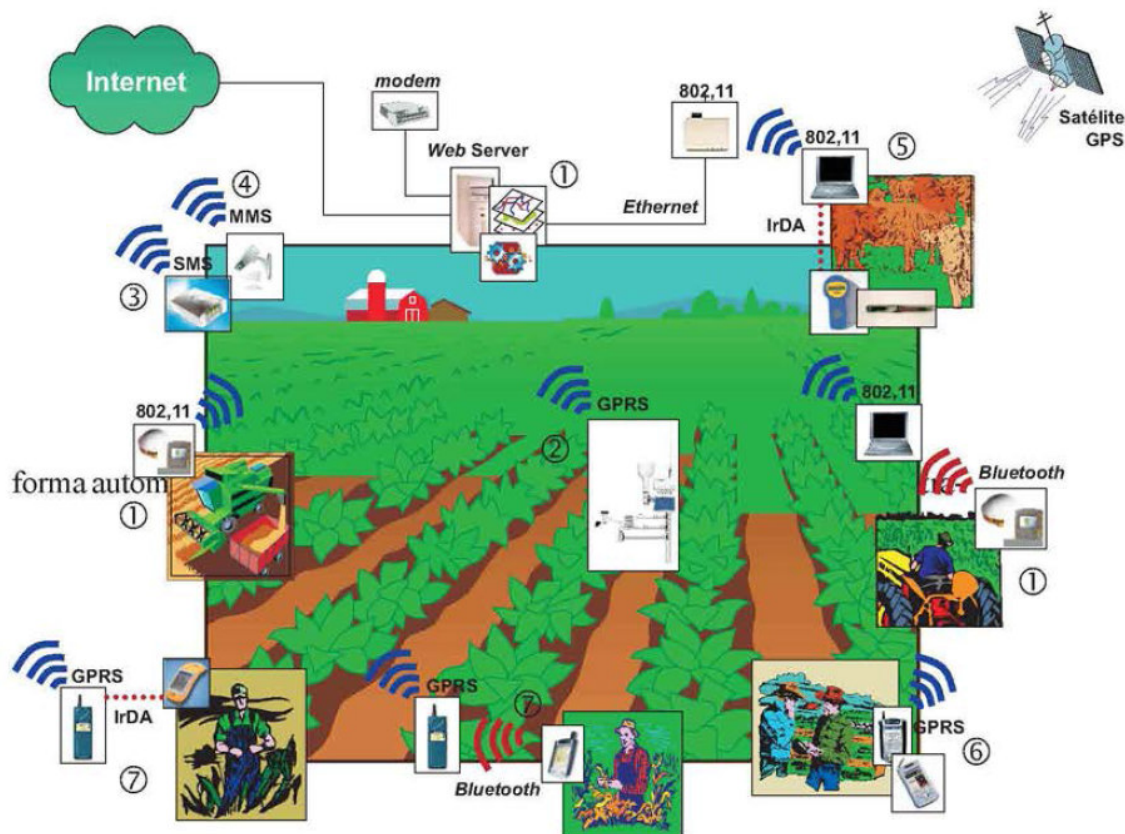


Figura 5 – m-Business na empresa agrícola

Fonte: Castro Neto, Pinto e Coelho (2005, p. 91).

No contexto dessas tecnologias - administração e gestão, controle, monitoramento e robótica e telecomunicações e Internet, o Grupo Consultivo Internacional em Pesquisa Agropecuária (GCIAR), citado por Bambini (2010), indica algumas áreas atuais de aplicação das tecnologias no agronegócio:

a) Repositórios de dados – as tecnologias de informação e comunicação possibilitam a geração de dados agrícolas e ambientais que podem ser armazenados com ou sem interação humana e disponibilizados para os mais diversos uso de suporte e apoio à tomada de decisão e planejamento.

b) Cálculos – as tecnologias possibilitam o gerenciamento, o compartilhamento e o processamento de grandes volumes de dados, a modelagem e a simulação, o processamento e a visualização de imagens que contribuem para o monitoramento e o melhoramento de culturas e animais.

c) Aplicações geoespaciais – possibilitam o gerenciamento, o processamento e a visualização de dados e informações relacionados à geografia e ao espaço. Esses dados fornecem subsídios para melhor planejamento do uso do solo e recursos naturais, aplicação de

insumos e defensivos agrícolas e mapeamento de áreas de degradação e perda de produtividade.

d) Sistemas de suporte à decisão e robótica – a análise de dados e projeções a partir de dados armazenados por meio de técnicas de mineração de dados e de Inteligência Artificial fornecem suporte para o processo de tomada de decisão. A robótica tem permitido automatizar a realização das mais diversas atividades agrícolas, sejam elas repetitivas ou perigosas.

e) As tecnologias em equipamentos e processos agrícolas – permitem o aumento da eficiência de equipamentos e processos sejam relacionados ao plantio, acompanhamento e monitoramento, produção animal, transporte, beneficiamento e armazenamento da produção.

f) Interconexão e aprendizado – as tecnologias de informação e comunicação têm possibilitado meios de conectar produtores, pesquisadores e demais envolvidos na cadeia produtiva setor agropecuário. As tecnologias contribuem para ampliar o acesso ao aprendizado e ao conhecimento.

3 MATERIAIS E MÉTODO

Neste capítulo são apresentados os materiais e o método utilizados na modelagem e na implementação do aplicativo desenvolvido como resultado deste trabalho.

3.1 MATERIAIS

O Quadro 1 apresenta as tecnologias e as ferramentas utilizadas para a modelagem e a implementação do sistema. Essas tecnologias envolvem as linguagens, os sistemas de banco de dados e a *Integrated Development Environment* (IDE) para implementação, além das ferramentas para a modelagem.

Tecnologia/Ferramenta	Aplicação
Astah Community	Modelagem do diagrama de casos de uso.
Linguagem Java	Desenvolvimento da aplicação.
NetBeans	Ambiente de desenvolvimento (IDE).
PostgreSQL	Banco de dados.
MySQL Workbench	Modelagem e composição do banco de dados.
iReport	Geração de relatórios.
Hibernate	Mapeamento objeto relacional

Quadro 1 – Tecnologias e ferramentas utilizadas

3.2 MÉTODO

O método é composto pelas principais atividades realizadas para o levantamento dos requisitos, a modelagem e a implementação do sistema.

Requisitos – o levantamento dos requisitos foi realizado a partir da observação das atividades realizadas na produção de leite e manejo de gado leiteiro em uma propriedade agrícola que realiza essas atividades. Conversas informais e acompanhamento das atividades foram realizados na propriedade que utilizará o sistema desenvolvido com o objetivo de levantar os interesses e as necessidades dos usuários definindo os requisitos do sistema.

No levantamento de requisitos foram identificadas as funcionalidades consideradas necessárias para o gerenciamento das atividades realizadas no manejo da ordenha e do rebanho. Juntamente com os requisitos funcionais foram identificados os requisitos não funcionais.

Análise e projeto – a modelagem dos requisitos identificados foi modelada gerando os diagramas para a representação das funcionalidades. Em seguida o diagrama do banco de dados com a definição das tabelas, seus campos e relacionamento, foi realizado. Uma visão dos formulários de entrada de dados e das operações realizadas foi possível obter a partir da modelagem desse diagrama. Os dados que seriam armazenados nessas tabelas permitiram, ainda, definir possíveis relatórios que seriam desenvolvidos.

Implementação – a implementação foi realizada utilizando o ambiente de desenvolvimento NetBeans com a linguagem Java e o banco de dados PostgreSQL. Para a geração dos relatórios foi utilizado o iReport.

Testes – os testes foram informais e realizados visando identificar erros de codificação e o atendimento aos requisitos. Testes de usuários e de funcionalidades serão realizados posteriormente quando da implantação do sistema na propriedade que serviu de base para o levantamento dos requisitos.

4 RESULTADO

Este capítulo apresenta o resultado da realização deste trabalho que é a modelagem e o desenvolvimento de um sistema para controle de gado leiteiro. No capítulo também é apresentada a implementação de operações básicas de um cadastro visando exemplificar o uso das tecnologias empregadas para o desenvolvimento posterior do sistema.

4.1 ESCOPO DO SISTEMA

O sistema proposto visa atender as necessidades dos produtores de gado leiteiro para manter um melhor controle do rebanho. Com o sistema será possível manter o controle de diversos setores do negócio, tais como:

a) Controle de receitas e despesas, gastos com a compra de medicamentos, insumos, materiais de limpeza, compra de animais, intervenções clínicas, venda de animais, venda do leite, controle de estoque e etc.

b) Controle sobre a produção de leite, mantendo um histórico da produção e da qualidade do leite que é obtida pelas análises realizadas a partir de amostras. Os dados das análises realizadas são mantidos armazenados no sistema.

c) Gerenciamento da sanidade dos animais mantendo um histórico de intervenções medicamentosas de cada animal.

A funcionalidade de negócio principal do sistema é o armazenamento de dados da ordenha, com a possibilidade de inclusão de quantidade de leite produzida por ordenha e por animal ou da média por animal por ordenha. As análises realizadas com as amostras de leite também estão relacionadas à ordenha. Além disso, o controle das interações medicamentosas auxilia a identificar as vacinas ministradas em cada animal. O registro da composição racial juntamente com a possibilidade de armazenamento de produção individual por animal permite identificar as raças ou combinações raciais mais produtivas, por exemplo.

4.2 MODELAGEM DO SISTEMA

O Quadro 2 apresenta a listagem dos requisitos funcionais identificados para o sistema.

Identificação	Nome	Descrição
RF01	Cadastrar rebanhos	O rebanho contém vários animais. Esse cadastro auxilia na obtenção de um controle mais preciso sobre um grupo específico de animais.
RF02	Cadastrar animais	Cadastro de animais, associando cada animal a um rebanho. Um animal possui uma composição racial e uma pelagem específica associados. Ambos provenientes de cadastros realizados no sistema
RF03	Cadastrar tanque	Cadastrar tanques de armazenamento de leite.
RF04	Cadastro de produtos	Produtos utilizados na alimentação dos animais, materiais de higiene e limpeza, vacinas e medicamentos e outros utilizados na manutenção do rebanho e na realização das atividades de ordenha. Produto pode ter uma marca associada que é proveniente de um cadastro próprio.
RF05	Cadastrar pessoas	Cadastro dos dados básicos para funcionário, fornecedor, cliente. Há um cadastro de tipo de pessoa que é utilizado quando do cadastro de pessoa para definir o seu papel (cliente, fornecedor) no sistema ou como usuário do sistema (funcionário, administrador).
RF06	Realizar controle de despesas	Lançamento de contas a pagar, tais como luz, água, telefone, intervenções clínicas, assistência de veterinário, medicamentos, compra de animais, produtos de limpeza, pagamento de funcionários, produção de silagem, compra de maquinário, insumos, manutenção de maquinário e instalações.
RF07	Cadastrar doenças	Registro de doenças que possam acometer os animais.
RF08	Realizar controle de vacinação	Manter um histórico por animal de todas as ocorrências de doenças e vacinas aplicadas.
RF09	Realizar controle de receitas	Lançamentos de receitas obtidas da produção do leite, venda de matrizes e bezerros e outros.
RF10	Registrar dados da ordenha	Registro da quantidade de leite por animal com data e horário da ordenha. As ordenhas estão relacionadas à coleta de leite que indica para quem, quantidade e data e hora que o leite ordenhado foi entregue (vendido). Em geral essa coleta é realizada por um transportador vinculado a um laticínio, considerado cliente do produtor.
RF11	Realizar controle de estoque	Permitir controlar o estoque dos produtos utilizados na propriedade como insumos, medicamentos e produtos de limpeza.
RF12	Emitir relatórios	Emitir relatório com filtros que possam auxiliar no gerenciamento da atividade e auxiliar na tomada de decisão.
RF13	Cadastrar usuários	Usuários com acesso ao sistema. Os usuários possuem permissões distintas. Administrador com acesso a todas as funcionalidades do sistema. Funcionário com acesso a todas as funcionalidades do sistema, exceto manutenção de usuários.

Quadro 2 – Requisitos funcionais

A listagem do Quadro 3 apresenta os requisitos não-funcionais identificados para o sistema, também denominados de requisitos suplementares. Os requisitos não funcionais

podem explicitar regras de negócio, restrições ao sistema de acesso, requisitos de qualidade, desempenho e segurança, dentre outros.

Identificação	Nome	Descrição
RNF01	Acesso ao sistema	O acesso ao sistema será realizado por meio de <i>login</i> e senha, com usuário previamente cadastrado.
RNF02	Cadastro de animais	Deve ser permitida a alteração do número do brinco dos animais. Um mesmo brinco pode ser reusado por animais distintos, mas não reusado pelo mesmo animal. Isso significa que um brinco será utilizado por outro animal quando o animal que estava utilizando o brinco é vendido, falece ou outras situações que fazem com ele não pertença mais ao rebanho leiteiro.
RNF03	Coleta do leite	A coleta do leite é realizada pelo cliente e deve estar relacionada a um tanque.
RNF04	Ordenha dos animais	Deve permitir informar a quantidade total de leite coletada e selecionar quais animais foram ordenhados. Ou caso queira se ter um controle mais preciso permitir informar a quantidade ordenhada de cada animal.
RNF05	Quantidade de leite de cada ordenha	Informar o total de leite obtido por evento de ordenha ou por animal. A quantidade por animal pode ser obtida pela média do evento de ordenha.

Quadro 3 – Requisitos não funcionais

A Figura 6 apresenta o diagrama de casos de uso identificados para o sistema. O sistema terá dois atores: o Funcionário com acesso a todas as funcionalidades relacionadas ao negócio e o Administrador que terá acesso a todas as funcionalidades do sistema. O Administrador, como apresentado no diagrama de casos de uso da Figura 6, herda as funcionalidades de funcionário e possui permissão exclusiva para a inclusão de novos usuários no sistema. O sistema é composto basicamente por cadastros e registro de dados de ordenhas e análises amostras de leite que são realizadas. O controle de estoque e de receitas e de despesas tem como base os dados de cadastros mantidos no sistema.

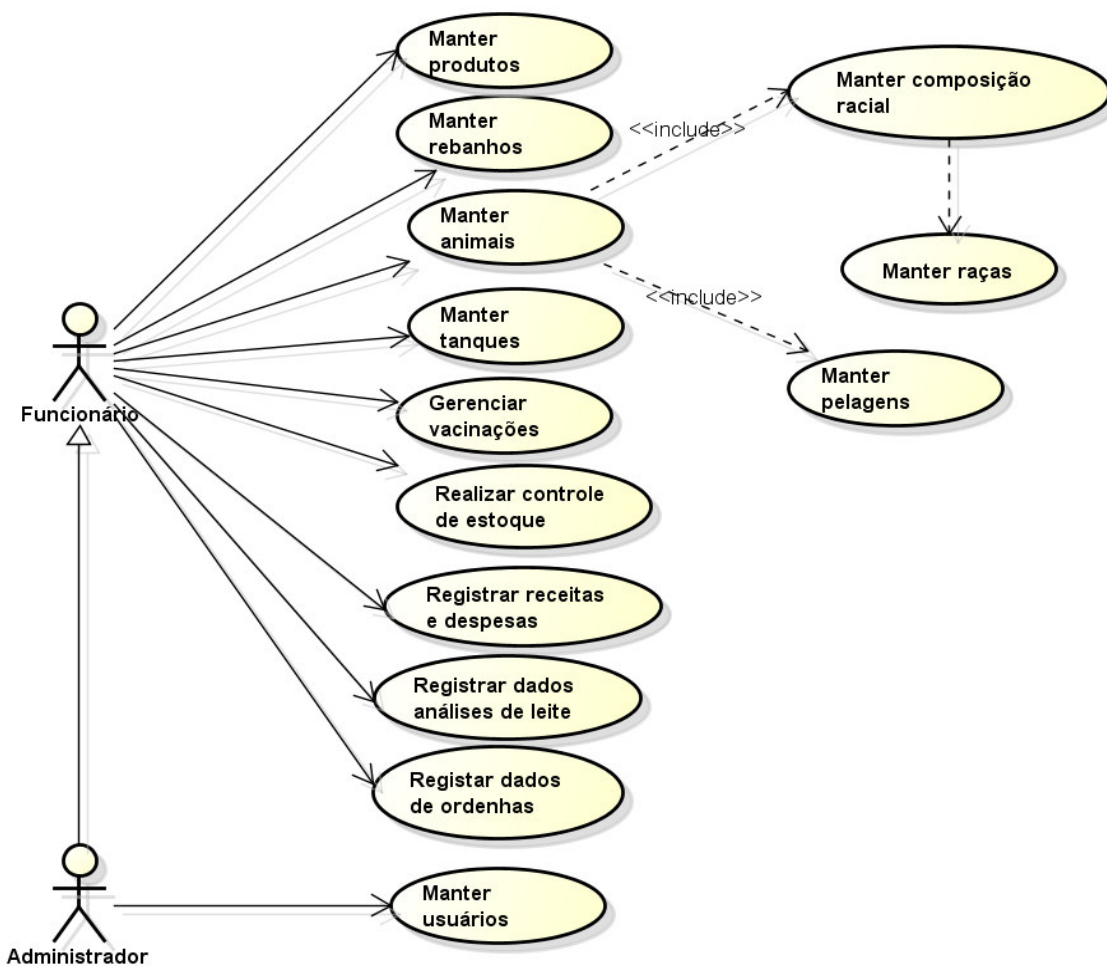


Figura 6 – Diagrama de casos de uso

O Quadro 4 apresenta a operação incluir dos casos de uso de cadastros. Essa expansão de caso de uso é aplicada para todas as funcionalidades do sistema representadas na Figura 6 que se referem aos cadastros.

Caso de uso:

Cadastrar insumos, rebanhos, animais, tanques, vacina, usuários e outros.

Descrição:

Efetuar um cadastro, incluindo os dados necessários a serem preenchidos em determinado formulário.

Evento Iniciador:

Formulário para incluir os dados em edição.

Atores:

Funcionário, Administrador.

Pré-condição:

Possuir acesso no sistema para poder efetuar o cadastro.

Sequência de Eventos:

1. O usuário insere os dados necessários para determinado cadastro. Inseridos os dados o usuário pressiona o botão salvar.
2. O sistema inclui os dados do respectivo formulário no banco de dados e informa que a operação foi realizada com sucesso.

Pós-Condição:

Dados do cadastro inseridos no banco de dados.	
Nome do fluxo alternativo (extensão)	Descrição
2 Dados não são válidos.	2.1 No momento de incluir os dados, o sistema faz a verificação e constata que há dados inválidos. É emitida uma mensagem. Retorna para o formulário de cadastro em estado de edição.

Quadro 4 – Operação incluir dos casos de uso de cadastro

O Quadro 5 apresenta a operação de alteração para os casos de uso de cadastro.

<p>Caso de uso: Cadastrar produtos, rebanhos, animais, tanques, vacinas, usuários e outros.</p> <p>Descrição: Alterar dados de um cadastro.</p> <p>Evento Iniciador: Cadastro a ser alterado aberto para edição.</p> <p>Atores: Funcionário, Administrador.</p> <p>Pré-condição: Formulário para alteração dos dados do cadastro em edição.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. O usuário altera os dados necessários e em seguida pressiona o botão salvar. 2. Sistema salva os dados, realizando uma operação de atualização no cadastro e informa que a operação foi realizada com sucesso. <p>Pós-Condiciono: Dados do cadastro alterado salvo no banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
2 Dados não são válidos.	2.1 No momento de alterar os dados, o sistema faz a verificação e constata que há dados inválidos. É emitida mensagem. Retorna para o formulário de cadastro em estado de edição.

Quadro 5 – Operação alterar dos casos de uso de cadastro

O Quadro 6 detalha a operação de exclusão dos casos de uso de cadastro.

<p>Caso de uso: Cadastrar produtos, rebanhos, animais, tanques, vacinas, usuários e outros.</p> <p>Descrição: Excluir um cadastro.</p> <p>Evento Iniciador: Cadastro para ser excluído selecionado.</p> <p>Atores: Funcionário, Administrador.</p> <p>Pré-condição: Cadastro para excluir selecionado e esse cadastro não possuir relacionamentos que o impeçam de ser excluído.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. O usuário pressiona o botão excluir. 2. O sistema solicita confirmação para exclusão. 3. O usuário confirma a exclusão. 4. O sistema exclui o cadastro informado que a operação foi realizada. <p>Pós-Condiciono: Cadastro excluído do banco de dados.</p>	
Nome do fluxo alternativo (extensão)	Descrição
3 Cadastro a ser excluído possui	3.1 O banco de dados será modelado da forma de exclusão "drop cascade", ou seja, quando o usuário deseja excluir o

dependências.	registro pai, o sistema apresentará uma mensagem dizendo que serão excluídos todos os registros filhos que estão vinculados à tabela pai.
	3.2 Se o usuário confirmar a exclusão, os registro vinculados serão excluídos.
	3.2 Se o usuário optar por não continuar a operação ele terá a opção de cancelar.

Quadro 6 – Operação excluir dos casos de uso de cadastro

O Quadro 7 detalha a operação de consulta dos casos de uso de cadastro. Outras consultas são realizadas como relatórios.

<p>Caso de uso: Cadastrar produtos, rebanhos, animais, tanques, vacinas, usuários e outros.</p> <p>Descrição: Consultar dados de um cadastro.</p> <p>Evento Iniciador: Listagem dos cadastros apresentada em um <i>grid</i>.</p> <p>Atores: Funcionário, Administrador.</p> <p>Pré-condição: Existência de cadastros no banco de dados.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. O usuário seleciona o cadastro que quer consultar. 2. O sistema abre o cadastro em modo de edição. <p>Pós-Condiciono: Dados do cadastro são apresentados no formulário.</p>

Quadro 7 – Operação consulta dos casos de uso de cadastro

O Quadro 8 detalha o caso de uso realizar controle de estoque.

<p>Caso de uso: Realizar controle de estoque.</p> <p>Descrição: Realizar a inclusão de dados para entrada saída de quantidade de itens em estoque.</p> <p>Evento Iniciador: Entrada ou saída para itens de estoque.</p> <p>Atores: Funcionário, Administrador.</p> <p>Pré-condição: Necessidade de entrada ou saída de quantidade de itens cadastrados no estoque.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. O usuário seleciona o item do qual quer ajustar a quantidade em estoque. 2. O sistema abre o cadastro em modo de edição. 3. Usuário atualiza a quantidade: entrada ou saída. 4. Sistema ajusta a quantidade do referido item no respectivo cadastro. <p>Pós-Condiciono: Dados do cadastro são apresentados no formulário.</p>
--

Quadro 8 – Realizar controle de estoque

O Quadro 9 apresenta a expansão do caso de uso registrar dados da ordenha.

<p>Caso de uso: Registrar dados da ordenha.</p> <p>Descrição:</p>

<p>Realizar a inclusão de dados de ordenhas realizadas.</p> <p>Evento Iniciador: Inclusão de dados de ordenhas.</p> <p>Atores: Funcionário, Administrador.</p> <p>Pré-condição: Necessidade de inclusão de dados relacionados à ordenha realizada.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. O usuário abre o formulário para a inclusão de dados de ordenha. 2. O sistema abre o cadastro em modo de edição. 3. Usuário inclui os dados e solicita que eles sejam salvos. 4. Sistema inclui os dados no banco, informado o usuário da realização da operação. <p>Pós-Condição: Dados de ordenha incluídos no banco de dados.</p>

Quadro 9 – Inclusão de dados de ordenha

O Quadro 10 apresenta a expansão do caso de uso registrar dados de análises de leite realizadas.

<p>Caso de uso: Registrar dados de análises de amostras de leite realizadas.</p> <p>Descrição: Realizar a inclusão de dados de análises de leite realizadas.</p> <p>Evento Iniciador: Inclusão de dados de análises de leite realizadas.</p> <p>Atores: Funcionário, Administrador.</p> <p>Pré-condição: Necessidade de inclusão de dados relacionados às análises de amostras de leite.</p> <p>Sequência de Eventos:</p> <ol style="list-style-type: none"> 1. O usuário abre o formulário para a inclusão de dados análise de leite realizadas. 2. O sistema abre o cadastro em modo de edição. 3. Usuário inclui os dados e solicita que eles sejam salvos. 4. Sistema inclui os dados no banco, informado o usuário da realização da operação. <p>Pós-Condição: Dados de análises de leite incluídos no banco de dados.</p>
--

Quadro 10 – Inclusão de dados de análises de leite

A Figura 7 apresenta o diagrama de entidades e relacionamentos do banco de dados do sistema. Por meio desse diagrama é possível verificar que um animal pertence a um rebanho e que há um controle de vacinação para cada animal. Na tabela de ordenhas é realizado um vínculo com a tabela animal permitindo identificar a quantidade ordenhada de cada animal. Caso não seja possível ou o usuário não queria armazenar a quantidade individual de cada animal é possível armazenar para cada animal a média obtida em cada evento de ordenha. A tabela pessoa armazena dados genéricos de clientes, fornecedores e funcionários. O controle de estoque é realizado por produtos que podem ser materiais de limpeza, insumos e medicamentos. A análise do leite é realizada por tanque e os resultados de cada análise realizada são armazenados. Há, ainda, um controle básico de receitas e despesas e de estoque.

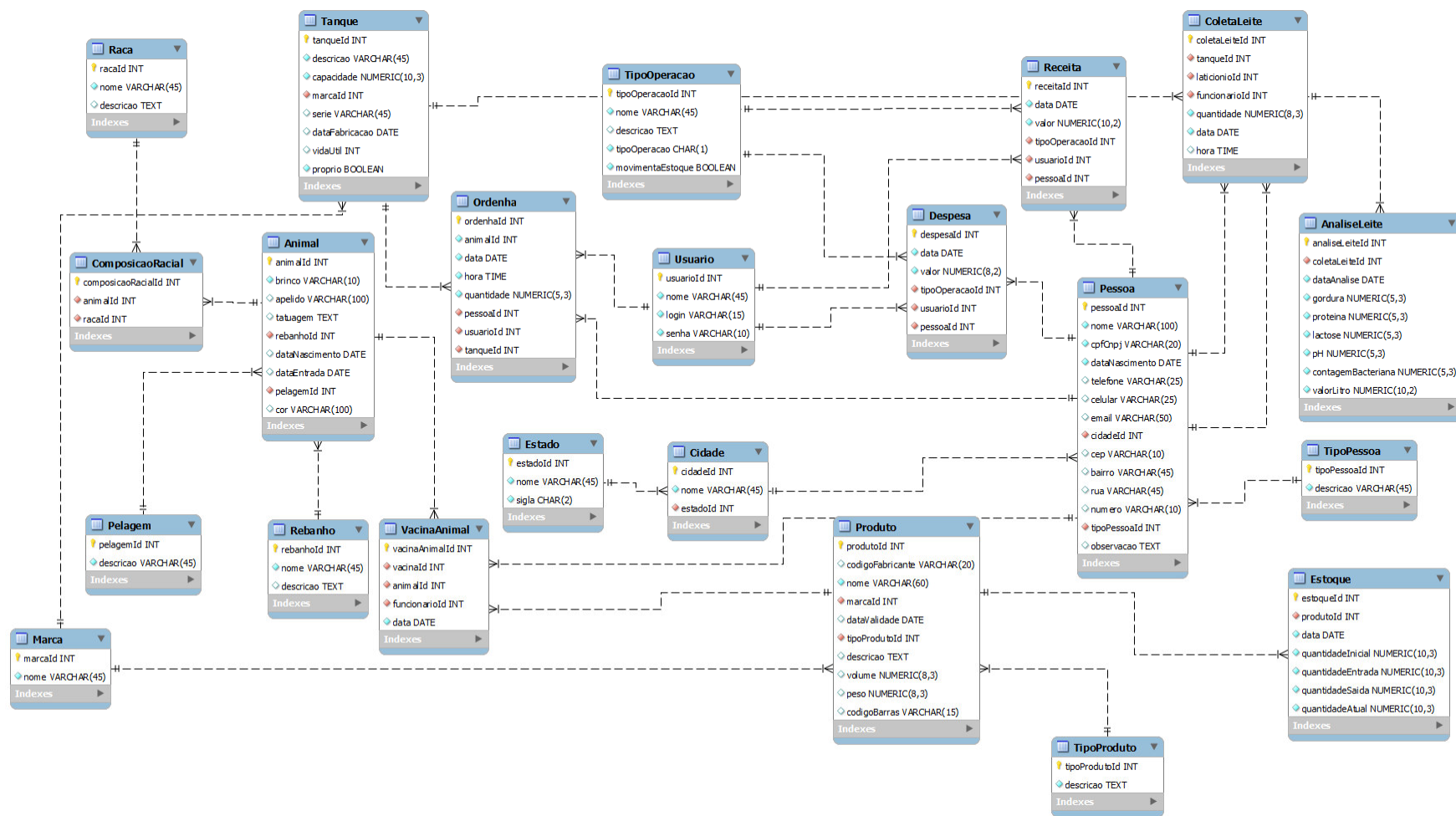


Figura 7 – Diagrama de entidades e relacionamentos

A seguir está a descrição das tabelas do banco de dados apresentadas na Figura 7.

No Quadro 11 estão descritos os campos da tabela Usuários que são as pessoas que possuem acesso ao sistema.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
usuarioId	Numérico	Não	Sim	Não	
nome	Texto	Não	Não	Não	
login	Texto	Não	Não	Não	
senha	Texto	Não	Não	Não	

Quadro 11 – Campos da tabela Usuario

A tabela Pessoa armazena dados comuns entre funcionários, usuários, fornecedores, clientes e outros. Os campos da tabela Pessoas estão apresentados no Quadro 12.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
pessoaId	Numérico	Não	Sim	Não	
nome	Texto	Não	Não	Não	
CpfCnpj	Texto	Não	Não	Não	
dataNascimento	Data	Não	Não	Não	
telefone	Texto	Sim	Não	Não	
celular	Texto	Sim	Não	Não	
email	Texto	Sim	Não	Não	
cidadeId	Numérico	Não	Não	Sim	Da tabela Cidade
cep	Texto	Sim	Não	Não	
bairro	Texto	Sim	Não	Não	
rua	Texto	Sim	Não	Não	
numero	Texto	Sim	Não	Não	
tipoPessoaId	Numérico	Não	Não	Sim	Da tabela TipoPessoa
observacao	Texto	Sim	Não	Não	

Quadro 12 – Campos da tabela Pessoa

Os tipos de pessoas são utilizados para identificar se fornecedor, cliente, usuário do sistema. O Quadro 13 apresenta os campos da tabela TipoPessoa.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
tipoPessoaId	Numérico	Não	Sim	Não	
descricao	Texto	Não	Não	Não	

Quadro 13 – Campos da tabela TipoPessoa

A tabela Animal (campos dessa tabela descritos no Quadro 14 apresenta os dados relativos a cada animal. Um animal pertence a um rebanho.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
animalId	Numérico	Não	Sim	Não	
brinco	Texto	Não	Não	Não	
apelido	Texto	Sim	Não	Não	
tatuagem	Imagem	Sim	Não	Não	
rebanhoId	Numérico	Não	Não	Sim	Da tabela Rebanho
dataNascimento	Data	Sim	Não	Não	
dataEntrada	Data	Sim	Não	Não	
pelagemId	Numérico	Não	Não	Sim	Da tabela Pelagem
cor	Texto	Sim	Não	Não	

Quadro 14 – Campos da tabela Animal

Os campos da tabela Rebanho são apresentados no Quadro 15.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
rebanhoId	Numérico	Não	Sim	Não	
nome	Texto	Não	Não	Não	
descricao	Texto	Sim	Não	Não	

Quadro 15 – Campos da tabela Rebanho

A tabela Ordenhas (Quadro 16) armazena os dados relativos aos eventos de ordenha realizados.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
ordenhaId	Numérico	Não	Sim	Não	
animalId	Numérico	Não	Não	Sim	Da tabela Animal
data	Data	Não	Não	Não	
hora	Hora	Não	Não	Não	
quantidade	Numérico	Não	Não	Não	Quantidade produzida por cada animal ou média do total do evento
peessoaId	Numérico	Não	Não	Sim	Da tabela Pessoa, tipo funcionário
usuarioId	Numérico	Não	Não	Sim	Da tabela Usuário. É o usuário que está autenticado no sistema
tanqueId	Numérico	Não	Não	Sim	Da tabela Tanque

Quadro 16 – Campos da tabela Ordenha

Os campos da tabela VacinaAnimal são apresentados no Quadro 17. Essa tabela registra os dados de vacinas ministradas nos animais.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
vacinaAminalId	Numérico	Não	Sim	Não	
vacinaId	Numérico	Não	Não	Sim	Da tabela Produto, tipo vacina
animalId	Numérico	Não	Não	Sim	Da tabela Animal
funcionarioId	Numérico	Não	Não	Sim	Da tabela Pessoa, tipo funcionário
data	Data	Não	Não	Não	

Quadro 17 – Campos da tabela VacinaAnimal

Os dados de produtos sejam insumos, material de limpeza, medicamentos e outros são armazenados na tabela Produto. Os campos dessa tabela são apresentados no Quadro 18.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
produtoId	Numérico	Não	Sim	Não	
codigoFabricante	Texto	Sim	Não	Não	
nome	Texto	Não	Não	Não	
marcaId	Numérico	Não	Não	Sim	Da tabela Marca
dataValidade	Data	Sim	Não	Não	
tipoProdutoId	Numérico	Não	Não	Sim	Da tabela TipoProduto
descricao	Texto	Sim	Não	Não	
volume	Numérico	Sim	Não	Não	
peso	Numérico	Sim	Não	Não	
codigoBarras	Texto	Sim	Não	Não	

Quadro 18 – Campos da tabela Produto

Os campos da tabela Marca são apresentados no Quadro 19. Os dados dessa tabela são usados no cadastro de produtos para identificar a marca dos mesmos.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
marcaId	Numérico	Não	Sim	Não	
descricao	Texto	Não	Não	Não	

Quadro 19 – Campos da tabela Marca

Os campos da tabela TipoProduto são apresentados no Quadro 20.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
tipoProdutoId	Numérico	Não	Sim	Não	
descricao	Texto	Não	Não	Não	

Quadro 20 – Campos da tabela TipoProduto

Os dados sobre estoque de produtos são armazenados na tabela Estoque. Os campos dessa tabela estão no Quadro 21.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
estoqueId	Numérico	Não	Sim	Não	
produtoId	Numérico	Não	Não	Sim	Da tabela Produto
data	Data	Não	Não	Não	
quantidadeInicial	Numérico	Não	Não	Não	
quantidadeEntrada	Numérico	Não	Não	Não	
quantidadeSaida	Numérico	Não	Não	Não	
quantidadeAtual	Numérico	Não	Não	Não	

Quadro 21 – Campos da tabela Estoque

As receitas obtidas com a venda de leite e animais, por exemplo, são registradas na tabela Receita (Quadro 22).

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
receitaId	Numérico	Não	Sim	Não	
data	Data	Não	Não	Não	
valor	Numérico	Não	Não	Não	
tipoOperacaoId	Numérico	Não	Não	Sim	Da Tabela TipoOperacao
usuarioId	Numérico	Não	Não	Sim	Da tabela Usuario é o usuário que está autenticado no sistema e inserindo os dados
pessoaId	Numérico	Não	Não	Sim	Da tabela Pessoa significa a pessoa origem da receita (cliente)

Quadro 22 – Campos da tabela Receita

Os campos da tabela TipoOperacao são apresentados no Quadro 23.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
tipoOperacaoId	Numérico	Não	Sim	Não	
Nome	Texto	Não	Não	Não	
descricao	Texto	Sim	Não	Não	
tipoOperacao	Texto	Não	Não	Não	Campo único R – Receita e D – Despesa.
movimentaEstoque	Lógico	Não	Não	Não	

Quadro 23 – Campos da tabela TiposOperacao

Os dados registrados das despesas, armazenados na tabela Despesa, são apresentados no Quadro 24.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
despesaId	Numérico	Não	Sim	Não	
data	Data	Não	Não	Não	
valor	Numérico	Não	Não	Não	
tipoOperacaoId	Numérico	Não	Não	Sim	Da Tabela TipoOperacao
pessoaId	Numérico	Não	Não	Sim	Da tabela Pessoa, significa a pessoa vinculada à despesa
usuarioId	Numérico	Não	Não	Sim	Da tabela Usuários. É o usuário que está autenticado no sistema

Quadro 24 – Campos da tabela Despesa

Os campos da tabela Tanque são apresentados no Quadro 25.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
tanqueId	Numérico	Não	Sim	Não	
descricao	Texto	Não	Não	Não	
capacidade	Numérico	Não	Não	Não	
marcaId	Numérico	Não	Não	Sim	Da tabela Marca
Serie	Texto	Sim	Não	Não	
dataFabricacao	Data	Sim	Não	Não	
vidaUtil	Inteiro	Sim	Não	Não	
próprio	Lógico	Não	Não	Não	Indica se o tanque é próprio ou não

Quadro 25 – Campos da tabela Tanque

Os campos da tabela ColetaLeite que se referem aos dados dos animais ordenhados em cada evento de ordenha e são apresentados no Quadro 26. O campo de hora permite identificar a ordem da coleta realizada no dia, por exemplo, primeira, segunda ou terceira.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
coletaLeiteId	Numérico	Não	Sim	Não	
tanqueId	Numérico	Não	Não	Sim	Da tabela Tanque
laticionioId	Númerico	Não	Não	Sim	Tabela Pessoa
funcionarioId	Numérico	Não	Não	Sim	Tabela Pessoa
quantidade	Numérico	Não	Não	Não	
data	Data	Não	Não	Não	
hora	time	Sim	Não	Não	

Quadro 26 – Campos da tabela ColetaLeite

Os campos da tabela AnaliseLeite que estão no Quadro 27 armazenam dados de análises realizadas de amostras de leite.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
analiseLeiteId	Numérico	Não	Sim	Não	
coletaLeiteId	Numérico	Não	Não	Sim	Da tabela ColetaLeite
dataAnalise	Data	Não	Não	Não	
gordura	Numérico	Não	Não	Não	
proteína	Numérico	Não	Não	Não	
lactose	Numérico	Não	Não	Não	
pH	Numérico	Não	Não	Não	
contagemBacteriana	Numérico	Não	Não	Não	
valorLitro	Numérico	Não	Não	Não	

Quadro 27 – Campos da tabela AnaliseLeite

Os campos da tabela Raca armazenam a descrição das raças dos animais. Os campos dessa tabela estão apresentados no Quadro 27.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
racaId	Numérico	Não	Sim	Não	
nome	Texto	Não	Não	Não	
descricao	Texto	Sim	Não	Não	

Quadro 27 – Campos da tabela Raca

Os campos da tabela ComposicaoRacial apresentados no Quadro 28 armazenam os dados referentes às raças de cada animal.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
composicaoRacialId	Numérico	Não	Sim	Não	
animalId	Numérico	Não	Não	Sim	Da tabela Animal
racaId	Numérico	Não	Não	Sim	Da tabela Raca

Quadro 28 – Campos da tabela ComposicaoRacial

O Quadro 29 apresenta os campos da tabela Pelagem armazenam as informações referentes ao tipo de pelagem dos animais.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
pelagemId	Numérico	Não	Sim	Não	
descricao	Texto	Não	Não	Não	

Quadro 29 – Campos da tabela Pelagem

Os campos da tabela Cidade, apresentados no Quadro 30, armazenam dados cidades que são utilizadas no cadastro de pessoas.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
cidadeId	Numérico	Não	Sim	Não	
nome	Texto	Não	Não	Não	
estadoId	Numérico	Não	Não	Sim	Da tabela Estado

Quadro 30 – Campos da tabela Cidade

A tabela Estado armazena os Estados (unidades de federação) utilizados no cadastro de cidades. A descrição dos campos dessa tabela está no Quando 31.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
estadoId	Numérico	Não	Sim	Não	
nome	Texto	Não	Não	Não	
sigla	Texto	Não	Não	Não	

Quadro 31 – Campos da tabela Estado

4.3 APRESENTAÇÃO DO SISTEMA

O acesso ao sistema ocorre por meio de *login* e senha que são cadastrados pelo administrador. O usuário de acesso é gravado em todas as operações realizadas, a fim de manter um histórico para consultas posteriores. A Figura 8 apresenta a tela *login*, ou seja, de autenticação, no sistema.



A tela de autenticação ao sistema apresenta um cabeçalho preto com o texto "Login de Acesso" em branco. Abaixo, há dois campos de entrada: o primeiro contém um ícone de usuário e o texto "Nome de Usuário"; o segundo contém um ícone de cadeado e o texto "senha". Na base da tela, há dois botões: um preto com o texto "Entrar" em branco e um laranja com o texto "Sair" em branco.

Figura 8 – Tela de autenticação ao sistema

Após a autenticação, o usuário tem acesso à tela principal do sistema. Por meio dessa tela todas as funcionalidades do sistema podem ser acessadas. Elas se encontram na parte superior da tela em um menu subdividido entre os diversos grupos de funcionalidades do sistema. A Figura 9 apresenta a tela principal do sistema.

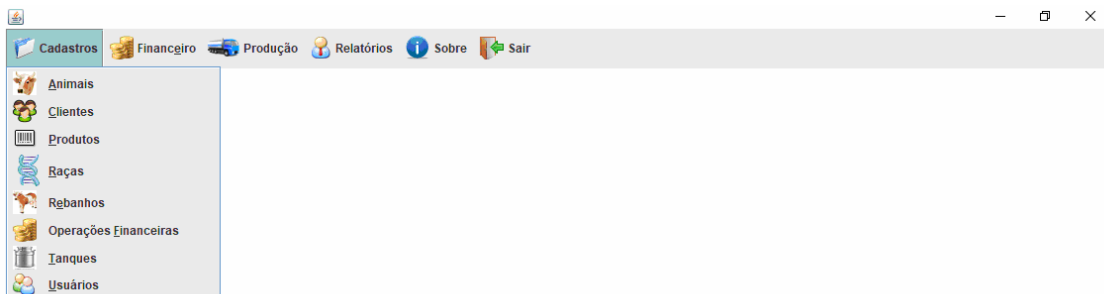
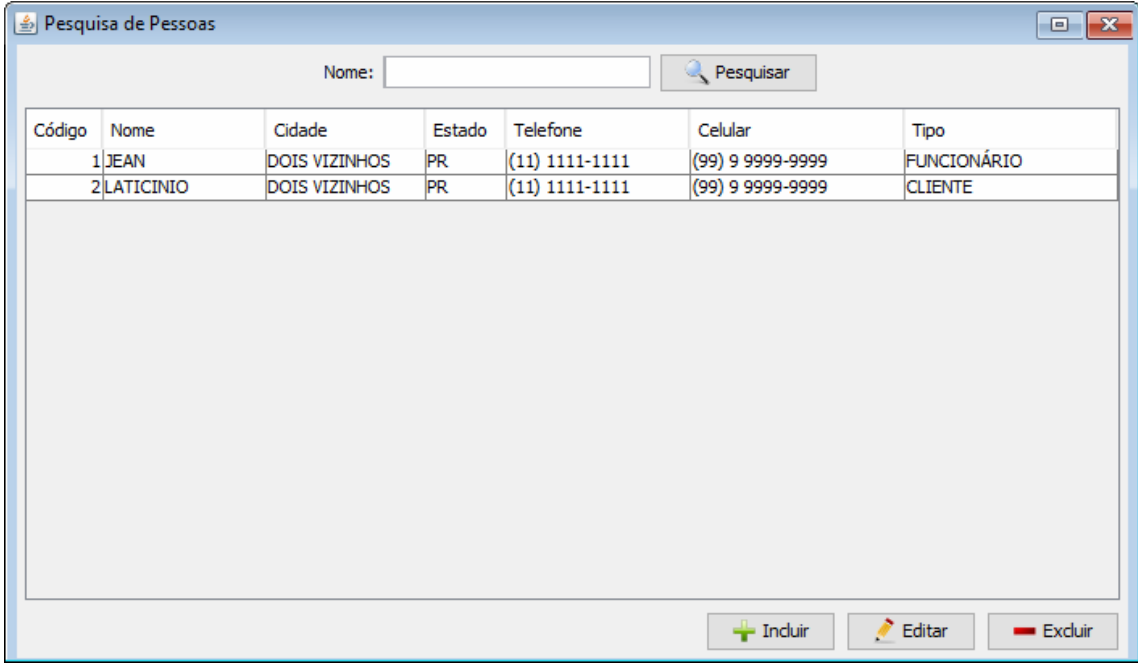


Figura 9 – Tela principal

Ao acessar qualquer submenu do sistema, tem-se acesso à tela de pesquisa da entidade selecionada. Nas telas de pesquisa são disponibilizadas as operações básicas de cadastro como

inclusão, alteração e exclusão de registros. A Figura 10 apresenta a tela de pesquisa de pessoas. Na parte inferior da tela são apresentados os botões para incluir, editar e excluir registros. Esse é o padrão adotado de todas as telas de cadastro do sistema.



The screenshot shows a window titled "Pesquisa de Pessoas". At the top, there is a search form with a text input labeled "Nome:" and a "Pesquisar" button. Below the search form is a table with the following data:

Código	Nome	Cidade	Estado	Telefone	Celular	Tipo
1	JEAN	DOIS VIZINHOS	PR	(11) 1111-1111	(99) 9 9999-9999	FUNCIONÁRIO
2	LATICINIO	DOIS VIZINHOS	PR	(11) 1111-1111	(99) 9 9999-9999	CLIENTE

At the bottom of the window, there are three buttons: "Incluir" (with a green plus icon), "Editar" (with a yellow pencil icon), and "Excluir" (with a red minus icon).

Figura 10 – Pesquisa de pessoas

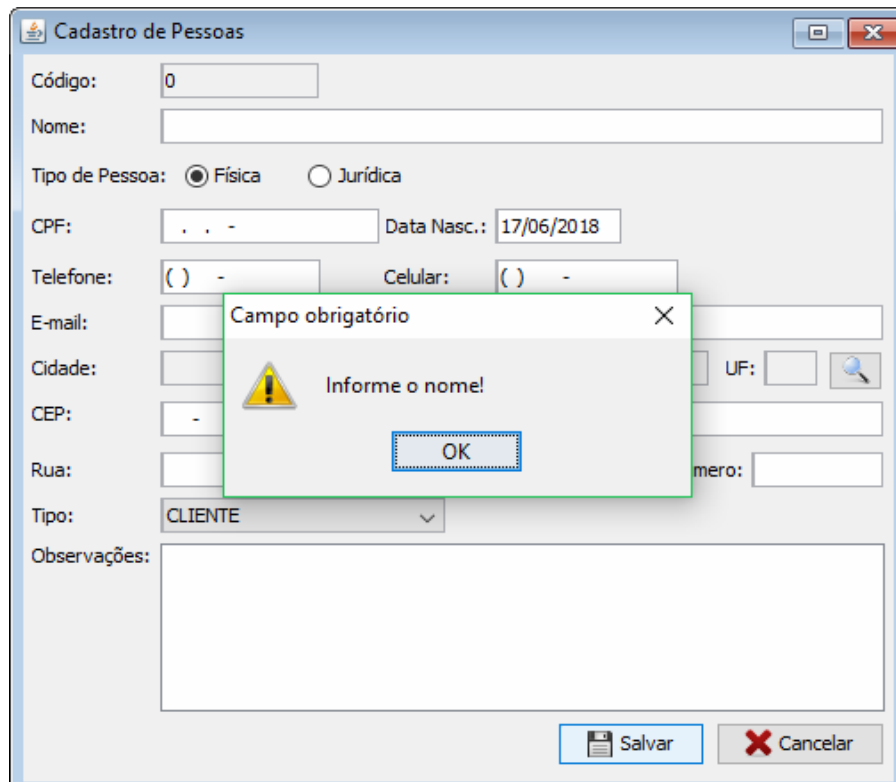
Ao clicar no botão “Incluir”, o sistema abrirá a tela de cadastro de pessoas. Essa tela possui os campos necessários para a inclusão de uma nova pessoa, sendo alguns de preenchimento obrigatório e, portanto, validados e outros opcionais, e há, ainda, os botões para salvar ou cancelar. Essa tela de cadastro é apresentada na Figura 11.

The image shows a software window titled "Cadastro de Pessoas". It contains the following fields and controls:

- Código:** Text box containing "0".
- Nome:** Empty text box.
- Tipo de Pessoa:** Radio buttons for "Física" (selected) and "Jurídica".
- CPF:** Text box with format ". . -".
- Data Nasc.:** Text box containing "17/06/2018".
- Telefone:** Text box with format "() -".
- Celular:** Text box with format "() -".
- E-mail:** Empty text box.
- Cidade:** Text box, followed by "UF:" and a dropdown menu with a search icon.
- CEP:** Text box containing "-".
- Bairro:** Empty text box.
- Rua:** Empty text box.
- Número:** Empty text box.
- Tipo:** Dropdown menu showing "CLIENTE".
- Observações:** Large empty text area.
- Buttons:** "Salvar" (Save) and "Cancelar" (Cancel) at the bottom right.

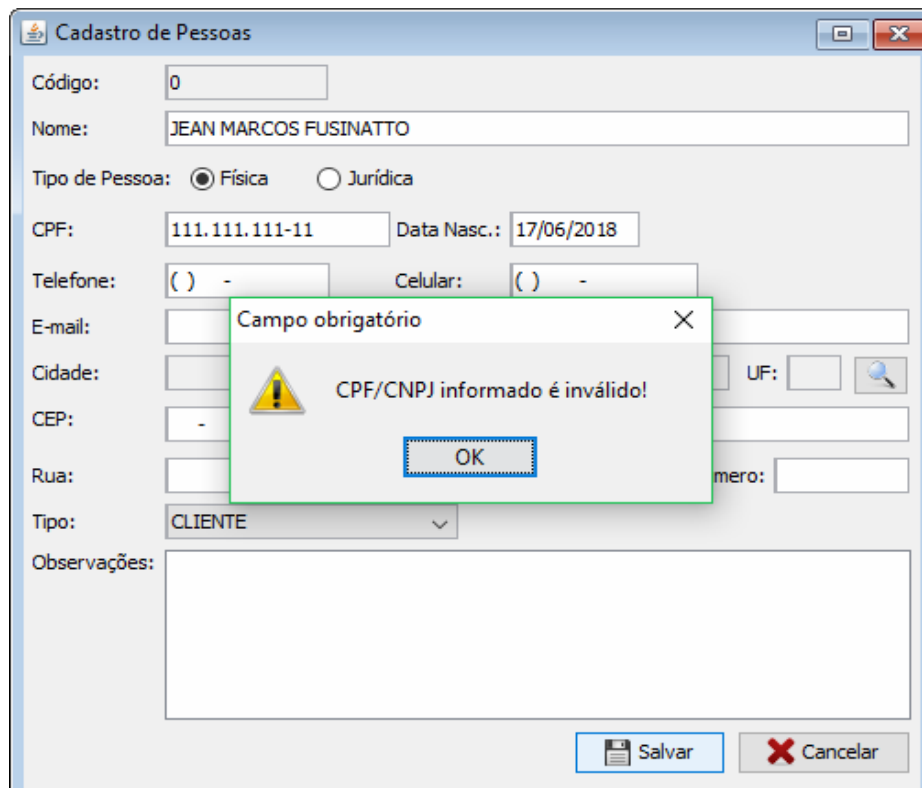
Figura 11 – Cadastro de pessoas

Quando o usuário clicar no botão “Salvar” será verificado se todos os campos obrigatórios foram informados e se todos os valores informados são válidos de acordo com a necessidade de cada campo. Caso o usuário não preencha algum campo obrigatório ou informe valor inválido para o tipo requerido, será apresentada uma mensagem solicitando o preenchimento correto do campo. Nas Figura 12 e 13, são apresentadas, respectivamente, as mensagens de preenchimento obrigatório e de preenchimento inválido.



The screenshot shows a web form titled "Cadastro de Pessoas". The form fields are: Código (0), Nome (empty), Tipo de Pessoa (Física selected, Jurídica unselected), CPF (empty), Data Nasc. (17/06/2018), Telefone (empty), Celular (empty), E-mail (empty), Cidade (empty), CEP (empty), Rua (empty), Tipo (CLIENTE), and Observações (empty). A modal dialog box titled "Campo obrigatório" is overlaid on the form, containing a warning icon and the text "Informe o nome!". The dialog has an "OK" button. At the bottom of the form are "Salvar" and "Cancelar" buttons.

Figura 12 – Mensagem de campo obrigatório



The screenshot shows the same "Cadastro de Pessoas" form, but now with the following values: Código (0), Nome (JEAN MARCOS FUSINATTO), Tipo de Pessoa (Física selected, Jurídica unselected), CPF (111.111.111-11), Data Nasc. (17/06/2018), Telefone (empty), Celular (empty), E-mail (empty), Cidade (empty), CEP (empty), Rua (empty), Tipo (CLIENTE), and Observações (empty). A modal dialog box titled "Campo obrigatório" is overlaid on the form, containing a warning icon and the text "CPF/CNPJ informado é inválido!". The dialog has an "OK" button. At the bottom of the form are "Salvar" and "Cancelar" buttons.

Figura 13 – Mensagem de campo inválido

Se o preenchimento dos campos for como o esperado, em termos de campos de preenchimento obrigatório e válidos, o sistema apresentará mensagem de sucesso na operação e gravará o novo registro, como apresentado na Figura 14. Após confirmação na mensagem a tela de pesquisa é apresentada novamente.

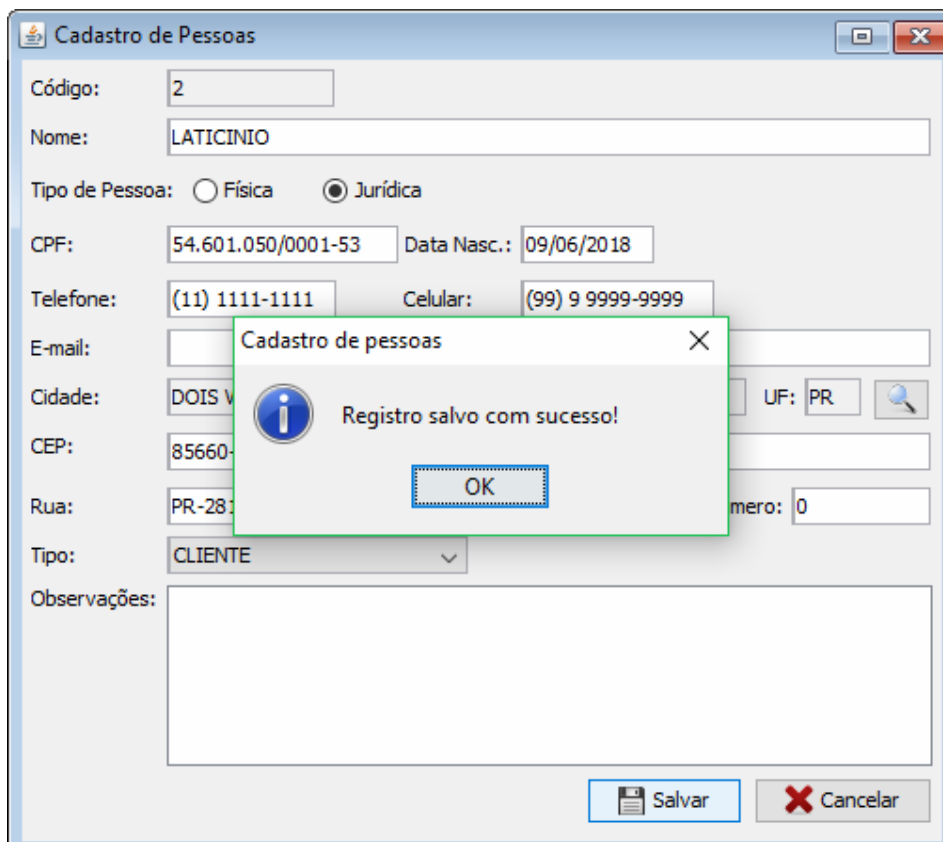


Figura 14 – Mensagem de sucesso na gravação

Para alterar um registro, o usuário deve selecionar, na tela de pesquisa, o registro que deseja alterar (clcando sobre ele) e, posteriormente, clicar no botão “Editar”. O processo de alteração é semelhante ao processo de inclusão, diferenciando pelo fato de que na alteração os campos são apresentados preenchidos com os valores armazenados no banco de dados para o respectivo registro.

Para realizar a exclusão de um registro, o usuário, da mesma forma que na alteração, deve selecioná-lo e, posteriormente, clicar no botão “Excluir”. O sistema solicitará a confirmação do usuário para excluir o registro, conforme apresenta a Figura 15.

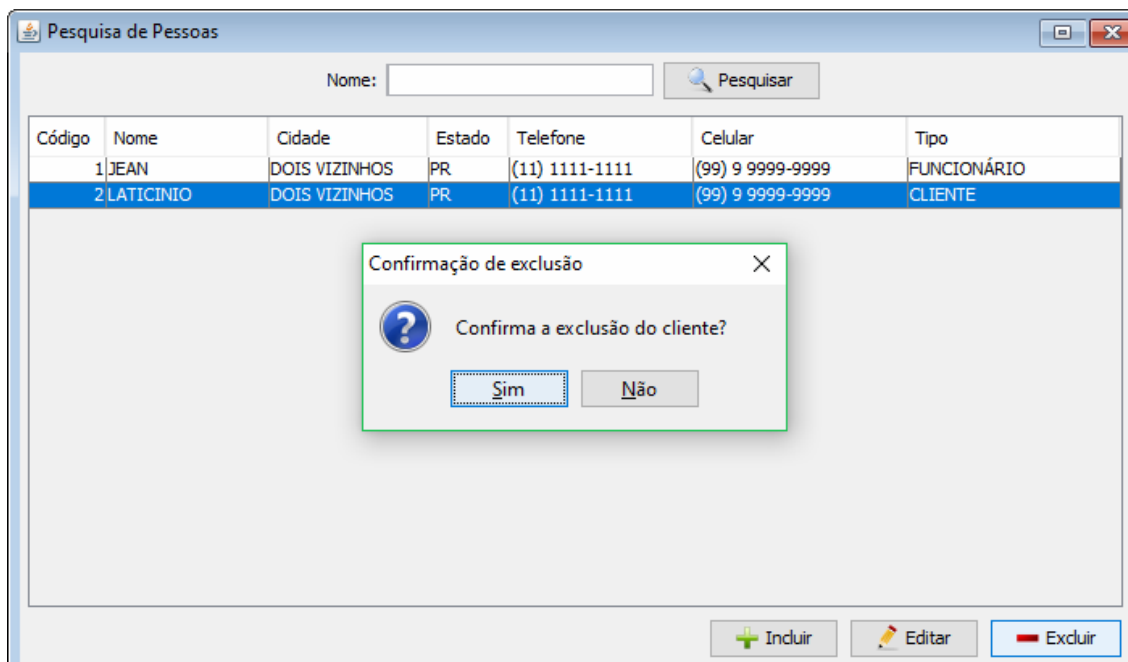


Figura 15 – Solicitação de confirmação para de exclusão

Uma das principais telas do sistema é a de registro de ordenha. Nela o usuário insere os dados das produções de cada animal de forma individual, ou caso deseje, da produção de um rebanho. Quando o usuário registra uma ordenha identificada com o tipo “Individual” deve ser selecionado o animal para o qual a ordenha se refere. Selecionando a opção de ordenha “Geral” é necessário selecionar o rebanho, dessa forma a quantidade informada será dividida igualmente, sendo calculada a média entre todos os animais que pertencerem ao rebanho selecionado. Outras informações importantes da ordenha também são inseridas nesta tela como a data e a hora de realização da ordenha, funcionário responsável e o tanque de destino do leite ordenhado. Na Figura16 é apresentada a interface da tela de registro de ordenha.

The screenshot shows a software window titled "Registro de Ordenha". It contains the following fields and controls:

- Tipo de Ordenha: dropdown menu with "Individual" selected.
- Data: text box with "17/06/2018".
- Hora: text box with "22:47".
- Funcionário: empty text box with a search icon.
- Tanque: dropdown menu.
- Quantidade: text box with "0.0".
- Animal: empty text box with a search icon.
- Rebanho: dropdown menu with "PRODUÇÃO" selected.
- Buttons: "Salvar" (Save) and "Cancelar" (Cancel).

Figura 16 – Registro de ordenha

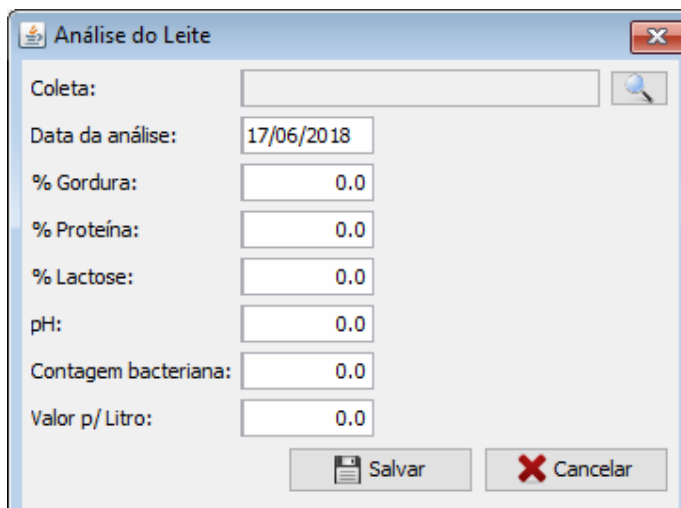
Outra tela importante do sistema é a de registro de coleta. A partir desta tela é realizada a baixa do estoque de leite do tanque. Nessa tela são registradas as seguintes informações sobre a coleta: data e hora, laticínio, funcionário responsável, tanque e quantidade coletada. Na Figura 17 está a interface da tela de registro de coleta do leite.

The screenshot shows a software window titled "Registro de Coleta do Leite". It contains the following fields and controls:

- Data: text box with "17/06/2018".
- Hora: text box with "23:02".
- Laticínio: empty text box with a search icon.
- Funcionário Responsável: empty text box with a search icon.
- Tanque: dropdown menu.
- Quantidade: text box with "0.0".
- Buttons: "Salvar" (Save) and "Cancelar" (Cancel).

Figura 17 – Registro de coleta do leite

A partir das coletas realizadas, o laticínio realiza a análise do leite. A análise determina a qualidade e, conseqüentemente, o valor pago pelo litro do leite. No sistema há uma tela para inserção das informações referentes à análise enviada pelo laticínio. As informações podem ser consultadas e comparadas posteriormente em relatórios para a tomada de decisão pelo produtor. A Figura 18 apresenta a interface utilizada para o registro das informações da análise realizada em amostra de leite.



Análise do Leite

Coleta:

Data da análise: 17/06/2018

% Gordura: 0.0

% Proteína: 0.0

% Lactose: 0.0

pH: 0.0

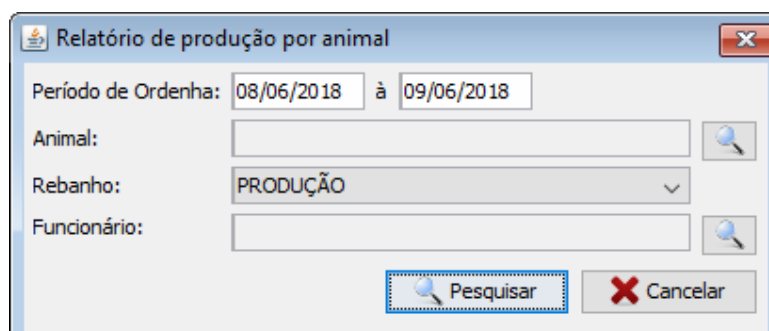
Contagem bacteriana: 0.0

Valor p/ Litro: 0.0

Salvar Cancelar

Figura 18 – Registro de análise do leite

Além das operações de cadastro o sistema possui funcionalidades que auxiliam na tomada de decisão como relatórios de produção, histórico da qualidade da produção, receitas e despesas. Nas Figuras 19 e 20 são apresentadas, respectivamente, a tela de filtros e o relatório de produção.



Relatório de produção por animal

Período de Ordenha: 08/06/2018 à 09/06/2018

Animal:

Rebanho: PRODUÇÃO

Funcionário:

Pesquisar Cancelar

Figura 19 – Filtros para geração do relatório de produção

No relatório apresentado na Figura 20 é apresentada a produção por animal de forma detalhada e a produção total dos animais selecionados. Os filtros para geração do relatório são: período, animal, rebanho e funcionário (como apresentado na Figura 19), sendo período o único filtro obrigatório.

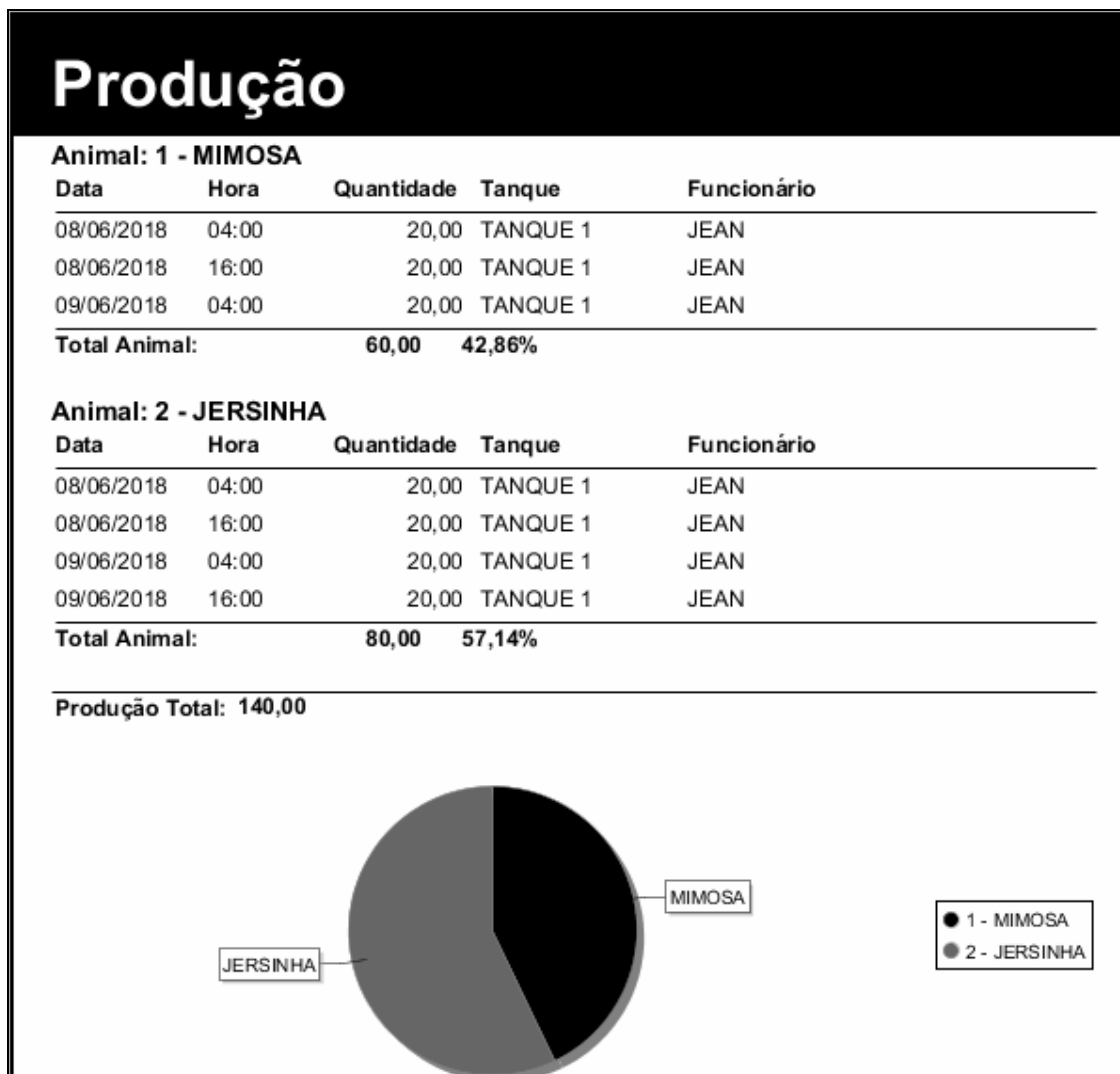
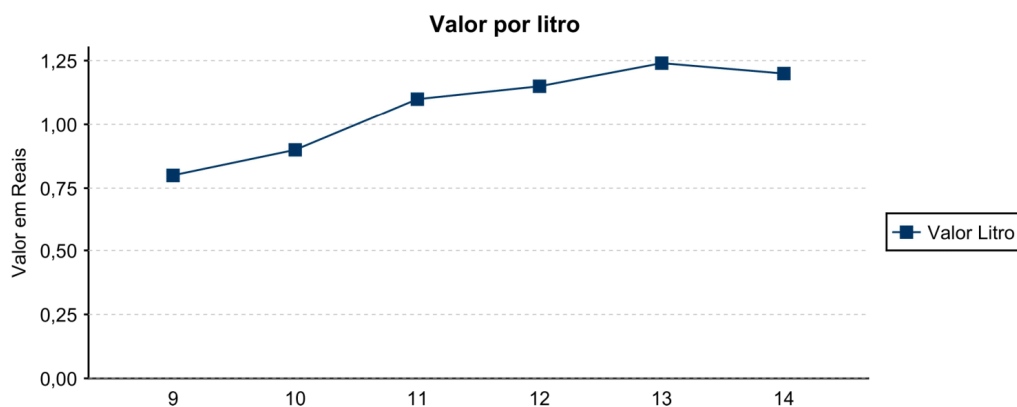
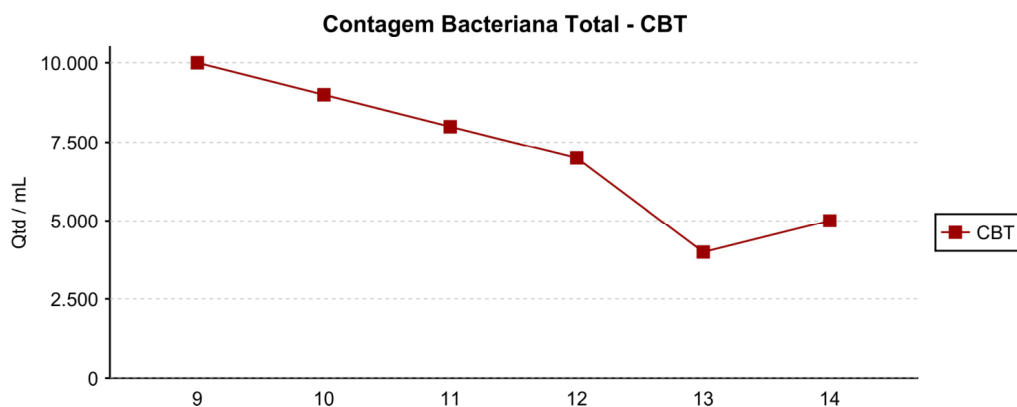
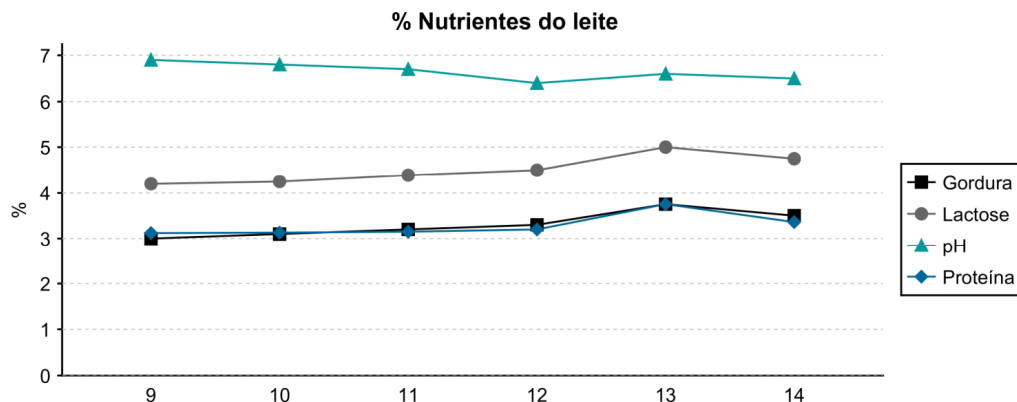


Figura 20 – Relatório de produção de leite

Outro relatório disponível é o de qualidade do leite. Por meio desse relatório, pelo histórico das análises enviadas pelo laticínio, o produtor pode acompanhar a evolução da qualidade do leite produzido pelo seu rebanho. A tela de filtros para a geração deste relatório contém apenas o filtro de período da análise. Na Figura 21 é apresentado um relatório de qualidade do leite.

Qualidade do Leite



	% Gordura	% Proteína	% Lactose	pH	CTB	Valor do litro
Mín.:	3.0	3.12	4.2	6.4	4000.0	0.8
Máx.:	3.75	3.75	5.0	6.9	10000.0	1.24

Figura 21 - Relatório de qualidade do leite

4.4 IMPLEMENTAÇÃO DO SISTEMA

O projeto do sistema foi dividido em três camadas de acordo com o padrão *Model-View-Controller* (MVC). Na camada *model* foram criadas todas as classes referentes às entidades/tabelas utilizadas no sistema assim como as classes *Data Access Object* (DAO) responsáveis pelas operações *Create, Read, Update e Delete* (CRUD) e pelas transações e conexões necessárias. Na camada de *view* ficam as classes da interface gráfica do sistema que foram desenvolvidas utilizando os componentes Swing do Java, e são utilizadas para que o usuário possa interagir com o sistema. A camada *controller* contém as classes responsáveis por integrar as camadas *model* e *view*.

Para realizar a conexão via *Java Database Connectivity* (JDBC) com o banco de dados PostgreSQL foi adicionado arquivo .jar referente ao PostgreSQL e configurado o arquivo *persistence.xml*, conforme apresentado na Listagem 1. Esta configuração indica o método de criação das tabelas e colunas e também informações de conexão como porta, nome da database, usuário e senha.

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.0" xmlns="http://java.sun.com/xml/ns/persistence"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd">
  <persistence-unit name="MilkPlusViewPU" transaction-type="RESOURCE_LOCAL">
    <provider>org.hibernate.ejb.HibernatePersistence</provider>
    <properties>
      <property name="javax.persistence.jdbc.url"
value="jdbc:postgresql://localhost:5433/ProjetoTCC"/>
      <property name="javax.persistence.jdbc.user" value="postgres"/>
      <property name="javax.persistence.jdbc.driver"
value="org.postgresql.Driver"/>
      <property name="javax.persistence.jdbc.password" value="pgsql"/>
      <property name="hibernate.cache.provider_class"
value="org.hibernate.cache.NoCacheProvider"/>
      <property name="hibernate.show_sql" value="true"/>
      <property name="hibernate.format_sql" value="true"/>
    </properties>
  </persistence-unit>
</persistence>
```

Listagem 1 – persistence.xml

Foi criada uma classe de gerenciamento de conexão para garantir que cada acesso ao sistema utilize uma única conexão. Na Listagem 2 é apresentada a classe de gerenciamento de conexão ao banco de dados.

```

package br.edu.utfpr.projetoTCC.modelo.dao;

import java.sql.Connection;
import java.sql.DriverManager;
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;

/**
 *
 * @author Jean
 */
public class DBConnect {

    private static final String DRIVER = "org.postgresql.Driver";
    private static final String URL = "jdbc:postgresql://localhost:5433/";
    private static final String DBNAME = "ProjetoTCC";
    private static final String USERNAME = "postgres";
    private static final String PASSWORD = "pgsql";

    private static DBConnect db;
    private Connection connection;
    private EntityManagerFactory factory;

    private DBConnect(){
        try {
            Class.forName(DRIVER).newInstance();
            this.connection = DriverManager.getConnection(URL + DBNAME, USERNAME,
PASSWORD);

            this.factory = Persistence.createEntityManagerFactory(
"MilkPlusViewPU" );
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static synchronized DBConnect getInstance(){
        if(db == null)
            db = new DBConnect();

        return db;
    }

    public Connection getConnection(){
        return this.connection;
    }

    public EntityManager getEntityManager() {
        return this.factory.createEntityManager();
    }
}

```

Listagem 2 – DBConnect.java

Todas as classes referentes às entidades do banco de dados foram mapeadas para a persistência utilizando *Java Persistence API (JPA)* por meio de *annotations* e o *framework* Hibernate. Na Listagem 3, a classe `Animal.java` é apresentada com suas *annotations* para mapeamento da tabela e colunas e também seus métodos construtor, *getters* e *setters*.

```

package br.edu.utfpr.projetoctcc.model.vo;

import java.util.Date;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;
import javax.persistence.Temporal;

/**
 *
 * @author Jean
 */
@Entity
@Table(name = "Animal")
public class Animal {
    @Id
    @SequenceGenerator(name = "SEQ_ANIMAL", sequenceName = "SEQ_ANIMAL_ID",
allocationSize = 1)
    @GeneratedValue(strategy = GenerationType.IDENTITY, generator = "SEQ_ANIMAL")
    private int animalId;

    @Column( length = 10, nullable = false )
    private String brinco;

    @Column( length = 100, nullable = false )
    private String apelido;

    @Column( nullable = false )
    private String tatuagem; // Imagem

    @ManyToOne
    @JoinColumn( name = "rebanhoId", referencedColumnName = "rebanhoId", nullable
= false )
    private Rebanho rebanho;

    @Temporal( javax.persistence.TemporalType.DATE )
    @Column( nullable = false )
    private Date dataNascimento;

    @Temporal( javax.persistence.TemporalType.DATE )
    @Column( nullable = false )
    private Date dataEntrada;

    @ManyToOne
    @JoinColumn( name = "pelagemId", referencedColumnName = "pelagemId", nullable
= false )

```

```
private Pelagem pelagem;

@Column( length = 100, nullable = false )
private String cor;

public Animal() {
    animalId = 0;
    brinco = "";
    apelido = "";
    tatuagem = "";
    rebanho = new Rebanho();
    dataNascimento = new Date();
    dataEntrada = new Date();
    pelagem = new Pelagem();
    cor = "";
}

public int getAnimalId() {
    return animalId;
}

public void setAnimalId(int animalId) {
    this.animalId = animalId;
}

public String getBrinco() {
    return brinco;
}

public void setBrinco(String brinco) {
    this.brinco = brinco;
}

public String getApelido() {
    return apelido;
}

public void setApelido(String apelido) {
    this.apelido = apelido;
}

public String getTatuagem() {
    return tatuagem;
}

public void setTatuagem(String tatuagem) {
    this.tatuagem = tatuagem;
}

public Rebanho getRebanho() {
    return rebanho;
}

public void setRebanho(Rebanho rebanho) {
    this.rebanho = rebanho;
}

public Date getDataNascimento() {
    return dataNascimento;
}
```

```

    }

    public void setDataNascimento(Date dataNascimento) {
        this.dataNascimento = dataNascimento;
    }

    public Date getDataEntrada() {
        return dataEntrada;
    }

    public void setDataEntrada(Date dataEntrada) {
        this.dataEntrada = dataEntrada;
    }

    public Pelagem getPelagem() {
        return pelagem;
    }

    public void setPelagem(Pelagem pelagem) {
        this.pelagem = pelagem;
    }

    public String getCor() {
        return cor;
    }

    public void setCor(String cor) {
        this.cor = cor;
    }
}

```

Listagem 3 – Animal.java

Para facilitar e padronizar a criação das classes, foi criada uma interface GenericDAO, na qual todas as classes DAO a implementam. Na Listagem 4 é apresentado o código da interface GenericDAO.

```

package br.edu.utfpr.projetoctcc.modelo.dao;

import java.sql.SQLException;
import java.util.List;

/**
 *
 * @author Jean
 */
public interface GenericDAO<T> {
    public int inserir(T objeto) throws SQLException;
    public void alterar(T objeto) throws SQLException;
    public void excluir(T objeto) throws SQLException;
    public List<T> listar(String filtro) throws SQLException;
    public T buscar(int codigo) throws SQLException;
}

```

Listagem 4 – GenericDAO.java

Na Listagem 5 é apresentado o código de criação da classe AnimalDAO que implementa a interface GenericDAO.

```

package br.edu.utfpr.projetotcc.model.dao;

import br.edu.utfpr.projetotcc.model.vo.Animal;
import java.sql.SQLException;
import java.util.List;
import javax.persistence.EntityManager;
import javax.persistence.Query;

/**
 *
 * @author Jean
 */
public class AnimalDAO implements GenericDAO<Animal>{

    @Override
    public int inserir(Animal objeto) throws SQLException {
        EntityManager manager = DBConnect.getInstance().getEntityManager();

        manager.getTransaction().begin();
        manager.persist( objeto );
        manager.getTransaction().commit();

        return objeto.getAnimalId();
    }

    @Override
    public void alterar(Animal objeto) throws SQLException {
        EntityManager manager = DBConnect.getInstance().getEntityManager();

        manager.getTransaction().begin();
        manager.merge( objeto );
        manager.getTransaction().commit();
    }

    @Override
    public void excluir(Animal objeto) throws SQLException {
        EntityManager manager = DBConnect.getInstance().getEntityManager();

        manager.getTransaction().begin();
        manager.remove( manager.getReference( Animal.class, objeto.getAnimalId() )
);
        manager.getTransaction().commit();
    }

    @Override
    public List<Animal> listar(String filtro) throws SQLException {
        EntityManager manager = DBConnect.getInstance().getEntityManager();

        Query query = manager.createQuery( "SELECT a FROM Animal AS a WHERE
a.apelido LIKE :filtro OR a.brinco LIKE :filtro" );
        query.setParameter( "filtro", "%" + filtro + "%" );

        return query.getResultList();
    }

    @Override

```

```

public Animal buscar(int codigo) throws SQLException {
    EntityManager manager = DBConnect.getInstance().getEntityManager();

    return manager.find( Animal.class, codigo );
}
}

```

Listagem 5 – AnimalDAO.java

A fim de padronizar as classes dos *controllers*, da mesma forma que com os DAO, foi criado uma interface *GenericController* conforme apresenta o código da Listagem 6.

```

package br.edu.utfpr.projetotcc.controller;

/**
 *
 * @author Jean
 */
public interface GenericController {
    public void pesquisar();
    public void abrirPesquisa();
    public void inserir();
    public void atribuirCampos();
    public void popularObjeto();
    public void salvar();
    public void excluir();
    public void editar();
    public int getIdSelecionado();
    public void createActionListenersToJanelaPesquisa();
    public void createActionListenersToJanelaCadastro();
    public boolean validar();
}

```

Listagem 6 – GenericController.java

Todos os *controllers* foram criados implementando a classe genérica tratando suas particularidades conforme Listagem 7. A classe *AnimalController.java* consiste, basicamente, em gerenciar todos os componentes das classes de cadastro e pesquisa de animais. Nela são recuperados os componentes por meio dos métodos *get* e é realizado o tratamento necessário para pesquisa, gravação ou alteração da entidade correspondente.

```

package br.edu.utfpr.projetotcc.controller;

import br.edu.utfpr.projetotcc.controller.util.UtillsController;
import br.edu.utfpr.projetotcc.model.dao.AnimalDAO;
import br.edu.utfpr.projetotcc.model.dao.PelagemDAO;
import br.edu.utfpr.projetotcc.model.dao.RebanhoDAO;
import br.edu.utfpr.projetotcc.model.vo.Animal;
import br.edu.utfpr.projetotcc.model.vo.Pelagem;
import br.edu.utfpr.projetotcc.model.vo.Rebanho;
import br.edu.utfpr.projetotcc.view.FrmAnimalCadastro;
import br.edu.utfpr.projetotcc.view.FrmAnimalPesquisa;
import java.awt.event.ActionEvent;

```

```

import java.awt.event.ActionListener;
import java.sql.SQLException;
import java.util.List;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author Jean
 */
public class AnimalController implements GenericController {

    private FrmAnimalPesquisa janelaPesquisa;
    private FrmAnimalCadastro janelaCadastro;
    private Animal animal;
    private UtilsController util;

    public AnimalController(FrmAnimalCadastro frmAnimalCadastro, FrmAnimalPesquisa
frmAnimalPesquisa) {
        this.janelaPesquisa = frmAnimalPesquisa;
        this.janelaCadastro = frmAnimalCadastro;
        this.util = new UtilsController();

        createActionListenersToJanelaPesquisa();
        createActionListenersToJanelaCadastro();
    }

    @Override
    public void pesquisar() {
        try {
            AnimalDAO dao = new AnimalDAO();
            List<Animal> animais =
dao.listar(this.janelaPesquisa.getTfFiltro().getText());
            DefaultTableModel modelo = (DefaultTableModel)
this.janelaPesquisa.getTbTabela().getModel();

            while (modelo.getRowCount() > 0) {
                modelo.removeRow(0);
            }

            for (Animal p : animais) {
                modelo.addRow(new Object[]{ p.getAnimalId(),
                    p.getBrinco(),
                    p.getApelido(),
                    util.dateToString(p.getDataNascimento()),
                    util.dateToString(p.getDataEntrada()),
                    p.getRebanho().getNome(),
                    p.getPelagem().getDescricao(),
                    p.getCor()
                });
            }
        } catch (Exception ex) {
            JOptionPane.showMessageDialog(janelaPesquisa, "Ocorreu um erro ao
carregar as informações",
                "Pesquisa de Animais", JOptionPane.WARNING_MESSAGE);
            ex.printStackTrace();
        }
    }
}

```



```

@Override
public void abrirPesquisa() {
    this.pesquisar();
    this.janelaPesquisa.setVisible(true);
}

@Override
public void inserir() {
    this.animal = new Animal();
    this.atribuirCampos();
    this.janelaCadastro.setVisible(true);
    pesquisar();
}

@Override
public void atribuirCampos() {

this.janelaCadastro.getTfCodigo().setText(Integer.toString(this.animal.getId()));
    this.janelaCadastro.getTfApelido().setText(this.animal.getApelido());
    this.janelaCadastro.getTfBrinco().setText(this.animal.getBrinco());
    this.janelaCadastro.getTfCor().setText(this.animal.getCor());

this.janelaCadastro.getTfDataEntrada().setText(
    util.dateToString(this.animal.getDataEntrada()));

this.janelaCadastro.getTfDataNascimento().setText(
    util.dateToString(this.animal.getDataNascimento()));
    this.janelaCadastro.getCbRebanho().setModel(
    new DefaultComboBoxModel(getRebanho().toArray()));

this.janelaCadastro.getCbRebanho().getModel().setSelectedItem(
    this.animal.getRebanho());
    this.janelaCadastro.getCbPelagem().setModel(
    new DefaultComboBoxModel(getPelagem().toArray()));

this.janelaCadastro.getCbPelagem().getModel().setSelectedItem(
    this.animal.getPelagem());
    this.janelaCadastro.getTfTatuagem().setText(
    this.animal.getTatuagem());
}

@Override
public void popularObjeto() {
    this.animal.setApelido(this.janelaCadastro.getTfApelido().getText());
    this.animal.setBrinco(this.janelaCadastro.getTfBrinco().getText());
    this.animal.setCor(this.janelaCadastro.getTfCor().getText());
    this.animal.setDataEntrada(
    util.stringToDate(this.janelaCadastro.getTfDataEntrada().getText()));
    this.animal.setDataNascimento(
    util.stringToDate(this.janelaCadastro.getTfDataNascimento().getText()));
    this.animal.setPelagem(
    this.janelaCadastro.getCbPelagem().getSelectedItem());
    this.animal.setRebanho(
    this.janelaCadastro.getCbRebanho().getSelectedItem());
    this.animal.setTatuagem(
    this.janelaCadastro.getTfTatuagem().getText());
}

@Override

```

```

public void salvar() {
    try {
        popularObjeto();
        AnimalDAO dao = new AnimalDAO();

        if (this.animal.getAnimalId() == 0) {
            dao.inserir(this.animal);
        } else {
            dao.alterar(this.animal);
        }
        pesquisar();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this.janelaCadastro, "Nao foi possivel salvar o registro");
        e.printStackTrace();
    }
}

@Override
public void excluir() {
    try {
        AnimalDAO dao = new AnimalDAO();
        this.animal = dao.buscar( this.getIdSelecioneado() );

        if (JOptionPane.showConfirmDialog(this.janelaPesquisa, "Confirma a exclusão do animal?") == JOptionPane.YES_OPTION) {
            dao.excluir( this.animal );
            this.pesquisar();
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this.janelaPesquisa, "Não foi possível excluir o registro");
        e.printStackTrace();
    }
}

@Override
public void editar() {
    try {
        AnimalDAO dao = new AnimalDAO();
        this.animal = dao.buscar( this.getIdSelecioneado() );
        this.atribuirCampos();
        this.janelaCadastro.setVisible( true );
        pesquisar();
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(this.janelaPesquisa, "Não foi possível carregar o registro");
        ex.printStackTrace();
    }
}

@Override
public int getIdSelecioneado() {
    int row = this.janelaPesquisa.getTbTabela().getSelectedRow();
    if (row == -1) {
        return 0;
    } else {
        return (int) this.janelaPesquisa.getTbTabela().getValueAt(row, 0);
    }
}
}

```

```

@Override
public void createActionListenersToJanelaPesquisa() {
    this.janelaPesquisa.getBtPesquisar().addActionListener(new
ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            pesquisar();
        }
    });

    this.janelaPesquisa.getBtIncluir().addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            inserir();
        }
    });

    this.janelaPesquisa.getBtEditar().addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            editar();
        }
    });

    this.janelaPesquisa.getBtExcluir().addActionListener(new ActionListener()
{
    @Override
    public void actionPerformed(ActionEvent e) {
        excluir();
    }
});
}

@Override
public void createActionListenersToJanelaCadastro() {
    this.janelaCadastro.getBtSalvar().addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            if( validar() ){
                salvar();
                janelaCadastro.setVisible(false);
            }
        }
    });

    this.janelaCadastro.getBtCancelar().addActionListener(new ActionListener()
{
    @Override
    public void actionPerformed(ActionEvent e) {
        janelaCadastro.setVisible(false);
    }
});
}

@Override
public boolean validar() {
    if( this.janelaCadastro.getTfApelido().getText().trim().equals("") ){

```

```

        JOptionPane.showMessageDialog(this.janelaCadastro, "Informe o apelido
do animal!", "Campo obrigatório", JOptionPane.WARNING_MESSAGE);
        this.janelaCadastro.getTfApelido().requestFocus();
        return false;
    }

    if( this.janelaCadastro.getTfBrinco().getText().trim().equals( "" ) ){
        JOptionPane.showMessageDialog(this.janelaCadastro, "Informe o brinco
do animal!", "Campo obrigatório", JOptionPane.WARNING_MESSAGE);
        this.janelaCadastro.getTfBrinco().requestFocus();
        return false;
    }

    if( !util.validarData(this.janelaCadastro.getTfDataEntrada().getText(),
false) ){
        JOptionPane.showMessageDialog(this.janelaCadastro, "Data de entrada
inválida!", "Campo obrigatório", JOptionPane.WARNING_MESSAGE);
        this.janelaCadastro.getTfDataEntrada().requestFocus();
        return false;
    }

    if( !util.validarData(this.janelaCadastro.getTfDataNascimento().getText(),
false) ){
        JOptionPane.showMessageDialog(this.janelaCadastro, "Data de nascimento
inválida!", "Campo obrigatório", JOptionPane.WARNING_MESSAGE);
        this.janelaCadastro.getTfDataNascimento().requestFocus();
        return false;
    }

    if( this.janelaCadastro.getCbRebanho().getSelectedItem() == null ){
        JOptionPane.showMessageDialog(this.janelaCadastro, "Selecione o
rebanho!", "Campo obrigatório", JOptionPane.WARNING_MESSAGE);
        this.janelaCadastro.getCbRebanho().requestFocus();
        return false;
    }

    if( this.janelaCadastro.getCbPelagem().getSelectedItem() == null ){
        JOptionPane.showMessageDialog(this.janelaCadastro, "Selecione a
pelagem!", "Campo obrigatório", JOptionPane.WARNING_MESSAGE);
        this.janelaCadastro.getCbPelagem().requestFocus();
        return false;
    }

    return true;
}

private List<Rebanho> getRebanho(){
    try {
        RebanhoDAO rebanhoDao = new RebanhoDAO();
        return rebanhoDao.listar( "" );
    } catch (SQLException ex) {
        ex.printStackTrace();
        return null;
    }
}

private List<Pelagem> getPelagem(){
    try {
        PelagemDAO pelagemDao = new PelagemDAO();

```

```

        return pelagemDao.listar( "" );
    } catch (SQLException ex) {
        ex.printStackTrace();
        return null;
    }
}

```

Listagem 7 – AnimalController.java

Para a geração e a visualização dos relatórios foi utilizada a classe JasperPrint.java e JasperView.java. A classe JasperPrint.java é utilizada para leitura do arquivo .jasper referente ao relatório e a classe JasperView.java é responsável pela apresentação. A Listagem 8 apresenta a utilização das classes no método pesquisar no *controller* dos relatórios de animais.

```

public void pesquisar() {

    Date dataInicial =
    util.stringToDate(this.janelaPesquisa.getTfDataInicial().getText());
    Date dataFinal =
    util.stringToDate(this.janelaPesquisa.getTfDataFinal().getText());

    EntityManager manager = DBConnect.getInstance().getEntityManager();
    String src = "";
    if( this.tipoRelatorio == 1 ){
        src = "RelatorioProducao.jasper";
    } else {
        src = "RelatorioVacinas.jasper";
    }

    HashMap hm = new HashMap();
    hm.put("dataInicial", dataInicial);
    hm.put("dataFinal", dataFinal);
    hm.put("animalId", this.animalId);
    hm.put("rebanhoId", this.rebanhoId);
    hm.put("funcionarioId", this.funcionarioId);

    try {
        JasperPrint jasperPrint = JasperFillManager.fillReport(src, hm,
        DBConnect.getInstance().getConnection());
        JasperViewer view = new JasperViewer(jasperPrint, false);
        view.setVisible(true);
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(janelaPesquisa, "Ocorreu um erro ao
        carregar as informações",
        "Relatório de animais", JOptionPane.WARNING_MESSAGE);
        ex.printStackTrace();
    }
}

```

Listagem 8 – RelatorioAnimalController.java

5 CONSIDERAÇÕES FINAIS

O objetivo deste trabalho foi realizar a modelagem e a implementação de um software para facilitar o gerenciamento das atividades relacionadas ao gado leiteiro. Esse gerenciamento envolve a produção leiteira em si (com o controle de ordenhas, produção de cada animal ou média de produção por ordenha, registro de análises realizadas no leite); o controle de receitas (venda de leite, animas e outros) e das despesas (compra e tratamento de animais, funcionários, custos operacionais e outras); e o acompanhamento dos animais (intervenções medicamentosas, atendimentos veterinários, entre outros).

A modelagem foi realizada gerando um banco de dados relativamente grande para um trabalho acadêmico. São diversos cadastros que são necessários para atender aos requisitos definidos para o sistema. A grande quantidade de dados que serão armazenados se justifica pela diversidade de tarefas realizadas e dos controles necessários nesse tipo de negócio. Esses dados são utilizados para a produção de relatórios e gráficos que permitam um acompanhamento mais efetivo da produção.

O sistema foi implementado para ambiente *desktop* pela dificuldade de disponibilidade de Internet nas propriedades rurais de pequeno porte. O desenvolvimento para *desktop* utilizando a linguagem Java trouxe alguns desafios de programação que foram superados com estudo em referenciais teóricos e auxílio em fóruns na Internet. A falta de prática no uso da referida linguagem contribuiu para essas dificuldades, mas o conhecimento de outra linguagem de programação para *desktop* auxiliou no entendimento dos processos e da lógica de implementação. Assim, as dificuldades maiores estiveram centradas na sintaxe da linguagem e na busca dos recursos da linguagem Java que pudessem ser utilizados para implementar a lógica e as funcionalidades definidas e pretendidas para o sistema.

O sistema será implementado para uso em fase de teste e, certamente, haverá a necessidade de ajustes em funcionalidades e requisitos. A possibilidade de implementação de novos relatórios, em decorrência da quantidade e diversidade de dados armazenados, é uma possibilidade bastante evidente a partir do uso do sistema.

REFERÊNCIAS

AGRICIÊNCIA. **AJAP – Associação de Jovens Agricultores de Portugal**. Programa AGRO –Medida 7. Zootecnia de Precisão. Disponível em:

<http://agriciencia.servehttp.com/agrinov_ajap/default.asp>. Acesso em: 28 out. 2014

AGROSOFT. O impacto da internet no agronegócio. **Revista Agrosoft**, Juiz de Fora, n. 9, p. 3-28, 2000.

ARRAES, Nilson A. M. **Levantamento das aplicações das tecnologias da informação no meio rural com estudo de caso sobre a oferta de software agrícola no Estado de São Paulo**. 1993. (Dissertação de Mestrado) - Faculdade de Engenharia Elétrica - Universidade Estadual de Campinas, Campinas.

ARVUS TECNOLOGIA. **Agricultura de precisão: ciclo da agricultura de precisão**. Disponível em: <http://www.arvus.com.br/publicacoes_exibe.html?id=1>. Acesso em: 12 out. 2014.

BAMBINI, Martha D. **TICs e setor agrícola: aplicações atuais e tendências**. Brasília, DF: Embrapa: SPI, 2010.

CARRASCAL, María J.; LOUIS, Pau F.; LUDWIG, Reiner Knowledge and information transfer in agriculture using hypermedia: a system review. **Computers and Electronics in Agriculture**, v. 12, p. 83-119, 1985.

CASTRO NETO, Miguel de; PINTO, Pedro Aguiar; COELHO, José Paulo Pimentel. **Tecnologias de informação e comunicação e a agricultura**. Porto: Sociedade Portuguesa de Inovação, 2005.

EMBRAPA INFORMÁTICA AGROPECUÁRIA. 2010. **SW AGRO Portal da tecnologia da informação para o agronegócio**. Disponível em: <<http://www.swagro.cnpia.embrapa.br/projeto/swagro/>>. Acesso em: 21 out. 2014.

EMPRESA BRASILEIRA DE PESQUISA AGROPECUÁRIA. **Produção, industrialização e comercialização (produção)**. Disponível em: <<http://www.cnpq.embrapa.br/nova/informacoes/estatisticas/producao/producao.php>>. Acesso em: 27 set. 2014.

FARMSOFT. **International agricultural software catalogue**. Belgique: Faculté des Sciences Agronomiques de Gembloux - Wallonne, 1994.

FURLANETO, Fernanda P. B.; MANZANO, Leandro M. **Agricultura de precisão e a rastreabilidade de produtos agrícolas**. Disponível em: <http://www.infobios.com.br/Artigos/2010_2/AgriculturaPrecisao/Index.htm>. Acesso em: 15 jul. 2010.

GEERS, R.; PUERS, B.; GOEDSELLS, V.; WOUTERS, P. Electronic identification and tracking in animals. **Cab International**, Oxon, v. 17, n. 2, p. 205-15, 1997.

GISLEITE. **Sistema de gestão informatizada de sistemas de produção de leite**. Disponível em: <<http://gisleite.cnpq.embrapa.br/>>. Acesso em: 27 set. 2014.

JESUS, José Carlos S.; ZAMBALDE, Andre L. **Administração Rural**. Lavras, MG: UFLA-FAEPE, 1988. (Curso de Pos-graduação Lato Sensu Informática na Agropecuária).

LOPES, Marco A. **Zootecnia de precisão**. Lavras, MG: Ufla, Departamento de Medicina Veterinária, 2010. (Notas de aula).

MAIA, Guilherme Baptista da Silva; PINTO, Arthur de Rezende; MARQUES, Cristiane Yaika Takaoka; ROITMAN, Fábio Brener; LYRA, Danielle Didier. Produção leiteira no Brasil. **Agropecuária**. BNDES Setorial 37, p. 371-398. 2013. Disponível em: <http://www.bndes.gov.br/SiteBNDES/export/sites/default/bndes_pt/Galerias/Arquivos/conhecimento/bnset/set3709.pdf>. Acesso em: 27 set. 2014.

MENDES, Cássia Isabel Costa; OLIVEIRA, Deise Rocha Martins dos Santos; Santos Anderson Rodrigo dos (editores técnicos). **Estudo do mercado brasileiro de software para o agronegócio**. Embrapa Informática Agropecuária. Campinas, SP. 2011

MILKPOINT. **Produção de leite no Brasil deve ser de 37 bilhões de litros em 2014**. 2013. Disponível em: <<http://www.milkpoint.com.br/cadeia-do-leite/giro-lacteo/producao-de-leite-no-brasil-deve-ser-de-37-bilhoes-de-litros-em-2014-86951n.aspx>>. Acesso em: 27 set. 2014.

MURAKAMI, Edson; SARAIVA, Antonio M. Rastreabilidade da informação nas cadeias produtivas: padrões de troca de dados. **Revista Brasileira de Agroinformática**, v. 7, n. 1, p. 58-66, 2005.

OFFICE OF TECHNOLOGY ASSESSMENT **Technology, public policy, and the changing structure of American Agriculture**. Washington, DC: U.S. Government Printing Office, 1992.

PAGLIS, Carlos M. **Tecnologia da informação na agricultura**. Lavras, MG: UFLA, Departamento de Agricultura, 2010. (Notas de aula).

SILVA, Alexandre P.; ALVES, Jean W.; BRAGA, Regina; CAMPOS, Fernanda. SBS-Agro: sistema de busca utilizando ontologias e retorno do usuário. **InfoUYclei 2002**. In: CONGRESSO URUGUAIO DE INFORMÁTICA. Centro Latinoamericano de Estudos em Informática. Montivideu: CLEI, 2002.

STEFANINI. **Big data, cloud, social e mobilidade são os pilares que têm determinado os investimentos em TIC**. Disponível em: <<http://stefanini.com/br/2014/06/tic-tendencias-mercado/>>. Acesso em: 27 out. 2014.

ZAMBALDE, André Luiz **A informática na modernização do sistema agroindustrial do café no Estado de Minas Gerais**. 2000. 182 p. Tese (Doutorado em Engenharia de Sistemas e Computação) – Universidade Federal do Rio de Janeiro, RJ.

ZAMBALDE, André Luiz; SCHNEIDER, Heloisa; LOPES, Marcos Aurélio; PAGLIS, Carlos Mauricio; BAMBINI, Martha Delphino. **Tecnologia da Informação no agronegócio**. SW Agro. Campinas: Embrapa Informática Agropecuária, 2011.