

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COORDENAÇÃO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

FILIPPE GUSTAVO SALLES LASECKI

SISTEMA web PARA GERENCIAMENTO DE CRIAÇÃO BOVINA

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2014

FILIPPE GUSTAVO SALLES LASECKI

SISTEMA web PARA GERENCIAMENTO DE CRIAÇÃO BOVINA

Trabalho de conclusão de curso apresentado à disciplina de Trabalho de Diplomação do curso superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para a obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Alexandre L'Erario.

CORNÉLIO PROCÓPIO

2014

RESUMO

LASECKI, Filipe Gustavo Salles Lasecki. **Sistema web para gerenciamento de criação bovina**. 2014. 71 f. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas), Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2014.

O mercado agropecuário está em constante crescimento, fazendo com que o grande volume de dados gerados necessite de gerenciamento. O software desenvolvido nesse projeto tem o objetivo de gerenciar uma propriedade pecuária mantendo os dados na *web*. O sistema é direcionado para o gerenciamento da produção, dos animais, dos eventos sanitários, das compras e das vendas. Os dados armazenados em nuvem podem ser acessados de qualquer dispositivo com acesso à internet e um *browser* para navegação. Esse trabalho foi desenvolvido utilizando a linguagem Java e o *framework JavaServer Faces* com o intuito de aprofundar os conhecimentos sobre a linguagem e sua associação com a tecnologia. As principais dificuldades na elaboração desse trabalho foram a falta de um escopo bem definido e uma documentação inicial escassa.

Palavras-chave: Sistema Web. Gerenciamento Pecuária. JavaServer Faces.

ABSTRACT

LASECKI, Filipe Gustavo Salles Lasecki. **web system for bovine farming management**. 2014. 71 f. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas), Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2014.

The livestock market is continuously growing, making that the large amount of generated data require management. The developed software in this project is designed to manage a livestock property while maintaining the data on the web. The system is directed to the managing of production, animals, health events, purchases and sales. The cloud stored data are accessible from any device with internet access and a web browser to navigation. This work was designed using Java and JavaServer Faces framework aiming to improving the knowledge on the language and its association with the technology. The main difficulties in the elaboration of this work were the absence of a well-defined scope and an sparse initial documentation.

Key words: web system. Cattle management. JavaServer Faces

LISTA DE FIGURAS

Figura 1 - Participação dos maiores rebanhos no efetivo mundial.	11
Figura 2 - Evolução do rebanho bovino por região geográfica no Brasil.	12
Figura 3 - Perfil da pecuária brasileira em 2011.	13
Figura 4 - Ciclo Scrum.	20
Figura 5 - Scrum Solo adaptado.	21
Figura 6 - Tarefas do projeto.	34
Figura 7 - Diagrama de Caso de Uso.	36
Figura 8 - Diagrama de entidade-relacionamento proprietário.	38
Figura 9 - Diagrama entidade-relacionamento animal.	39
Figura 10 - Diagrama entidade-relacionamento evento sanitário.	40
Figura 11 - Diagrama entidade-relacionamento comercialização.	41
Figura 12 - Diagrama entidade-relacional agenda.	42
Figura 13 - Diagrama entidade-relacionamento menu.	42
Figura 14 - Protótipo de Tela: Cadastro de Animal.	43
Figura 15 - Protótipo de tela: Cadastro de animal, aba Vacinação.	43
Figura 16 - Protótipo de tela: Cadastro de Venda.	44
Figura 17 - Diagrama de Classes.	45
Figura 18 - Diagrama de pacotes.	46
Figura 19 - Interface de login.	52
Figura 20 - Interface de recuperação de senha.	53
Figura 21 - Interface de cadastro de usuário.	53
Figura 22 - Interface inicial do sistema.	54
Figura 23 - Interface de cadastro de animal, aba Animal.	55
Figura 24 - Interface de cadastro de animal, aba Pesagem.	56
Figura 25 - interface de cadastro de animal, aba vacina.	57
Figura 26 - Interface de cadastro de animal, aba cobertura.	58
Figura 27 - Interface cadastro de animal, aba exames.	59
Figura 28 - Interface de cadastro de animal, aba Prole.	59
Figura 29 - Interface Venda de animais.	60
Figura 30 - Interface de compra de animais.	61
Figura 31 - Interface de compra de animais, aba parcelas.	61
Figura 32 - Interface de compra de animais, aba animais.	62
Figura 33 - Interface de cadastro de propriedade.	63
Figura 34 - Interface de cadastro de propriedade, aba fazendas.	63
Figura 35 - Interface de geração de relatório de comércio.	64
Figura 36 - Relatório de venda.	64

LISTA DE TABELAS

Tabela 1 - Limites do PostgreSQL	17
Tabela 2 - Requisitos removidos do planejamento.....	35
Tabela 3 - Sprints	51
Tabela 4 - Continuação da Tabela 3 - Sprints	52
Tabela 5 - Cronograma proposto.	70
Tabela 6 - Cronograma Final.....	71

SUMÁRIO

1	INTRODUÇÃO.....	8
1.1	OBJETIVO	8
1.2	MOTIVAÇÃO.....	9
1.3	ESTRUTURA DO TRABALHO.....	10
2	FUNDAMENTAÇÃO TEÓRICA	11
2.1	PECUÁRIA.....	11
2.2	TECNOLOGIAS UTILIZADAS.....	15
2.2.1	Recursos de hardware.....	15
2.2.2	Recursos de software	15
2.3	METODOLOGIA DE DESENVOLVIMENTO.....	19
3	DESENVOLVIMENTO	22
3.1	SISTEMA	22
3.2	REQUISITOS	22
3.2.1	Requisitos funcionais.....	23
3.2.2	Requisitos não funcionais.....	32
3.2.3	Planejamento.....	33
3.3	Modelagem	35
3.4	Implementação.....	46
3.5	APRESENTAÇÃO DO SISTEMA.....	52
4	CONSIDERAÇÕES FINAIS.....	65
4.1	DIFICULDADES ENCONTRADAS.....	65
4.2	APRENDIZADO	65

1 INTRODUÇÃO

Segundo o relatório *Food and Farming Futures* (FORESIGHT, 2011), a população passará dos 7 bilhões atuais para 9 bilhões até 2050, aumentando a demanda de água, energia e comida. O mesmo relatório afirma que o consumo per capita/ano de carne pode passar dos 37kg para 52 kg em 2050. Para suprir essa necessidade estima-se que será necessário aumentar a produção de alimentos em 40%.

Dados levantados em 2011 (RABELO, 2012) apontaram o Brasil como tendo o segundo maior rebanho de gado do mundo, com cerca de 180 milhões de cabeças. Desse modo, o agronegócio representa 30% do PIB, 40% das exportações e 37% dos empregos no Brasil. Esses números mostram que assim como outros setores da economia, a pecuária deve buscar meios mais eficientes de gerenciamento a fim de otimizar a sua produção.

Os pecuaristas devem ver a sua produção como um empresário vê a sua empresa. Ele deve gerenciar seus recursos, produtos e clientes de uma forma rápida e eficaz, sem que exista perda de informações. Essa forma de gerenciamento servirá para auxiliá-lo em futuras decisões.

Uma boa gestão do negócio pode fazer toda a diferença no mundo globalizado, em que informações bem organizadas se tornam uma importante e valiosa arma na tomada de decisões. Uma forma de organizar essas informações é utilizar ferramentas que facilitem o seu armazenamento, gerenciamento e sua disseminação.

SVEIBY (2003) diz que “a Gestão do Conhecimento não é mais uma moda de eficiência operacional. Faz parte da estratégia empresarial”. Como medida utilizada para reduzir a perda de informações, destaca-se o uso de sistemas informalizados. Esses sistemas facilitam o gerenciamento e armazenamento de informações através da automação da manutenção e pela facilidade de acesso de diferentes lugares.

1.1 OBJETIVO

Esse trabalho descreve o desenvolvimento do software de armazenamento e consulta de dados relacionados à área de bovina cultura.

Também apresenta a revisão de conceitos relacionados ao tema e os artefatos gerados para documentação do sistema, assim como a apresentação das dificuldades encontradas e os conhecimentos adquiridos na criação do mesmo.

O software desenvolvido nesse projeto tem por finalidade o gerenciamento bovino. Isso é feito através do armazenamento de informações sobre o gado, compra e venda de animais, parceiros de negócio e eventos sanitários como vacinação, exames e controle de peso. Permitindo que o usuário tenha acesso a esses dados de uma maneira organizada através de relatórios extraídos do sistema.

O objetivo desse projeto é trazer para o produtor pecuário um modo simples porém eficaz de gerenciar de seu rebanho.

1.2 MOTIVAÇÃO

A pecuária é uma atividade cada vez mais complexa e extensa, que exige de seus administradores um alto grau de conhecimento de suas propriedades. Existem no mercado inúmeros softwares de gerenciamento. No entanto existem poucas empresas de desenvolvimento voltadas para o gerenciamento do agronegócio. Muitas dessas empresas de software são ineficientes tanto em qualidade quanto em produtividade (CAMPOS, 2001), pelo fato de faltar conhecimento por parte delas a respeito do negócio, fazendo com que a criação de ferramentas de gestão para o setor acabe se tornando limitada.

As ferramentas que ocupam uma melhor posição no mercado são a Procreate (PROCREARE, 2013) uma ferramenta muito robusta, porém *off-line*, fazendo com que todas as informações fiquem restritas à apenas uma máquina; o Bonanza (CPT SOFTWARE, 2013) que não controla eventos sanitários e a A3Pecuaria (SOFTFOR, 2013), um sistema online que requer instalação de software e com foco em compra e venda de animais.

Atualmente o acesso a computadores nas áreas rurais se tornou mais comum, fazendo com que o uso dessas ferramentas fique cada vez mais fácil (CETIC, 2011). O PNBL (Programa Nacional de Banda Larga) propõe até em 2014, levar internet para mais de 40 milhões de domicílios nas áreas urbanas e rurais (BRASIL, 2010). Existe também o fato da terceira geração de tecnologia móvel (3G) estar trazendo a internet para o dia a dia no campo. Esses fatores ajudam a impulsionar a utilização de

ferramentas computacionais no campo além de permitirem a utilização de softwares online (CETIC, 2011).

Softwares *online* têm a vantagem de possuírem base de dados em nuvens, que faz com que as informações neles armazenadas possam ser acessadas de qualquer lugar, sem que exista dependência de apenas um computador. Embora exista essa vantagem sobre a tecnologia *desktop*, não existe nenhum software web brasileiro voltado para a pecuária disponível para uso nos dias de hoje.

1.3 ESTRUTURA DO TRABALHO

Este documento está dividido em quatro capítulos, sendo este voltado á contextualizar o leitor, destacar o objetivo da escrita e descrever as principais motivações que despertaram o interesse nesse estudo.

O capítulo dois, formado por parte das pesquisas realizadas durante o desenvolvimento, apresenta conceitos necessários tanto para o desenvolvimento quanto para melhor compreensão do trabalho.

O capítulo três descreve como foi realizado o desenvolvimento do sistema, tanto na parte de análise de requisitos e modelagem do sistema, quanto a parte de codificação do mesmo.

O capítulo quatro apresenta a conclusão sobre o trabalho, mostrando as dificuldades encontradas e o aprendizado adquirido durante o seu desenvolvimento.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os conceitos utilizados para o desenvolvimento do sistema, apresentando os conceitos técnicos da pecuária nos dias atuais e as ferramentas utilizadas para o desenvolvimento do software proposto nesse projeto.

2.1 PECUÁRIA

A pecuária abrange toda atividade de criação de animais para fins econômicos. Dentro da pecuária a criação de gado é classificada em bovino, suíno ou equino cultura. Por sua utilidade ao homem como meio de transporte e fornecedora de alimentos e couro, a criação de gado bovino é a mais difundida. (COELHO, 1990).

O Brasil possui o segundo maior rebanho bovino do mundo com mais de 200 milhões de cabeças, representando 21% do rebanho mundial, perdendo somente para a Índia, que representa 32% do rebanho mundial, com 320 milhões de cabeças (NETO, 2012). A Figura 1 mostra a participação dos maiores rebanhos no efetivo mundial.

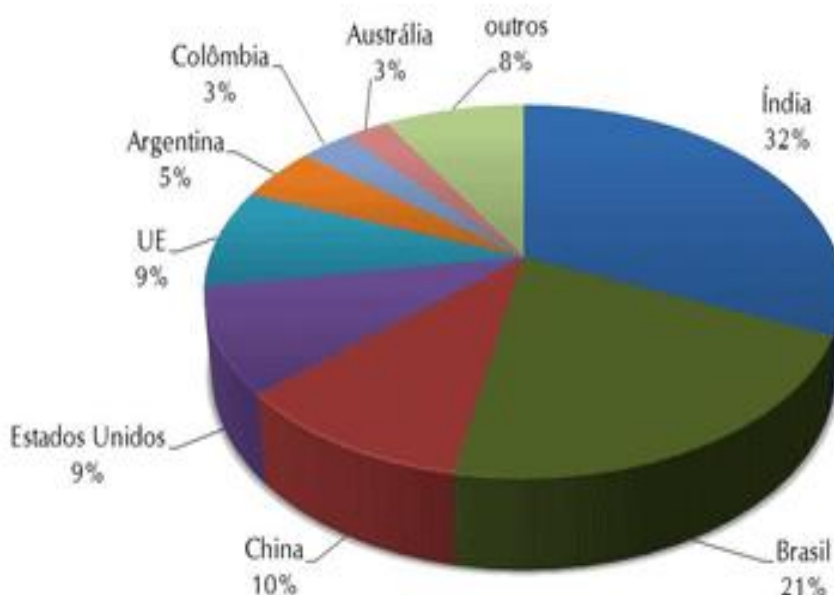


Figura 1 - Participação dos maiores rebanhos no efetivo mundial.
Fonte: NETO (2012).

A Figura 2 é um gráfico que representa a distribuição do rebanho bovino no Brasil nos anos de 1996 e 2007. Graças à vegetação do tipo cerrado a região Centro-

oeste é a que tem a maior participação no rebanho, representando mais de um terço das cabeças de gado no território brasileiro. Apesar de a região norte ser a menor produtora de bovinos, seu rebanho vem crescendo com o passar dos anos, tendo um aumento de 8% (FOCUS,2010).

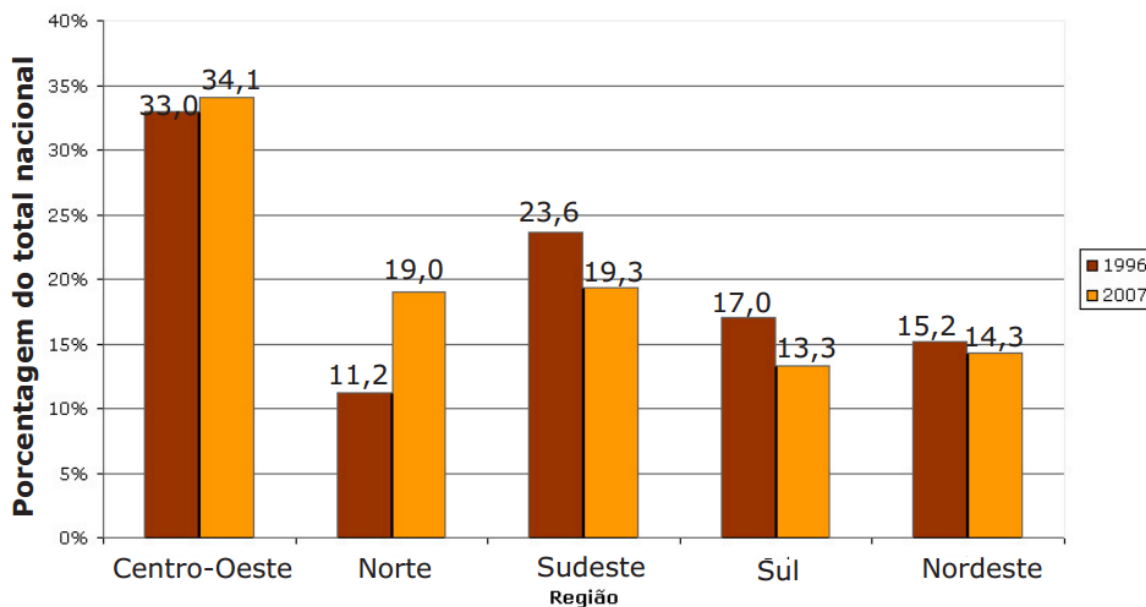


Figura 2 - Evolução do rebanho bovino por região geográfica no Brasil.

Fonte: FOCUS (2010)

A Figura 3 ilustra os dados referentes à pecuária brasileira no ano de 2011. O Brasil possuía uma área de 172 milhões de hectares de pastagem, com uma ocupação de 1,2 cabeças de gado por hectare. No ano o mercado interno movimentou o equivalente a 7,6 milhões de toneladas de carne. As exportações de carne chegaram a 1,5 milhões de toneladas, sendo dividida em carne *in natura* que representa 71% das exportações, carne industrializada, 17%, miúdos e outros somam 11% das exportações. Com esses números o Brasil conseguiu assumir a posição de maior exportador de carne bovina do mundo. Além da exportação de animais abatidos o Brasil também exporta cerca de 405 mil cabeças no ano.

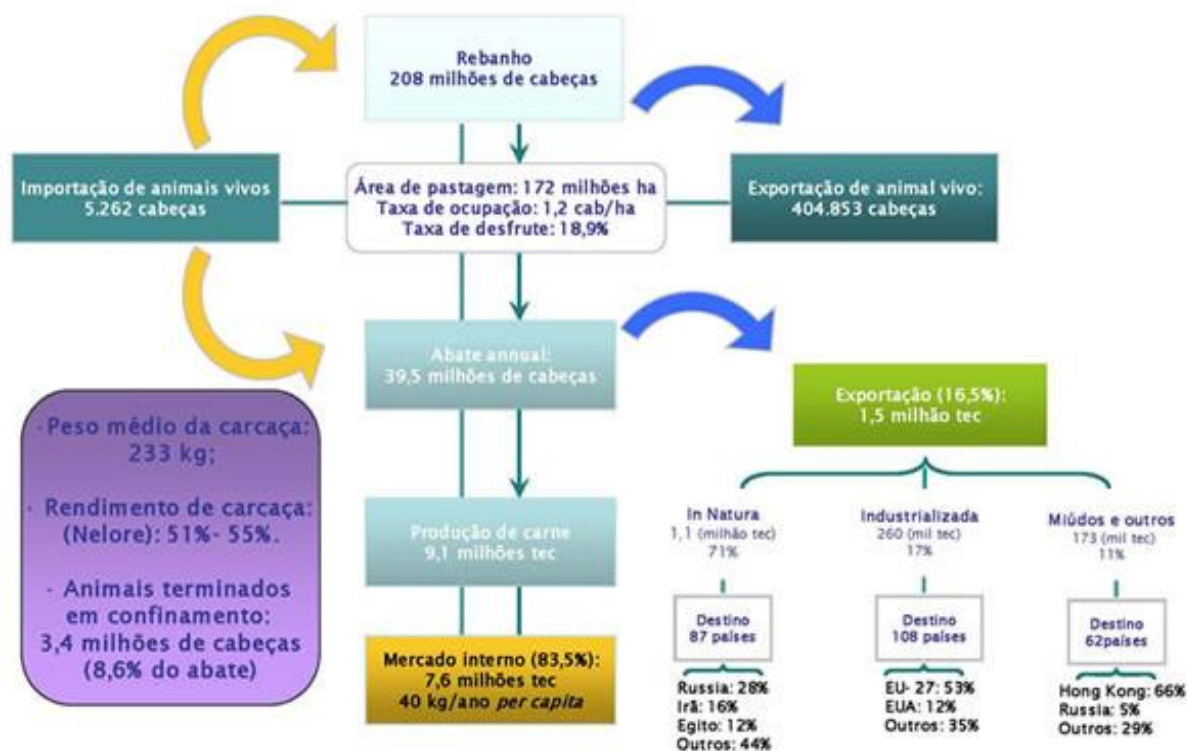


Figura 3 - Perfil da pecuária brasileira em 2011.

Fonte: ABIEC (2011)

A pecuária bovina, nos dias atuais, é dividida em dois seguimentos de acordo com o produto retirado para comércio. A bovino cultura leiteira prioriza a criação de animais para produção de leite e seus derivados. A bovino cultura de corte cria animais para a produção de carne e couro. Dentro de cada um desses dois seguimentos existe uma rede de comercialização de matrizes e reprodutores (MARINEI, 2013).

O produtor determina o tipo de criação que deseja usar, sendo influenciado de acordo com o clima, do seguimento ou da escassez de alimento, outro fator que pode influenciar é a condição financeira do produtor. Os tipos de criação podem ser extensiva ou intensiva.

Na criação intensiva o animal é criado em ambientes fechados onde é alimentado com ração, essa atividade também pode ser chamada de confinamento. Essa atividade exige uma maior necessidade de investimento e maior gasto anual com a produção (SIMÕES et al. 2009).

O tipo de criação extensiva, o animal é criado solto, onde se alimenta de pastagens. Nesse tipo de criação os gastos anuais são menores, pois não exige grandes investimentos em tecnologia (SIMÕES et al. 2009).

O pecuarista pode também optar pela criação semi-intensiva, que utiliza das duas técnicas de criação. Estima-se que 97% das cabeças de gado no Brasil são criadas no sistema extensivo e apenas 3% no sistema intensivo (ABIEC, 2011).

O Brasil possui uma grande variedade de raças de bovinos, destacando-se as de origem europeia e indiana, cada uma com características próprias. Existem inúmeras pesquisas que visam o melhoramento dessas raças, para que o produtor alcance maior produtividade.

De acordo com a Embrapa, de nada adianta ter uma boa pastagem e animais de ótima linhagem sanguínea, sendo que eles podem ficar doentes, se não tratados corretamente. Por isso o manejo sanitário é fundamental, para manter uma boa qualidade do rebanho. Os eventos sanitários são atividades veterinárias que visam à prevenção e manutenção da saúde do rebanho. Entre os eventos sanitários estão: vacinação, vermifugação, controle de carrapatos, controle da mosca do chifre, tratamento de doenças reprodutivas (EMBRAPA, 2006).

2.2 TECNOLOGIAS UTILIZADAS

Nesta sessão são descritos os recursos de hardware e software que foram pesquisados, aprendidos, utilizados e explorados para a melhor realização do trabalho.

2.2.1 Recursos de hardware

- Dell Aspire
- Processador i7-3537U 2.0 Ghz
- 8 Gb
- 1 Gb vídeo

2.2.2 Recursos de software

Nessa sessão são descritos os recursos de software utilizados para o desenvolvimento do sistema.

2.2.2.1 Linguagem

A linguagem escolhida para o desenvolvimento desse sistema foi a linguagem Java, uma linguagem de programação orientada a objeto e uma plataforma de computação criada pela Sun Microsystems e lançada pela primeira vez em 1995 (ORACLE, 2013).

Por ser rápido e seguro o Java é utilizado nos mais diversos lugares, como laptops, datacenters, consoles de jogos, supercomputadores científicos, telefones celulares e até na internet. Permite a portabilidade de aplicativos entre ambientes computacionais, possui um sistema sólido e um desempenho eficiente, isso faz com que hoje ela seja utilizada por mais de 9 milhões de programadores (ORACLE, 2013).

A linguagem Java fornece um conjunto completo de classes para desenvolver sistemas web e desktop, graças a essas classes o tempo para o desenvolvimento é reduzido e traz uma melhor qualidade para o sistema desenvolvido (MENDES, 2009).

As principais características da linguagem são:

- Orientação a objetos;
- Suporte à *multithreads*;

- Linguagem interpretada;
- Linguagem Neutra quanto à arquitetura;
- Portável;
- Linguagem Robusta;
- Linguagem Segura;
- Linguagem de alto desempenho.

2.2.2.2 IDE

O sistema foi desenvolvido através da IDE NetBeans, um ambiente de desenvolvimento *open source* que oferece todas as ferramentas necessárias para a criação de aplicações desktop, web e mobile. Essa ferramenta permite ao programador escrever, compilar, debugar e instalar aplicações. Ela também possui um grande conjunto de bibliotecas, módulos e APIs que auxiliam o programador na criação do código. Possui uma documentação vasta e bem organizada e pode ser encontrada em português

A IDE NetBeans é excelente para quem quer desenvolver uma aplicação web de uma forma rápida e simples, pois ela auxilia o programador durante todo o desenvolvimento do código, mostrando falhas na digitação, variáveis não declaradas, métodos inexistentes, atalhos de código, e auxílio na importação de bibliotecas, além de aceitar diversos plug-ins necessários para o desenvolvimento de projetos, entre outras funcionalidades. (GOLÇAVES, 2007)

Para a criação dos layouts dos relatórios extraídos do sistema foi utilizado a ferramenta Ireport, uma ferramenta gratuita e *open source*, que auxilia na criação desses layouts, que podem conter gráficos, imagens e sub-relatórios. (TOFFOLI, 2013) O Ireport foi escolhido devido a sua interface ser bastante intuitiva, facilitando o aprendizado de suas funcionalidades e por consequência facilitando o desenvolvimento dos relatórios utilizados nesse trabalho.

2.2.2.3 Banco de dados

Para o armazenamento das informações inseridas através do sistema foi utilizado o banco de dados PostgreSQL, um sistema *open source* de gerenciamento de banco de dados objeto relacional, utilizado para armazenar informações e

administrar o acesso a elas. Pode ser utilizado nos sistemas operacionais Windows e Linux (MILANE,2008).

Essa ferramenta teve origem em 1986 na Universidade de Berkeley, na Califórnia, seu nome original era POSTGRES. O projeto foi financiado por diversos órgãos com o intuito de criar um modelo e regras para um novo sistema de armazenagem de dados. Em 1994 foi introduzida a linguagem SQL na ferramenta, pelos desenvolvedores Andrew Yu e Jolly Chen (MILANE,2008).

Esse banco foi escolhido pois não limita o tamanho do seu banco de dados, podendo armazenar uma quantidade ilimitada de informações como pode ser visto na Tabela 1. Um dos pontos fortes do PostgreSQL é a segurança, sendo ele recomendado para aplicações complexas, que exijam o controle de grande volume de dados (POSTGRESQL, 2009).

Tabela 1 - Limites do PostgreSQL

Objeto	Limite
Banco	Ilimitado
Tabela	32 Terabytes
Linha	1,6 Terabytes
Campo	1 Gigabyte
Número de linhas	Ilimitado
Número de colunas	250-1600 (depende do tipo de coluna)
Número de Índices	Ilimitado

Fonte: (LUCCAS,2006)

Algumas das principais características desse banco de dados são:

- Suporta transações;
- Suporta Replicação;
- Suporta cluster;
- Suporta Multithreds;
- Possui Segurança SSL e criptografia;

No mais, a utilização da licença BSD¹ (*Berkeley Software Distribution*), permite que esse SGBD seja utilizado livremente em aplicações pessoais e comerciais. Outra característica é também possuir bibliotecas e drivers para várias linguagens de programação, alguns exemplos são: C/C++, Java/JSP, PHP, ASP, .Net, Perl, Python, Ruby e Tcl (MILANE,2008).

Como ferramenta de administração de banco de dados foi utilizado o PgAmin, uma ferramenta gratuita que atende as necessidades do usuário que inclui um editor com destaque de sintaxe SQL, um editor de código do lado do servidor, um SQL / batch / shell agente de agendamento de trabalho, o apoio ao mecanismo de replicação Slony-I².

2.2.2.4 Modelagem

Os diagramas neste trabalho apresentados foram construídos a ferramenta de modelagem Visual Paradigm, uma ferramenta de modelagem corporativa, que fornece meios para criação de diversos tipos de diagrama UML, como diagrama de caso de uso, diagrama de classes, diagrama de sequência entre outros, essa ferramenta é utilizada para organização e documentação no processo de desenvolvimento de software. Para esse trabalho foi utilizada a versão 11 Community, uma versão gratuita, embora não possua todos os recursos das versões pagas, ela aderiu bem ao objetivo desse trabalho (ARAUJO, 2007).

Para a modelagem do diagrama de entidade-relacionamento foi utilizada a ferramenta WorkBench MySql, uma ferramenta de modelagem e administração de banco de dados MySql. Ela pode ser utilizada nas plataformas Linux, Windows e Mac OS X. (MYSQL, 2013) Apesar de ser uma ferramenta para o banco de dados MySql, essa ferramenta pode ser utilizada perfeitamente para a construção da estrutura do banco de dados PostgreSQL.

Para a criação dos mapas mentais foi utilizada a ferramenta Xmind, um software *open source*, distribuído gratuitamente, ele pode ser encontrado nas versões para Windows, Mac e Linux. Tem como funcionalidade a criação de mapas mentais, que são meios de gerenciar ideias. Uma grande vantagem no Xmind é que ele possui

1 O detentor dos direitos autorais cede os direitos comerciais, mas exige os créditos pela autoria e propriedade

2 Slony-I é um sistema de "mestre de vários escravos" de replicação para PostgreSQL.

diversos tipos de mapas mentais, permitindo também a exportação para diversos tipos de extensão. (XMIND, 2013)

Para controle das atividades desempenhadas durante a implementação do sistema foi utilizada a técnica Kanban, uma técnica que auxilia o desenvolvedor a visualizar as suas tarefas e o andamento das mesmas. Essa técnica tem como origem os sistemas de cartão utilizados na indústria de produção. Sua implementação se resume em 3 etapas: visualizar os processos, limitar o trabalho em processo, e gerenciar o tempo que cada atividade leva em cada fase (MARIOTTI, 2012).

Para auxiliar na utilização da técnica do Kanban foram utilizadas as ferramentas POMODORO, uma ferramenta utilizada para gerenciamento de tempo em um projeto, e o KanbanWorkflow, uma ferramenta online que mistura a técnica de gerenciamento de processos Kanban e a técnica de gerenciamento de tempo, o POMODORO.

2.3 METODOLOGIA DE DESENVOLVIMENTO

O método de desenvolvimento utilizado nesse projeto foi uma adaptação da metodologia ágil Scrum, o Scrum Solo. O método de desenvolvimento *Scrum* permite que o projeto seja dividido em fases, em que no final de cada fase é entregue uma parte funcional do projeto.

Os principais conceitos e envolvidos (SCHWABER et al, 2013):

- *Scrum Master* tem a responsabilidade de garantir que o *Scrum* seja aplicado, fazendo com que o *Scrum Team* siga as regras, as teorias e as práticas do *Scrum*.
- *Scrum Team* é a equipe de desenvolvimento, eles são responsáveis pelo desenvolvimento de uma versão utilizável do produto no final de cada *Sprint*.
- *Product Owner* é a pessoa responsável pelo projeto, sendo ela o responsável por gerenciar os *Product Backlog*.
- *Product Backlog* é uma lista de atividades e funções do sistema que irá ser desenvolvido, o *Product Backlog* pode sofrer alterações no decorrer do projeto, para tornar o produto mais apropriado, competitivo e útil

- *Sprint* é o período de tempo de um mês ou menos, no qual uma parte funcional do software é criada.
- *Sprint Backlog* é um conjunto de itens do *Product Backlog*, que serão desenvolvidos em cada *Sprint*.
- Reuniões diárias são eventos de aproximadamente 15 minutos, em que o *Scrum Team* pode sincronizar as atividades revendo o que foi feito nas últimas 24 horas e planejar as atividades das próximas 24 horas.

A Figura 4 representa a metodologia de desenvolvimento *Scrum* e suas fases.

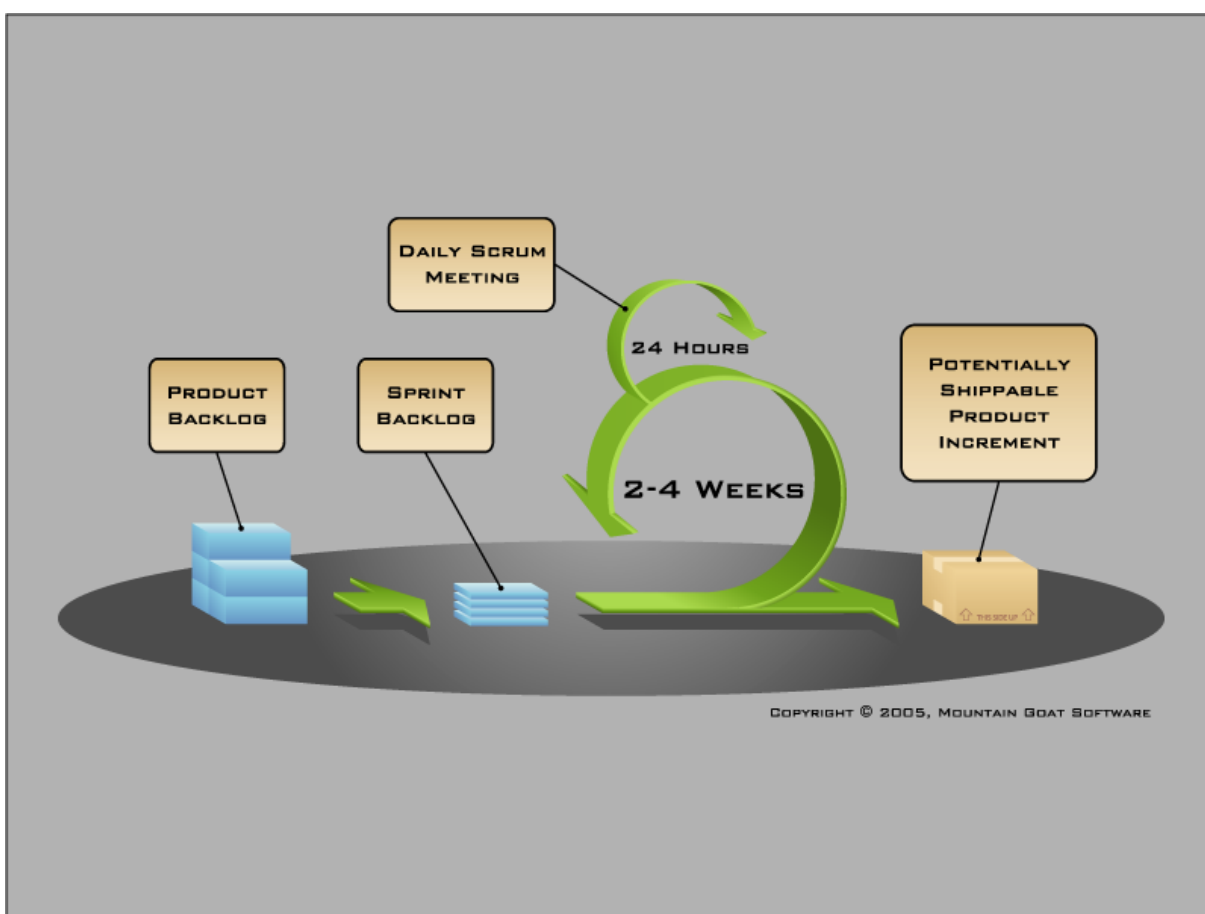


Figura 4 - Ciclo Scrum.
Fonte: MOUNTAIN, 2005

O Scrum solo é um processo de desenvolvimento voltado para programadores solos, em que o mesmo tem a função de *Scrum Master*, de *Product Owner* e de desenvolvedor. Diferentemente do *Scrum*, o *Scrum solo* possui *Sprints* de desenvolvimento com duração de uma semana, no qual não existem reuniões diárias, apenas revisão diárias do que foi desenvolvido.

As revisões diárias tem o papel de controlar o processo de desenvolvimento, verificando se os prazos estão sendo cumpridos e todas as funcionalidades desenvolvidas estão de acordo com as especificadas.

Para cada *Sprint* são realizados testes paralelos ao desenvolvimento e no final são realizados testes e validações de todas as funcionalidades desenvolvidas.

No final de cada *Sprint* e após os testes são entregues incrementos funcionais do software.

Na Figura 5 é possível ver o processo de desenvolvimento do *Scrum solo* utilizado nesse projeto.

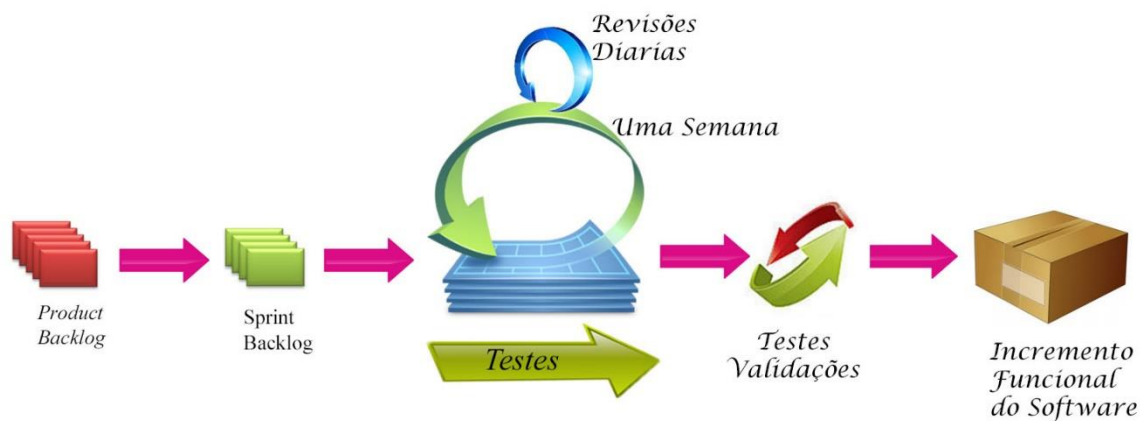


Figura 5 - Scrum Solo adaptado.
Autor: Aatoria própria.

Além do *Product Backlog*, do *Sprint Backlog* e das partes funcionais do software geradas dentro do processo *Scrum*, também foi gerado o cronograma final das atividades desempenhadas dentro do desenvolvimento do sistema, esse cronograma foi gerado através do controle diário realizado em cada *Sprint* e catalogado na ferramenta *kanbanWorkFlow*.

3 DESENVOLVIMENTO

Este capítulo apresenta os requisitos funcionais e não funcionais definidos no projeto, as fases de modelagem e implementação do sistema e a apresentação do sistema concluído.

3.1 SISTEMA

O sistema desenvolvido nesse trabalho é um sistema de gerenciamento pecuário web, criado com o intuito de auxiliar o pecuarista na criação e manejo de animais do tipo bovino. Com esse sistema o usuário pode manter as informações relevantes de sua propriedade e de seus parceiros de negócio.

A escolha da tecnologia web deu-se ao fato da facilidade de acesso aos dados, já que um sistema web permite que o usuário acesse seus dados de qualquer hardware com acesso à internet e um *browser* para navegação. Outro fator que culminou na escolha foi o fato de haver uma escassez de softwares na área da pecuária utilizando essa tecnologia.

As principais funcionalidades do sistema são:

- Manter animais;
- Manter criador;
- Manter venda;
- Manter compra;
- Visualizar relatórios de animais;
- Visualizar relatórios de compra e venda;
- Visualizar relatórios de eventos sanitários.

3.2 REQUISITOS

Nessa sessão são descritos os requisitos funcionais e não funcionais levantados para a criação do sistema.

Os dados de entrada que se seguirem de um * (asterisco) deverão ser preenchidos obrigatoriamente pelo usuário.

3.2.1 Requisitos funcionais

RF. 1: Cadastro de Usuário

Descrição: O administrador do sistema pode cadastrar quantos usuários desejar, para que o mesmo possa acessar os dados inseridos no sistema.

Entrada: Nome completo*, CPF/CNPJ*, inscrição estadual, cidade*, nome de usuário*, E-mail*, senha*.

Processo: Os dados do usuário serão validados para que todas as informações necessárias ao cadastro sejam inseridas, após a validação o sistema irá inserir os dados no banco de dados.

Saída: Mensagem confirmando se o cadastro foi bem sucedido ou não.

Prioridade: Alta.

RF. 2: Acesso do usuário

Descrição: Para o acesso ao sistema o usuário deverá entrar com o usuário de acesso e a senha correspondente.

Entrada: Usuário de acesso* e senha*.

Processo: O sistema deve validar se o usuário e a senha correspondem.

Saída: Caso os dados sejam corretos o usuário terá acesso a página inicial do sistema, caso estejam incorretos o sistema deverá mostrar uma mensagem de erro ao usuário.

Prioridade: Alta.

RF. 3: Recuperação de senha

Descrição: Caso o usuário esqueça a senha, ele deve ter um modo de recuperar a senha

Entrada: usuário de acesso*, CPF/CNPJ*.

Processo: O sistema deve verificar se o usuário está cadastrado no sistema e verificar se a informação no CPF/CNPJ corresponde aquele usuário, caso corresponda o sistema deve enviar um e-mail à conta de e-mail cadastrada, com a nova senha do usuário.

Saída: Mensagem de confirmação de envio de e-mail caso as informações estejam corretas ou mensagem de erro caso alguma informação esteja incorreta.

Prioridade: Alta.

RF. 4: Manter animais

Descrição: O sistema deve permitir que sejam cadastros, editados e excluídos animais

Entrada: Código do animal, número do brinco*, nome, sexo*, nascimento*, raça*, pelagem*, lote e destinação*.

Processo: Os dados do animal devem ser validados para que sejam cadastradas todas as informações necessárias, após a validação os dados devem ser inseridos no banco de dados.

Saída: Se os dados estiverem corretos e forem inseridos no banco de dados o sistema deverá mostrar uma mensagem de confirmação de cadastro ou uma mensagem de erro informando o problema ao cadastrar os dados.

Prioridade: Alta.

RF. 5: Manter lote de animais

Descrição: O sistema deve permitir que sejam cadastrados animais em grande quantidade.

Entrada: Número de animais machos e fêmeas*, nascimento*, raça*, pelagem* e a destinação*.

Processo: O sistema deve cadastrar um animal para cada animal do lote, todos os animais cadastrados terão os mesmos dados, apenas o sexo do animal vai variar de acordo com a quantidade de fêmeas e machos do lote.

Saída: Mensagem de confirmação de cadastro ou mensagem de erro caso o cadastro não tenha sido concluído.

Prioridade: Alta.

RF. 6: Manter vacinação

Descrição: O sistema deve permitir que sejam cadastradas as vacinações referentes ao animal.

Entrada: animal*, vacina*, veterinário, dose*, data de aplicação*.

Processo: O sistema deve validar as informações inseridas pelo usuário, para que não haja inconsistência nos dados ou dados duplicados, após a validação o sistema deverá inserir as informações no banco de dados.

Saída: A vacinação cadastrada deve aparecer na lista de vacinações do animal caso o cadastro seja bem sucedido, caso contrário deve aparecer uma mensagem de erro.

Prioridade: Média.

RF. 7: Manter reprodução

Descrição: O sistema deve permitir o cadastro de informações referentes a reprodução dos animais.

Entrada: número do animal fêmea e macho*, data da cobertura*, tipo de cobertura*.

Processo: O sistema deve validar para que sejam inseridos todos os dados necessários para o cadastro da reprodução e não deve permitir que os dados sejam duplicados

Saída: A reprodução cadastrada deve aparecer na lista de reproduções do animal caso o cadastro seja bem sucedido, caso contrário deve aparecer uma mensagem de erro.

Prioridade: Média.

RF. 8: Cadastro de pesagem de animal

Descrição: O sistema deve permitir que seja inseridas informações sobre a pesagens dos animais.

Entrada: código do animal*, peso*, data de pesagem*.

Processo: o sistema deve validar para que todos os dados sejam inseridos para o cadastro da pesagem e não deve permitir pesagens no mesmo dia para o mesmo animal.

Saída: Caso o cadastro seja bem sucedido a pesagem cadastrada deve aparecer na lista de pesagem do animal, caso contrário deve aparecer uma mensagem de erro.

Prioridade: Média.

RF. 9: Cadastro de morte de animal

Descrição: O sistema deve permitir o registro de morte de animais, juntamente com informações a respeito dela.

Entrada: código do animal*, data da morte*, motivo da morte*, observação.

Processo: O sistema deve validar se o animal já tenha sido cadastrado como morto, e deve validar se o código do animal, a data da morte e o motivo da morte tenham sido inseridos para cadastrar o óbito.

Saída: Mensagem de cadastro bem sucedido ou mensagem de erro.

Prioridade: Media.

RF. 10: Cadastro de desaparecimento de animal

Descrição: O sistema deve permitir o registro o desaparecimento de animal, juntamente com os dados do desaparecimento.

Entrada: código do animal*, data de desaparecimento*, possível causa*, observação.

Processo: O sistema deve validar para que todos os dados sejam inseridos no cadastro de desaparecimento.

Saída: Mensagem de cadastro bem sucedido ou mensagem de erro.

Prioridade: Media.

RF. 11: Cadastro de exames

Descrição: Caso algum animal passe por exames, o sistema deve ser capaz de armazenar as informações referentes a esse exame.

Entrada: código do animal*, exame*, data do exame*, veterinário*.

Processo: o sistema deve validar se todos os dados necessários para o cadastro foram inseridos, após a validação o sistema deve salvar as informações no banco de dados.

Saída: Caso o cadastro seja bem sucedido o exame cadastrado deve aparecer na lista de exames do animal, caso contrário deve aparecer uma mensagem de erro.

Prioridade: Media

RF. 12: Cadastro de tratamento de doenças

Descrição: Caso o animal necessite ter tratamento para alguma doença, o sistema deve ser capaz de armazenar as informações referentes ao tratamento

Entrada: código do animal*, doença*, tratamento*, veterinário, data de início*, data de término*.

Processo: o sistema deve validar se todos os dados necessários para o cadastro de animais tenham sido inseridos, após a validação o sistema deve inserir os dados no banco de dados.

Saída: Caso o cadastro seja bem sucedido o tratamento cadastrado deve aparecer na lista de tratamentos do animal, caso contrário deve aparecer uma mensagem de erro.

Prioridade: Media.

RF. 13: Cadastro de compra

Descrição: O sistema deve permitir armazenar informações referentes a compra de animais individuais e em lote, também deve permitir que as compras sejam parceladas

Entrada: Valor da compra*, número de animais comprados*, tipo de compra*, número de parcelas, valor de cada parcelas, juros, moeda*, vencimento, vendedor*, número do brinco (compra de apenas um animal), nome (compra de apenas um animal), sexo*, nascimento*, raça*, pelagem*, destinação*.

Processo: Para realizar a compra de um ou mais animal, os dados desses animais devem ser previamente cadastrados no sistema, para que depois sejam efetuado o registro da compra. O sistema deve validar para que todos os dados referentes a compra seja inseridos, após a validação os sistema deve inserir os dados da compra no sistema.

Saída: Mensagem de cadastro bem sucedido ou mensagem de erro.

Prioridade: Alta.

RF. 14: Cadastro de venda

Descrição: O sistema deve permitir armazenar informações referentes a venda de animais individuais e em lote, também deve permitir que as vendas seja parceladas

Entrada: Valor da venda*, número de animais vendidos*, tipo de venda*, número de parcelas, valor de cada parcelas, moeda*, vencimento, comprador*, código do animal vendido*, tipo de pagamento*.

Processo: o sistema deve validar os dados referentes a venda e os dados referentes ao animal, após a validação o sistema deve cadastrar os dados da venda e alterar o proprietário dos animais vendidos.

Saída: Mensagem de cadastro bem sucedido ou mensagem de erro.

Prioridade: Alta.

RF. 15: Cadastro de agenda

Descrição: O sistema deve ter a função de registrar eventos e disponibilizar esses registros para o usuário.

Entrada: data do evento*, hora do evento*, descrição*.

Processo: o sistema deve validar se todos os dados foram inseridos e se a data do evento é posterior a data atual, após a validação o sistema deve inserir os dados no banco de dados.

Saída: Caso o cadastro seja bem sucedido o evento cadastrado deve aparecer na lista de eventos, caso contrário deve aparecer uma mensagem de erro.

Prioridade: Média.

RF. 16: Manter Lactação

Descrição: o sistema deve armazenar informações referentes a lactação das matrizes.

Entrada: código do animal*, litros de leite retirado*, taxa de gordura*

Processo: o sistema deve validar para que todos os dados sejam inseridos corretamente, após a validação o sistema deve inserir os dados no banco de dados.

Saída: Caso o cadastro seja bem sucedido o evento cadastrado deve aparecer na lista de eventos, caso contrário deve aparecer uma mensagem de erro.

Prioridade: Media.

RF. 17: Manter Dieta

Descrição: O sistema deve armazenar informações a respeito da alimentação do animal, tanto alimentação comum entre eles quanto alimentação exclusiva de algum animal.

Entrada: tipo de dieta*, alimentos da dieta*, objetivo*, período*.

Processo: O sistema deve validar para que todos os dados sejam inseridos corretamente, após a validação o sistema deve inserir os dados no banco de dados. Após o cadastro o sistema deve permitir que o usuário possa ligar a dieta a um animal ou um grupo de animais.

Saída: Mensagem de cadastro bem sucedido ou mensagem de erro

Prioridade: Baixa.

RF. 18: Controle de gastos alimentícios

Descrição: O sistema deve armazenar os gastos relativos a alimentação dos animais.

Entrada: dieta cadastrada, valor da dieta.

Processo: o Sistema deve através das dietas cadastradas e animais que utilizam da dieta, calcular os gastos relativos a esses tratamentos.

Saída: sem saída.

Prioridade: Baixa.

RF. 19: Controle de gastos médicos

Descrição: O sistema deve através do cadastro de exames, tratamentos e vacinas, disponibilizar dados referentes ao gastos com esses eventos sanitários, tanto para animal individual, quanto para vários animais.

Entrada: Evento sanitário cadastrado, valor do evento.

Processo: Ao cadastrar um evento sanitário o sistema deve armazenar o valor referente ao evento e disponibilizar essa informação ao usuário.

Saída: Sem saída.

Prioridade: Media.

RF. 20: Manter produtores

Descrição: O sistema deve permitir o armazenamento de dados de outros produtores rurais.

Entrada: nome*, inscrição estadual*, CPF/CNPJ, telefone*, estado*, cidade*, logradouro*, bairro, endereço*, número.

Processo: o sistema deve validar se todos os dados necessários para o cadastro do proprietário estão corretos e inserir as informações no banco de dados

Saída: Caso o cadastro seja bem sucedido deve aparecer uma mensagem de confirmação, caso contrário o sistema deve mostrar o erro que impossibilitou o cadastro.

Prioridade: Alta.

RF. 21: Manter propriedades de produtores

Descrição: O sistema deve permitir o armazenamento de dados referentes as propriedades dos produtores rurais cadastrados no sistema.

Entrada: nome*, tamanho*, estado*, cidade*, logradouro*, bairro, endereço*, número.

Processo: o sistema deve validar se todos os dados necessários para o cadastro da propriedade estão corretos e inserir as informações no banco de dados

Saída: Caso o cadastro seja bem sucedido a propriedade cadastrada deve aparecer na lista de propriedades do produtor, caso ocorra um erro uma mensagem deve aparecer e mostrar o erro.

Prioridade: Média.

RF. 22: Relatório de controle de animais

Descrição: O sistema deve disponibilizar relatórios que mostrem ao usuário quanto animais existem no rebanho, qual a quantidade de fêmeas e machos, qual a faixa etária dos animais, quando fêmeas deve mostrar se estão prenhas e qual estagio estão.

Entrada: código do animal, período de nascimento, raça, pelagem, sexo.

Processo: O sistema deve pegar os dados de entrada e filtrar os dados que existem no banco de dados referentes ao animais cadastrados e retornar esses dados ao usuário em formato de relatório.

Saída: código do animal, nome, data de nascimento, raça, pelagem, sexo, pai, mãe.

Prioridade: Alta.

RF. 23: Relatório de produção

Descrição: o sistema deve disponibilizar relatórios que mostrem ao usuário a quantidade de nascimentos por período.

Entrada: código do animal inicial e final, período de nascimento inicial e final, raça, pelagem, sexo

Processo: O sistema deve pegar os dados de entrada e filtrar os dados que existem no banco de dados referentes a produção de animais e retornar esses dados ao usuário em formato de relatório.

Saída: código do animal, nome, quantidade de filhos, quantidade de cruzamentos

Prioridade: Media.

RF. 24: Relatório de reprodução

Descrição: o sistema deve disponibilizar relatórios que mostrem ao usuário quais animais são mais produtivos e menos produtivos, qual a taxa de cobertura por reprodutor, quais os índices de reprodução assistida, livre e de inseminação.

Entrada: código do animal, período de reprodução inicial e final, raça, pelagem, sexo.

Processo: O sistema deve pegar os dados de entrada e filtrar os dados que existem no banco de dados referentes a reprodução de animais e retornar esses dados ao usuário em formato de relatório.

Saída: código do animal, nome, nascimento, dados dos filhos

Prioridade: Media.

RF. 25: Relatório de perdas

Descrição: O sistema deve disponibilizar relatórios que mostrem ao usuário quais as perdas sofridas, tanto por morte quanto por desaparecimento de animais.

Entrada: código do animal, período da perda inicial e final, motivo da perda.

Processo: O sistema deve pegar os dados de entrada e filtrar os dados que existem no banco de dados referentes as perdas de animais e retornar esses dados ao usuário em formato de relatório.

Saída: código do animal, nome, tipo de perda, motivo da perda, data da perda

Prioridade: Media.

RF. 26: Relatório de comercio

Descrição: O sistema deve disponibilizar relatórios que mostrem ao usuário as compras e vendas efetivas.

Entrada: código, código do comprador/vendedor, código do animal, pendencia da parcela (quitada, pendente, ambas), período de vencimento inicial e final, período de cadastro inicial e final, tipo de relatório (analítico, sintético), situação da parcela (Em dia, atrasada, ambas).

Processo: O sistema deve pegar os dados de entrada e filtrar os dados que existem no banco de dados referentes a comercialização de animais e retornar esses dados ao usuário em formato de relatório.

Saída: código da compra/venda, data da compra, comprador, número da parcelas, data de vencimento, valor da parcela, juros, desconto, situação.

Prioridade: Alta.

RF. 27: Relatório de lactação

Descrição: O sistema deve disponibilizar ao usuário relatórios referentes a produção de leite.

Entrada: código do animal, período de ordenha inicial e final.

Processo: O sistema deve pegar os dados de entrada e filtrar os dados que existem no banco de dados referentes a produção de leite e retornar esses dados ao usuário em formato de relatório.

Saída: código do animal, período da lactação, quantidade de litros, porcentagem de gordura.

Prioridade: Media.

3.2.2 *Requisitos não funcionais*

RNF. 1 Facilidade de Uso

Descrição: O usuário deve ter facilidade na utilização do serviço.

RNF. 2 Acesso a informações

Descrição: O usuário deve ser capaz de acessar os dados do sistema através da internet, não necessitando a instalação de nenhum software.

RNF. 3 Segurança

Descrição: O sistema não deve permitir que usuário não cadastrados tenham acesso aos dados.

3.2.3 Planejamento

Por falta de planejamento no início do projeto, foram encontradas diversas dificuldades durante o desenvolvimento do software, as principais dificuldades encontradas foram:

- Falta de delimitação do escopo do projeto, isso fez com que o projeto ficasse maior do que o esperado;
- Em consequência do tamanho do projeto o cronograma proposto acabou por se tornar inviável, já que o tempo necessário para o desenvolvimento do projeto era maior do que o tempo proposto.
- Outra dificuldade encontrada foi referente à linguagem escolhida para o desenvolvimento do sistema, devido ao vago conhecimento do desenvolvedor.

Para que os problemas fossem corrigidos foi necessário uma reelaboração do projeto, isso fez com que o início do desenvolvimento atrasasse. Porém evitou outros possíveis atrasos por causa do retrabalho durante o seu desenvolvimento.

Uma das medidas tomadas para o planejamento do sistema foi criar mapas mentais que permitiriam uma melhor visualização do escopo do projeto. Esses mapas mentais foram criados com o intuito de auxiliar na visualização detalhada de cada fase do projeto e por consequência a visualização do projeto por completo.

A Figura 6 é o mapa mental criado para a visualização de cada fase do projeto e suas respectivas tarefas, como pode ser visto, o projeto foi dividido em planejamento, controle e atividades.

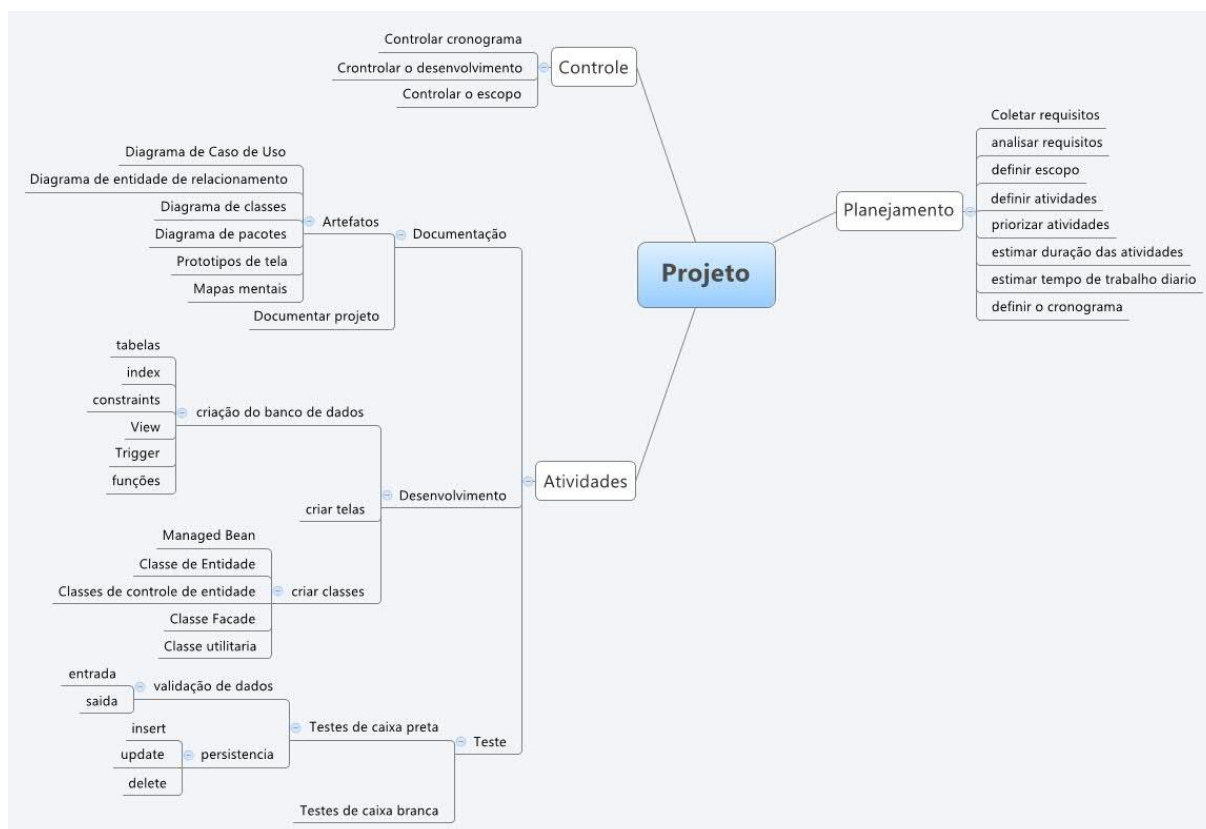


Figura 6 - Tarefas do projeto.

Fonte: Autoria própria.

Na fase de planejamento foram coletados e analisados os requisitos do sistema, com isso pode se definir o escopo do projeto, as atividades a serem executadas e a definição do cronograma do projeto.

As atividades foram divididas em documentação, desenvolvimento, e testes do sistema, na fase de documentação foram criados os diagramas de Caso de Uso, Diagrama de Entidade de relacionamento, Diagrama de Classes, Diagrama de Pacotes, Protótipo de Tela e mapas mentais, o resultado dessa fase pode ser visto no Capítulo 3.3 desse trabalho.

A fase de desenvolvimento é responsável pela criação do sistema funcional, nessa fase foram criados o banco de dados, as classes e as Telas do sistema, embora a fase de teste esteja separada da fase de desenvolvimento, os testes foram executados em paralelo com o desenvolvimento, os detalhes dessa fase podem ser vistos no capítulo 3.4 desse trabalho.

A parte de controle é referente ao controle do projeto, para visualizar e prevenir possíveis problemas no decorrer do desenvolvimento.

Em consequência do novo planejamento, alguns dos requisitos foram retirados do projeto, esses requisitos não afetam o objetivo do sistema que é o controle de animais. A Tabela 2 é referente à lista das funcionalidades retiradas do sistema.

Tabela 2 - Requisitos removidos do planejamento.

Requisito	Referencia
Manter de lactação	RF. 16
Manter dieta	RF. 17
Controle de gastos alimentícios	RF. 18
Controle de gastos médicos	RF. 29
Relatório de lactação	RF. 37
Relatório de produção	RF. 23
Relatório estatístico de reprodução	RF. 24

Autor: Aatoria Própria.

3.3 Modelagem

A primeira fase do projeto foi a definição dos artefatos de modelagem do sistema, nessa fase foram criados os diagramas de Caso de Uso, diagrama de Entidade Relacionamento, diagrama de Classes, diagrama de Pacotes e os Protótipos de Tela.

O primeiro diagrama a ser criado foi o diagrama de caso de uso, em que as necessidades do usuário são representadas de forma gráfica. Na Figura 7 pode ser visto o diagrama de caso de uso gerado, onde o usuário é representado pelo ator e as funcionalidades do sistema são descritas através dos balões (caso de uso).

O diagrama de caso de uso é importante pois fornece um meio de comunicação claro entre o cliente e o analista, através dele são descritos os requisitos funcionais do sistema, fornecendo uma descrição clara e consistente do que o sistema deve fazer, isso faz com que ele seja utilizado ao longo do projeto para documentar e servir de base para a modelagem do projeto.

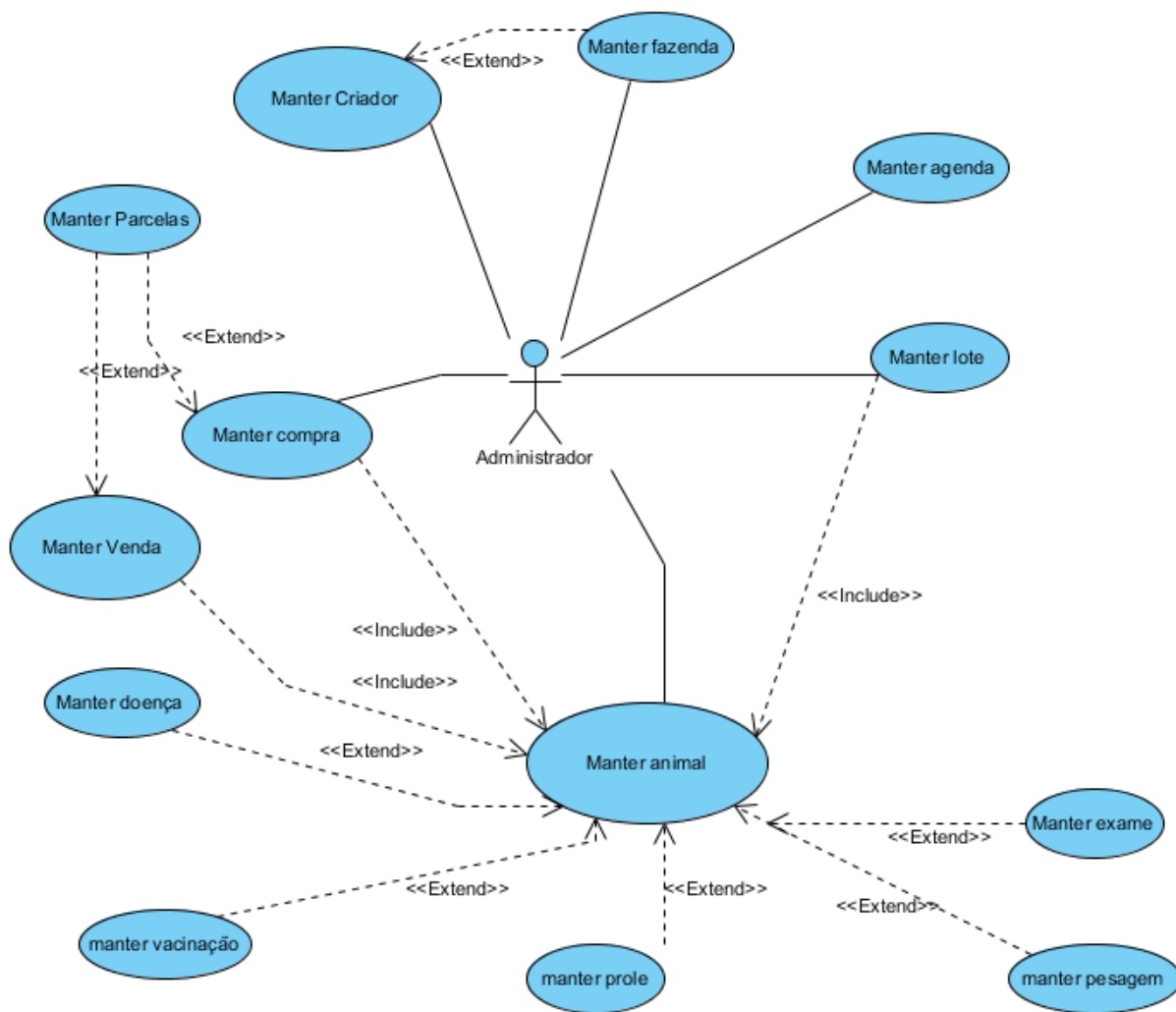


Figura 7 - Diagrama de Caso de Uso.
Fonte: Autoria própria.

Como pode ser visto na Figura 7 o usuário é ligado diretamente as funcionalidades de manter animal, manter fazenda, manter criadores, manter agenda, manter compra e manter venda. Algumas dessas funcionalidades podem fazer com que obrigatoriamente o usuário execute outras funções, um exemplo é o manter lote, onde obrigatoriamente ao registrar um lote de animais o proprietário automaticamente mantém o registro de animais individuais, esse tipo de ação é representado pela flecha pontilhada *include*.

Em outros casos o usuário tem a opção de executar outras ações, como no caso de manter compra, em que o usuário, pode ou não optar por fazer uma compra parcelada, o que faria que a funcionalidade de manter parcelas fosse ou não executada, essa ação é representada pela flecha pontilhada *extend*.

Após a criação do diagrama de caso de uso, pode se iniciar a fase de modelagem de dados, onde foi gerado o diagrama de entidade-relacionamento (DER), pela ferramenta MySQL WorkBank, esse diagrama é responsável pela visualização dos relacionamentos entre as tabelas do banco de dados

Fazer com que o DER atendesse as necessidades do sistema e seguisse as regras de normalização de banco de dados foi um grande desafio encontrado na elaboração desse projeto.

A normalização é a forma de organizar a estrutura de um banco de dados, para tal, existem regras de normalização. Cada regra é chamada de “forma normalizada”, se o banco de dados se encontra de acordo com a primeira regra, ele é considerado na “primeira forma normalizada”, sendo que a de mais alto nível é a “terceira forma normalizada”.

Abaixo encontram-se as regras de normalização (MICROSOFT, 2004):

- Primeira forma normalizada:
 - Eliminar repetições em tabelas individuais;
 - Criação de tabelas separadas para cada conjunto de dados;
 - Identificar cada conjunto de dados com uma chave primaria.
- Segunda forma normalizada:
 - Criar tabelas separadas para conjuntos de valores que se aplicam a vários registros.
 - Relacionar as tabelas relacionadas com chaves estrangeiras.
- Terceira forma normalizada:
 - Eliminar os campos que não dependem de chave.

Devido a isso o projeto sofreu atraso. As vinte horas programadas para a elaboração do DER não foram cumpridas. Foram necessárias 40 horas de trabalho para a sua finalização. Apesar das dificuldades encontradas e do atraso nessa parte do projeto, essa normalização é importante pois elimina o retrabalho na modificação do banco de dados, facilitando assim a sua ampliação.

O DER pode ser visto na Figura 8, na Figura 9, Figura 10, Figura 11, Figura 12 e Figura 13. Ele foi dividido para facilitar a sua visualização. Nele estão as entidades juntamente com seus atributos e os relacionamentos existentes entre elas e suas respectivas cardinalidades.

Nas figuras as linhas contínuas representam os relacionamentos em que as chaves estrangeiras se tornam chaves primárias na tabela em que são referenciadas.

As linhas pontilhadas representam as chaves estrangeiras que apenas referenciam as tabelas. Os campos das entidades que estão marcados com os losangos azuis representam os atributos que não podem ter seu valor nulo. Os campos marcados com losangos brancos aceitam valores nulos. Os campos marcados com um losango rosa vazado é uma chave estrangeira que aceita valores nulos. As chaves primárias de cada entidade são marcadas pelo desenho de uma chave amarela.

A Figura 8, é a parte do diagrama onde fica a estrutura dos dados do proprietário, como pode ser visto na figura o usuário do sistema também pode ser cadastrado como um proprietário.

Cada proprietário pode ser vinculado a uma ou várias fazendas,

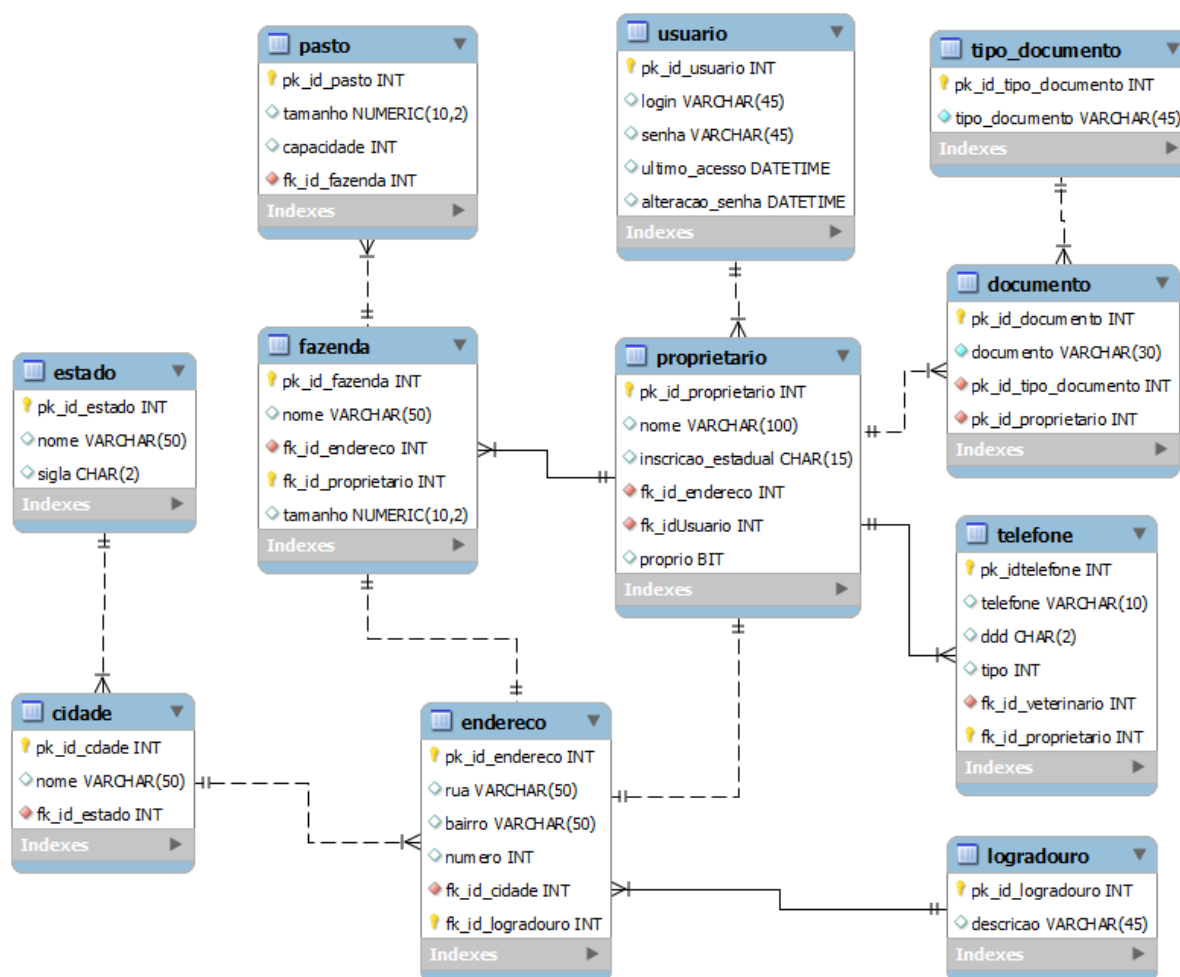


Figura 8 - Diagrama de entidade-relacionamento proprietário.

Fonte: Autoria própria.

A Figura 9, é a parte do diagrama onde fica a estrutura do dados do animal e mostra como ele é ligado ao seu proprietário.

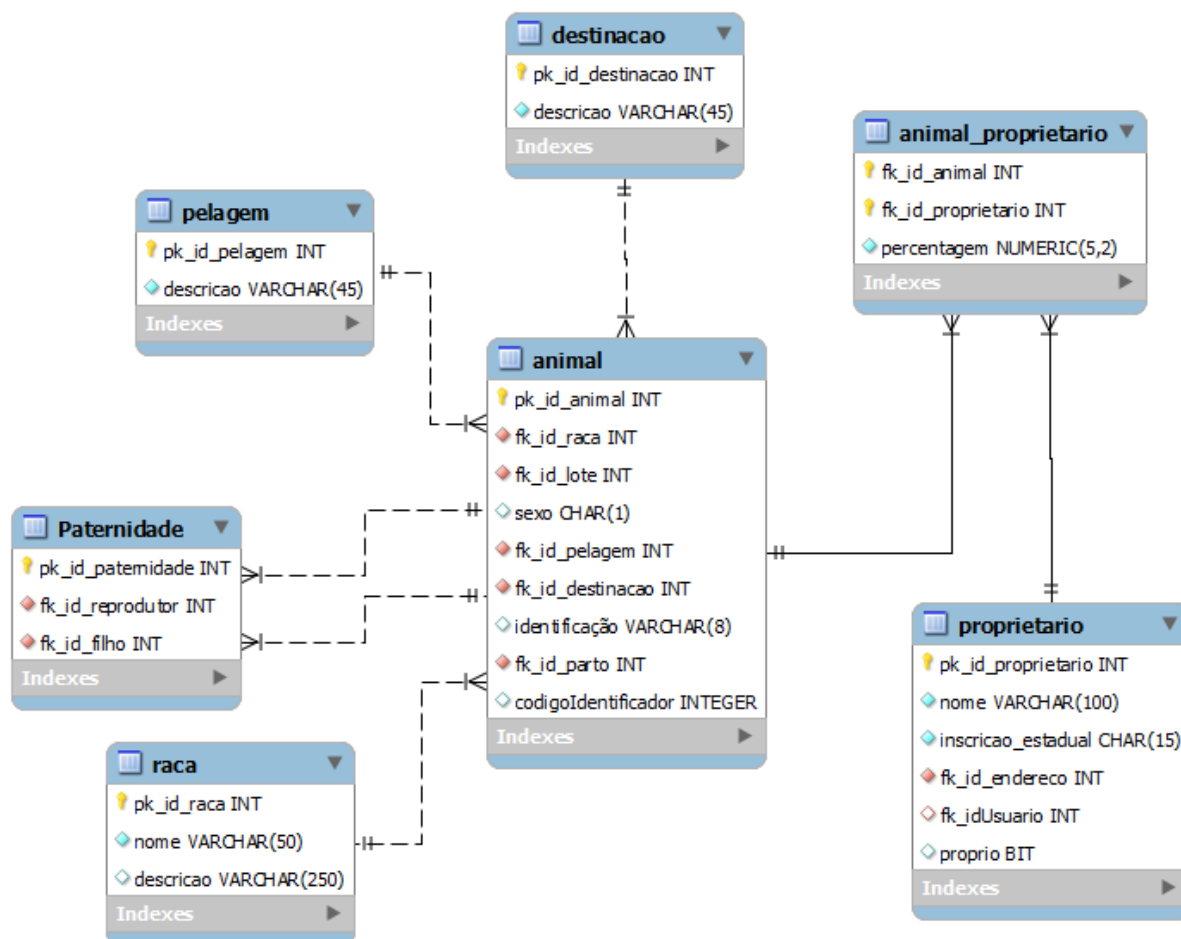


Figura 9 - Diagrama entidade-relacionamento animal.

Fonte: Autoria própria.

A Figura 10 é a parte do diagrama que possui a estrutura dos dados dos eventos sanitários do animal.

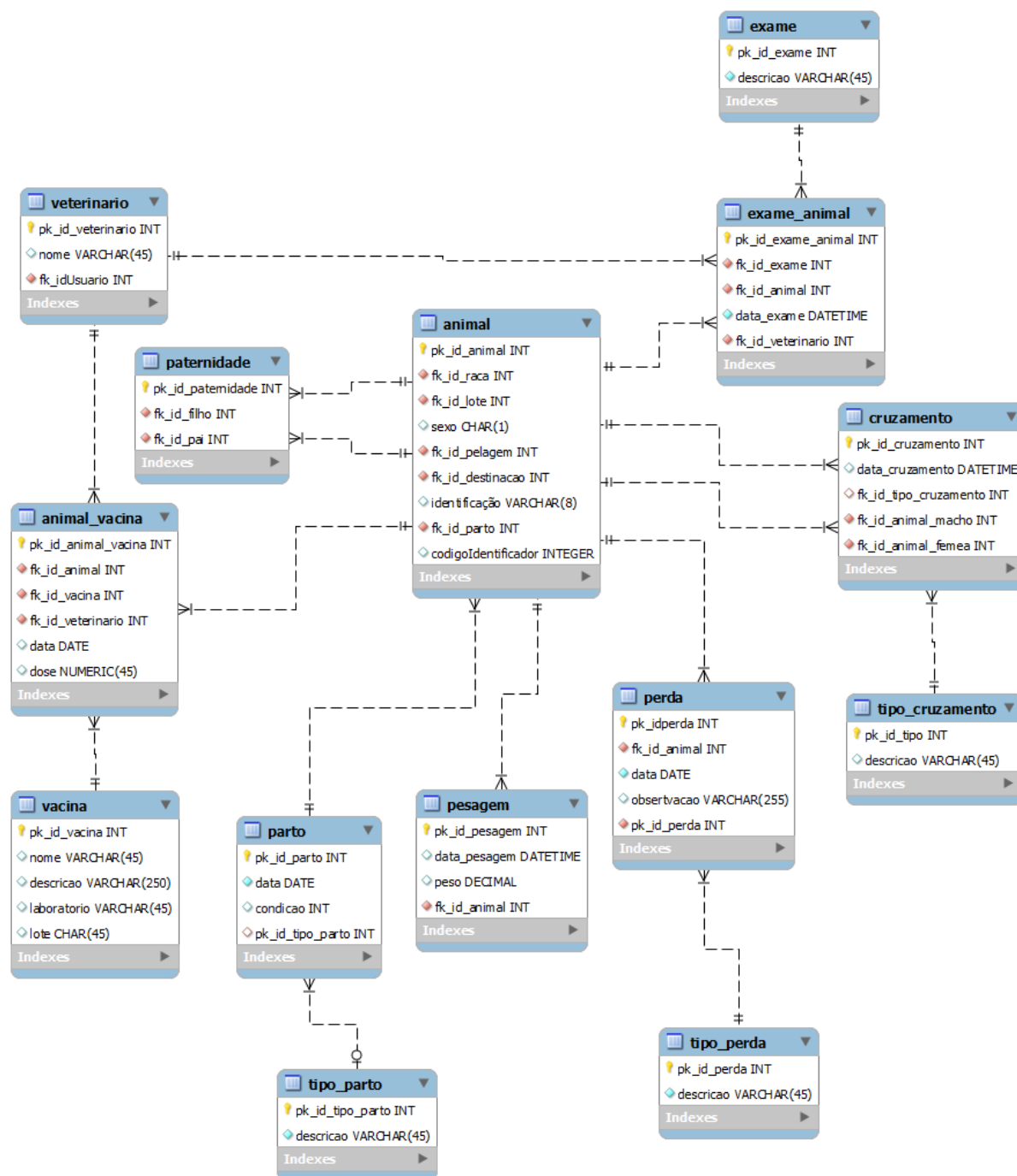


Figura 10 - Diagrama entidade-relacionamento evento sanitário.

Fonte: Autoria própria.

A Figura 11 é a parte do diagrama de entidade-relacionamento onde fica a estrutura de dados da comercialização de animais

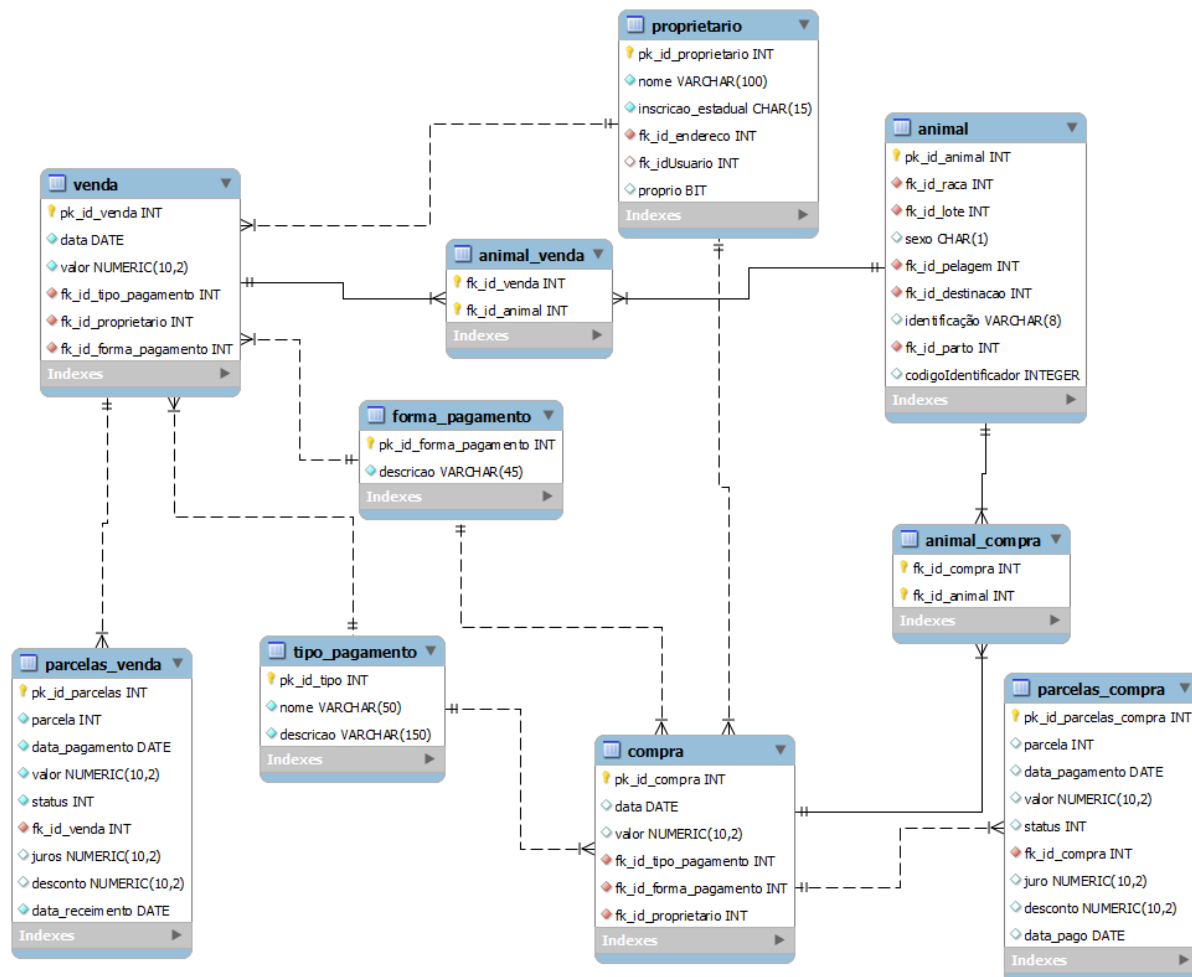


Figura 11 - Diagrama entidade-relacionamento comercialização.
Fonte: Autoria própria.

A Figura 12 é a estrutura da tabela agenda, onde ficam armazenadas as informações de eventos, essa tabela não possui relacionamento com nenhuma outra tabela do banco de dados.

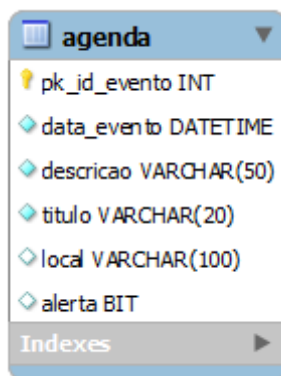


Figura 12 - Diagrama entidade-relacional agenda
Fonte: Autoria própria.

Existe também a tabela “Menu” representada pela Figura 13, essa tabela não está ligada a nenhuma outra tabela do banco de dados, ela somente existe para a criação do menu da aplicação, nela são armazenadas as informações de título do menu, pagina que o menu acessa e ícone do menu.

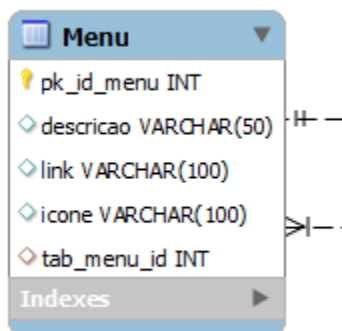


Figura 13 - Diagrama entidade-relacionamento menu.
Fonte: Autoria própria.

Os protótipos de tela apresentados a seguir foram gerados na ferramenta Evolus Pencil. Os protótipos de tela tem o objetivo de criar layouts de tela, telas essas que foram implementadas no sistema. Esses layouts são criados para visualizar a aparência e o funcionamento do sistema, isso facilita a apresentação do sistema para o cliente.

A Figura 14 é o protótipo da tela de cadastro e edição de Animais, como pode ser visto ela é dividida em abas, isso facilita o cadastro pelo usuário, pois cada aba

possui dados específicos referentes ao animal que está sendo cadastrado. A aba em destaque é a “Animal”, essa aba contém os principais dados do animal, esses dados são indispensáveis para se armazenar as informações a respeito do animal.

Figura 14 - Protótipo de Tela: Cadastro de Animal.
Fonte: Autoria própria.

A Figura 15 é o protótipo da tela de cadastro de animal, porém diferente da figura anterior ela mostra a aba de “Vacina”, nessa tela os dados referentes a vacinação do animal são inseridos. O usuário pode inserir uma quantidade infinita de vacinações, essas vacinações são inseridas em uma tabela, onde o usuário pode visualizar todas as vacinas inseridas.

Vacina	Laboratorio	Lote	Data Vacinação	Dose Aplicada	Veterinario
Content 1	Content 2				
Content 3	Content 4				

Figura 15 - Protótipo de tela: Cadastro de animal, aba Vacinação.
Fonte: Autoria própria.

A Figura 16 é o protótipo de tela do relatório de vendas, essa tela permite que o usuário filtre as buscas da maneira que desejar, desde buscas mais genéricas até buscas mais específicas.

O protótipo de tela do relatório de vendas apresenta uma barra de navegação superior com os itens 'Inicio', 'Cadastro v', 'Comercio v' e 'Relatorios v'. Abaixo, o título 'Relatorio Vendas' precede um formulário com os seguintes elementos:

- Campos de texto para: Codigo Venda, Codigo Animal, Nascimento Ini, Vencimento Ini, Codigo comprador, Nascimento Fim, e Vencimento Fim.
- Controles de seleção:
 - 'Situação Parcela' com opções: Quitada, Pendente, Ambas.
 - 'Tipo' com duas opções: Em dia, Atrasada, Ambas; e Sintetico, Analitico.
- Botão 'Gerar Relatório' localizado no canto inferior esquerdo.

Figura 16 - Protótipo de tela: Cadastro de Venda.
Fonte: Autoria própria.

Como pode ser visto nas figuras anteriores dos protótipos de tela, todas as telas possuem um menu de navegação, esse menu é dividido em cadastro, comércio e relatórios.

Outro diagrama criado foi o diagrama de classes, esse diagrama é responsável pela estrutura do código fonte do software, através desse diagrama é possível visualizar como os objetos irão se comunicar. Assim como o diagrama de caso de uso, esse diagrama foi gerado pela ferramenta Visual Paradigma.

O diagrama de classes foi criado com a seguinte estrutura:

- Classe Managed Bean: essa classe possui a lógica de aplicação da página que o usuário está utilizando, ela possui o código Java do lado do servidor que é responsável por responder as requisições do usuário.
- Classe de negócio: essa classe possui a regra de negócio, nelas são controlados e validados os dados que serão inseridos e alterados das classes de entidade.
- Classe de controle de entidade: essa classe controla o modo como os dados são inseridos, excluídos ou buscados no banco de dados.
- Classes de entidade: essa classe faz o Mapeamento objeto relacional do banco de dados.

- Utilitárias: Essa classe possui métodos que auxiliam as outras classes.

A Figura 17 representa a estrutura citada acima, nela estão representadas as classes utilizadas para o cadastro do animal:

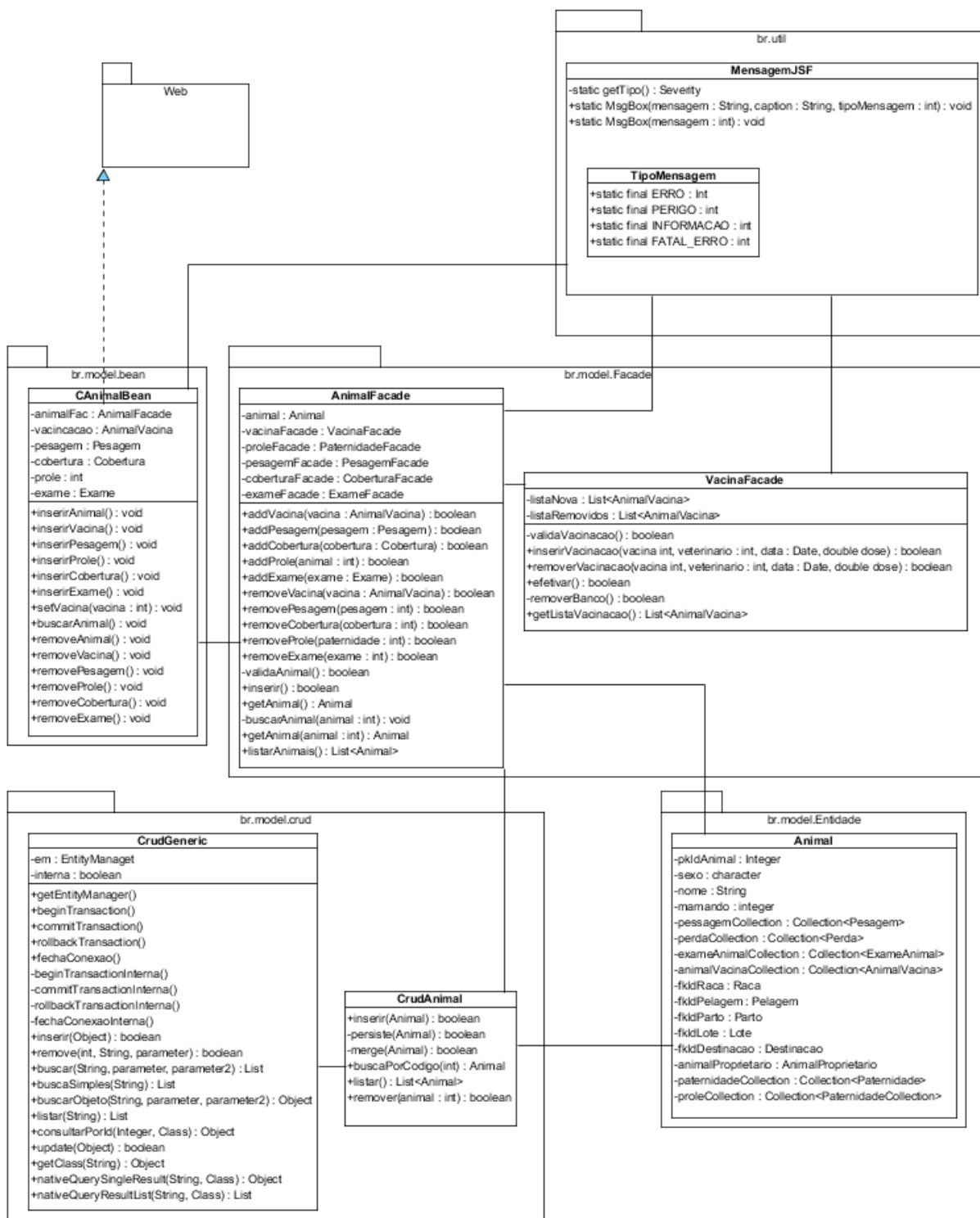


Figura 17 - Diagrama de Classes.
 Autor: Autoria própria.

Para haver uma melhor organização do código, os objetos foram separados em pacotes distintos, cada um com a coleção de classes referentes a ele, esses pacotes podem ser vistos na Figura 18:

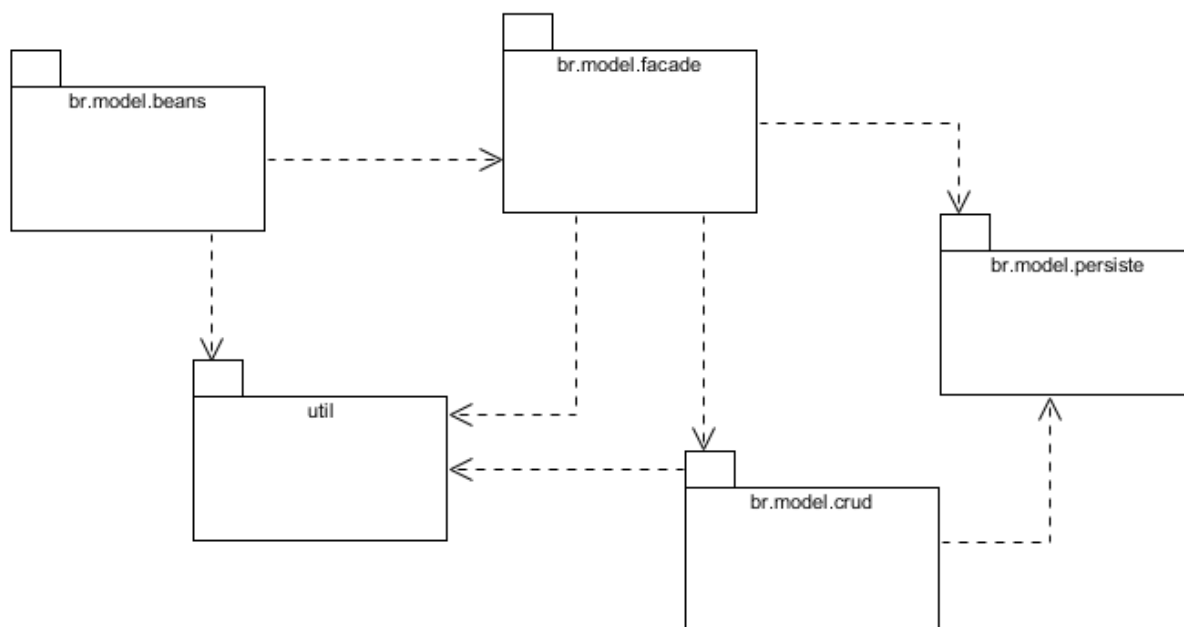


Figura 18 - Diagrama de pacotes.
Autor: Autoria própria.

- As classes Manager Bean foram alocadas no pacote `br.model.Beans`;
- As classes de negócios foram alocadas no pacote `br.model.facade`;
- As classes de controle de entidade foram alocadas no pacote `br.model.crud`;
- As classes de entidade foram alocadas no pacote `br.model.persiste`;
- As classes utilitárias foram alocadas no pacote `br.util`;

3.4 Implementação

No primeiro Sprint iniciou-se da criação do sistema com a construção do banco de dados, o ferramenta de Banco de Dados(SGDB) utilizado foi o PgAdmin 3.

O primeiro passo para a construção do banco de dados foi a criação das entidades e os seus relacionamentos, sua construção foi feita seguindo o modelo de entidade e relacionamento mostrado anteriormente nesse capítulo.

O passo seguinte na construção do banco de dados, foi a criação dos índices e *constraints*, para auxiliar nas buscas e nas validação dos dados que são inseridos através do sistema. As *procedures*, *triggers* e *views* foram sendo criadas no decorrer do projeto, conforme a sua necessidade.

Após a construção do banco de dados teve início o segundo *Sprint*, com a codificação do sistema, a mesma foi feita na IDE NetBeans 7.2, utilizando a linguagem Java e o Framework JavaServer Faces. As principais bibliotecas utilizadas no desenvolvimento foram:

- EclipseLink (JPA 2.0), biblioteca responsável pelo mapeamento objeto relacional do banco de dados;
- PrimeFaces 3.4.2, biblioteca de componentes;
- JasperReport 5.2, biblioteca responsável pela geração de relatórios;
- Driver JDBC4 PostgreSQL 9.1, drive utilizado para fazer a conexão entre a aplicação e o banco de dados PostgreSQL.

Esse Sprint foi responsável pela criação de uma conexão entre a aplicação e o banco de dados, para isso o primeiro passo foi estabelecer uma conexão JDBC da aplicação com o banco de dados, as configurações dessa conexão são armazenadas no Data Source.

O segundo passo foi a criação da classe *CrudGeneric*, essa classe possui todos os métodos necessários para que a aplicação acesse os dados do banco de dados. Esses métodos por sua vez são genéricos, o que faz com que qualquer entidade mapeada possa ser persistida ou buscada através dessa classe.

O terceiro passo foi o mapeamento das entidades do banco de dados, esse mapeamento foi feito automaticamente pela IDE. Algumas classes do sistema são criadas por configurações padrões da IDE, por tanto, precisaram sofrer algumas alterações para se adequarem às funcionalidades do sistema, algumas dessas alterações são:

- O uso da anotação `@SequenceGenerator`, afim de melhorar a geração da chave primária da entidade;
- Alteração da anotação do tipo de relacionamento entre as entidades, pois a geração automática somente cria as anotações `@ManyToOne` (muitos para um) e `@OneToMany` (um para muitos), e em muitos casos é necessário a anotação `@OneToOne` (um para um).

O segundo Sprint baseou-se em popular o banco de dados com os dados utilizados no cadastro do sistema, dados como estados, cidades, logradouros, raças, pelagens. Esses dados foram cadastrados antes da implementação do sistema, para que o mesmo pudesse ser testado conforme a sua construção.

Para a criação das telas foi utilizado o *framework JavaServer Faces* utilizando a biblioteca de componentes *PrimeFaces*, essa biblioteca possui diversos componentes que fazem com que a criação da interface do sistema fique mais simples, pois não necessita que os componentes sejam criados do zero, componentes como calendário, tabelas, formulários entre outros componentes são adicionados e renderizados na tela em formato html (CAELUM,2014).

Para o bloqueio de caracteres inválidos, mudança de componentes em tempo de execução da página, em validações e para executar comandos de acordo com a ação executadas na tela, foi utilizado o *JavaScript*.

A nomenclatura utilizada para as telas foi utilizada da seguinte forma:

- Para as telas que são feitos os cadastros é utilizada a consoante “C” seguido do nome da entidade principal a ser persistida no banco, por exemplo, a tela de cadastro de animal que ficou com o nome “CAnimal.xhtml”.
- Para telas que são retirados os relatórios é utilizada a consoante “R” seguida do nome simplificado do relatório, por exemplo, a tela de relatório de comercialização que ficou com o nome “RComercializacao.xhtml”.

Para cada tela existe uma classe do tipo Managed Bean, essa classe é responsável por fazer a intermediação entre o JSF e o código da aplicação, ele tem a função de executar eventos, processa-los e delegar para a camada de negócio, as classes Facade. Através do Managed Bean é possível especificar o tipo de escopo utilizado para a tela através de anotações.

O nome dessa classe é o mesmo que o da tela que ele é responsável, por exemplo a tela “CAnimal.xhtml” que possui o Manager Bean “CAnimal.java”.

As classes *Manager Bean* delegam tarefas para as classes facades, que são classes responsáveis por tratar as informações e por controlar as classes de persistência, essas classes tem a função de simplificar as dependências entre os objetos, por exemplo, a classe “CAnimal.java” conhece apenas o objeto “AnimalFacade.java”, esse por sua vez conhece a classe de persistência “Animal.java”

e as classes Facades responsáveis pelas classes “Pesagem”, “Paternidade”, “Cobertura” entre outras e apenas as classes Facade possuem conhecimento a respeito da regra de negócio da aplicação, ficando a cargo delas fazer as validações e estabelecer o modo como as informações serão persistidas no banco de dados.

O nome das classes Facade é composto pelo nome da entidade principal seguido da palavra “Facade”, como mostrado no exemplo acima da classe “AnimalFacade.java”.

No terceiro Sprint foram criadas as telas de cadastro de usuário, de acesso ao sistema e de recuperação de senha.

Para segurança do usuário a senha inserida no banco de dados é criptografada com o algoritmo de criptografia MD5, um algoritmo hash de 128 bits unidirecional. Por ser unidirecional ele não pode ser descriptografado, fazendo com que a senha de origem não possa ser recuperada. Para a recuperação do acesso do usuário pelo sistema de recuperação de senha, é necessário a geração de outra senha para o usuário.

Para envio de e-mail com a senha nova foi utilizada as bibliotecas mail.jar e activation.jar. Com a utilização dessas bibliotecas foi construída uma classe responsável pelo envio de e-mail, a classe “Email.java”.

No quarto Sprint foi criada a tela de cadastro de animal, foi criado também um menu dinâmico, para facilitar na criação de telas e padronizar o menu. Esse menu é carregado com os dados da tabela do banco de dados “Menu”, onde a mesma tem as informações quanto ao nome, o ícone do item do menu e o link para a página a ser acessada.

No quinto Sprint foi criado o cadastro de vacinação, reprodução, pesagem e exame. As telas para esses cadastros foram criadas como abas na tela de cadastro de animal, sendo que todos os dados que são inseridos nessa tela, independentemente da aba em que estão são salvos na mesma ação.

Os dados referentes a vacinações, reproduções, pesagens, tratamentos e exames, são armazenados em listas conforme são inseridos e só são salvos quando os dados do animal também são salvos.

A classe CrudAnimal é responsável por salvar todos esses dados, essa classe verifica através da existência da chave primaria se o objeto já existe no banco de dados, caso exista ela executa o *update* caso não exista ela executa o *insert*. Caso

ocorra um erro ao salvar algum dado é dado *rollback* na transação, fazendo que o animal não seja salvo com dados incompletos.

No sexto Sprint foi criada a tela de produtor e de cadastro de fazenda, os dois cadastros forma criados na mesma tela, sendo separados por abas. Como o proprietário pode ser pessoa jurídica ou pessoa física foi criado um *JavaScript* para alterar a máscara e a descrição do campo de CPF/CNPJ de acordo com o tipo de pessoa que o usuário escolher para o proprietário. Para realizar as validações da documentação do produtor foi criada a classe documentação, que possui validações para CPF, CNPJ e inscrição estadual.

Os campos Estado e Cidade estão parametrizados, possuindo o registro de todos os estados e cidades brasileiras, para facilitar a buscar pela cidade do criador, foi estabelecido que o campo de cidade só fosse habilitado e carregado após a escolha do estado.

Como no caso anterior do cadastro de animal, a ação de salvar salva todos os dados independente da aba em que esteja, no caso do produtor e da fazenda a classe responsável por salvar os dados é a classe “CrudProprietario”.

No sétimo Sprint foi criada a Tela de Compra e Venda. Assim como as outras telas de cadastro, essas foram criadas em abas, onde cada aba armazena um tipo de informação referente à venda ou compra de animais.

No oitavo Sprint foi criada a tela inicial do sistema, essa tela foi criada utilizando o componente de layout da biblioteca PrimeFaces, o que permitiu que a tela fosse dividida, podendo comportar diversas informações.

Para a criação dos layouts dos relatórios foi utilizada a ferramenta IReport. Para cada relatório disponibilizado pelo sistema foi gerado um layout, esse layout fica em um arquivo com extensão .jasper, gerado pela ferramenta. Os dados que serão mostrados no relatório são disponibilizados por procedures, que recebem os dados dos filtros da tela e retornam os dados filtrados. A comunicação dos dados retornados pela procedure e o layout do relatório fica a cargo da classe “Relatorio”, essa classe também é responsável pela disponibilização do relatório em forma de PDF ao usuário.

O nono e o décimo Sprint tiveram as funcionalidades de relatório implementadas, no nono Sprint foi criada a tela de relatório de animais, e no decimo as telas de relatórios de comercialização e de perda de animais. Em ambos os casos foram criadas procedures distintas para receber os filtros e retornar os dados requisitados para a construção dos relatórios.

Para que fosse possível o envio dos dados para a classe “Relatorio” para a geração dos relatórios, os dados tiveram que ser transformados em uma lista de objetos, onde essa lista seria lida pela classe e assim os dados seriam alocados do layout criado o IReport.

Ao fim dos Sprints foram realizados testes integrados, a fim de verificar se algum erro passou a existir após os primeiros testes em tela.

Para esses testes não foi utilizado nenhum padrão de documento, o sistema foi sendo testado a medida em que os dados foram sendo cadastrados no sistema.

A Tabela 3 e Tabela 4 possuem as funcionalidades geradas em cada Sprint de desenvolvimento.

Tabela 3 - Sprints

Sprint	Atividades	Prioridade
1º Sprint	Banco de dados	Alta
	Conexão do Sistema com o banco de dados	Alta
2º Sprint	Popular o banco de dados	Alta
3º Sprint	Cadastro de usuário	Alta
	Acesso ao sistema	Alta
	Recuperação de senha	Alta
4º Sprint	Cadastro de animal	Alta
	Cadastro de lote de animais	Media
5º Sprint	Cadastro vacinação	Media
	Cadastro de reprodução	Media
	Cadastro de pesagem	Media
	Cadastro de perda	Media
	Cadastro de exames	Media
6º Sprint	Cadastro de Produtor	Alta
	Cadastro de Propriedades	Media
7º Sprint	Cadastro de compra	Alta
	Cadastro de venda	Alta
8º Sprint	Cadastro de eventos	Media
	Tela Inicial do Sistema	Media
9º Sprint	Relatório de controle de animais	Alta

Fonte: Autoria própria.

Tabela 4 - Continuação da Tabela 3 - Sprints

Sprint	Atividades	Prioridade
10º Sprint	Relatório de perdas de animal	Media
	Relatório de comercialização	Alta

Fonte: Autoria própria.

3.5 APRESENTAÇÃO DO SISTEMA

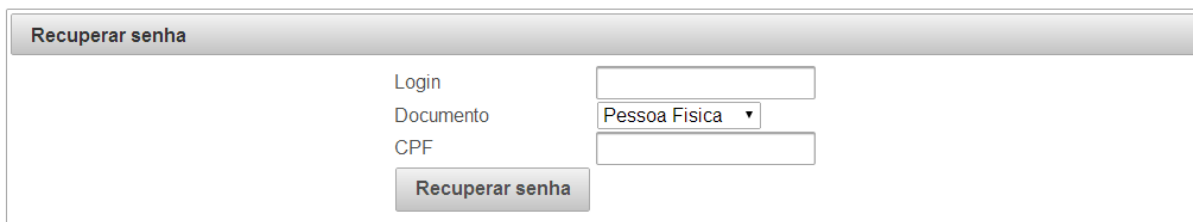
Essa sessão se destina a apresentação do sistema construído durante o projeto.

A Figura 19 é a tela de login do sistema, através dessa tela o usuário pode ter acesso ao sistema ou acessar a tela de recuperação de senha Figura 20. Para ter acesso ao sistema o usuário deve já estar cadastrado e no momento de logar deve digitar usuário e senha correspondentes.

Figura 19 - Interface de login.

Fonte: Autoria própria.

A Figura 20 é a tela de recuperação de senha, através dessa tela o usuário pode solicitar uma nova senha para acessar o sistema, caso tenha a esquecido. Para solicitar a nova senha o usuário deve informar o usuário de login e o documento cadastrado para aquele login, se no caso for uma pessoa física deve digitar o CPF no caso de pessoa jurídica deve digitar o CNPJ. Se os dados estiverem corretos um e-mail será enviado com a nova senha e uma mensagem de aviso de envio de e-mail aparecerá para o usuário.



Recuperar senha

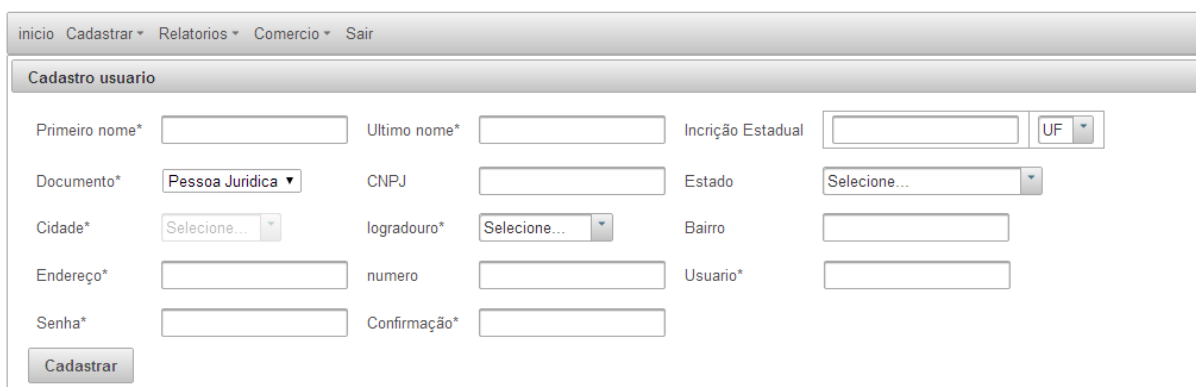
Login

Documento

CPF

Figura 20 - Interface de recuperação de senha.
Fonte: Autoria própria.

A Figura 21 é a tela de cadastro de novos usuários, através dessa tela o administrador do sistema pode cadastrar os usuários que terão acesso ao sistema. Se todos os dados forem preenchidos corretamente e o cadastro for bem sucedido, aparecerá uma mensagem informando que o cadastro foi concluído e já pode acessar o sistema.



inicio Cadastrar Relatorios Comercio Sair

Cadastro usuario

Primeiro nome* Ultimo nome* Incrição Estadual UF

Documento* CNPJ Estado

Cidade* logradouro* Bairro

Endereço* numero Usuario*

Senha* Confirmação*

Figura 21 - Interface de cadastro de usuário.
Fonte: Autoria própria.

A Figura 22 é a tela inicial ao acessar o sistema, através dela o usuário pode ter informações sobre a agenda de compromissos e informações sobre a pecuária. Nessa tela o usuário também pode adicionar eventos a agenda. E através do menu que está presente em todas as telas do sistema o usuário pode navegar por todas as telas do sistema.

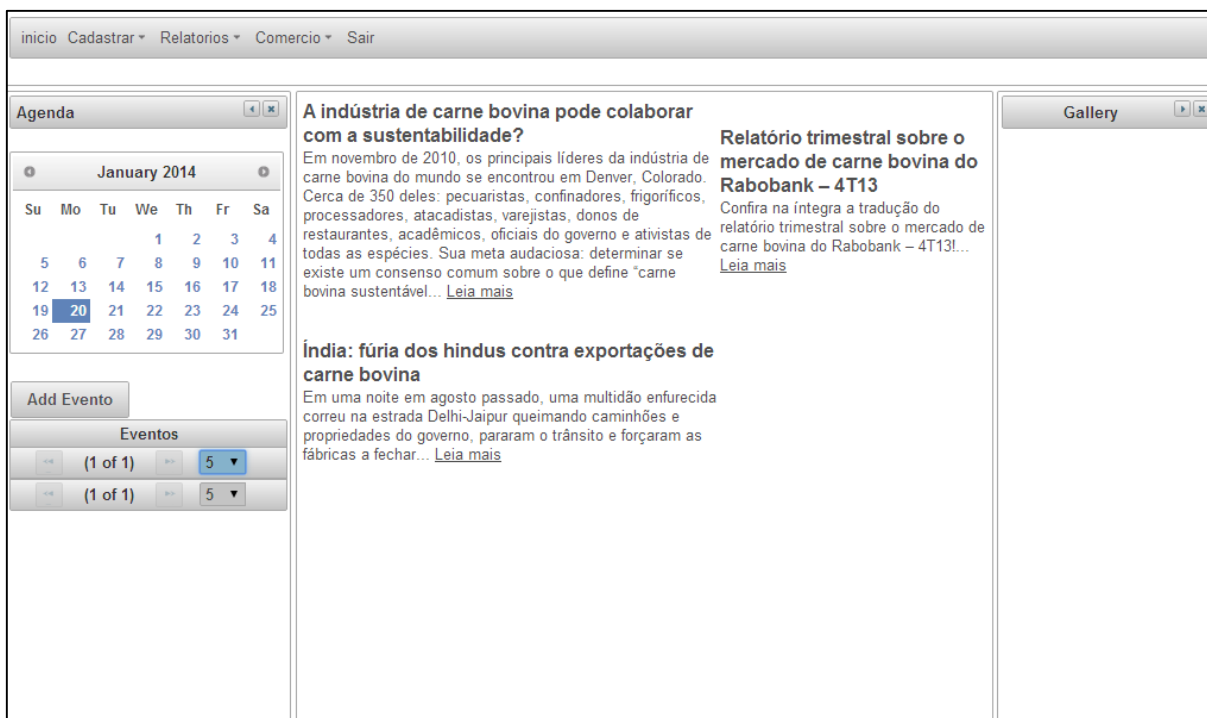


Figura 22 - Interface inicial do sistema.

Fonte: Autoria própria.

A Figura 23 é a tela de cadastro de animais, através dessa tela o usuário pode cadastrar, visualizar, editar e excluir os dados referentes aos animais, essa tela é dividida em seis abas, onde cada aba representa informações distintas dos animais. Como pode ser visto essa tela contém os dados principais dos animais, como código identificador, raça, pelagem, data de nascimento, sexo, entre outras informações, a tela apresenta também a lista de todos os animais cadastrados.

Cadastrar ▾ Relatórios ▾ início Comercio ▾

Cadastro de Animal

Animal | Pesagem | Vacina | Coberturas | Tratamento | Prole

Código: Brinco: Nome:

Sexo: Macho Femea Nascimento: Raça:

Pelagem: Lote: Destinação:

salvar

Lista de Animais

Código	Sexo	Pelagem	Raça	Nascimento	comandos
4		mista	Guzerá		<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
3		branca	Nelore	18/05/2012	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
12		branca	Brangus	18/05/2012	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
13		branca	Brangus	18/05/2012	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
18		branca	Nelore	01/11/3910	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
20		branca	Nelore	01/11/3910	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
24		branca	Nelore	01/11/3910	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
26		branca	Nelore	01/11/3910	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>

Figura 23 - Interface de cadastro de animal, aba Animal.

Fonte: Autoria própria.

A Figura 24 é a tela de cadastro de animais na aba de pesagem, nessa tela são cadastradas as pesagens que são feitas do animal, após o cadastro da pesagem ela pode ser visualizada na lista de pesagens, onde pode ser visto outras pesagens já realizadas com o animal. A partir dessa lista de pesagem é possível editar e excluir as pesagens.

início Cadastrar Relatórios Comercio Sair

Cadastro de Animal

Animal Pesagem Vacina Coberturas Exame Prole

Pesagem: 05/01/13 Peso: 647 add Pesagem

Lista de pesagens

Peso	Data da Pesagem	comandos
647.000	05/01/2013	Editar Excluir
600.000	18/12/2013	Editar Excluir
110.000	18/10/2013	Editar Excluir

salvar

Lista de Animais

Codigo	Sexo	Pelagem	Raça	Nascimento	comandos
61		preta	Bonsmara	13/12/2013	Editar Excluir
62		branca	Brangus	17/10/2013	Editar Excluir
65		mista	Guzerá	17/10/2013	Editar Excluir

Figura 24 - Interface de cadastro de animal, aba Pesagem.

Fonte: Autoria própria.

A Figura 25 é a tela de cadastro de animal na aba de vacina, através dessa tela é possível cadastrar as vacinações realizadas no animal, essa aba também traz a lista de todas as vacinações realizadas no animal, a partir dessa lista é possível editar e excluir a vacinação.

inicio Cadastrar Relatorios Comercio Sair

Cadastro de Animal

Animal Pesagem **Vacina** Coberturas Exame Prole

Vacina Selezione... Data Vacinação: Dose Veterinario Selezione... add Vacina

Lista de Vacinas Aplicadas

Vacina	Laboratorio	Lote	Data Vacinacao	Dose Aplicada	Veterinario	comandos
Poliguard	Vallée	bg212	20/10/2013	30	Vitor	Editar Excluir
Fertiguard Selenium Max	Vallée	vn939	14/11/2013	49	Reinaldo	Editar Excluir
Poli-Star	Vallée	kj923	01/01/2014	54	Reinaldo	Editar Excluir

salvar

Lista de Animais

Codigo	Sexo	Pelagem	Raça	Nascimento	comandos
61		preta	Bonsmara	13/12/2013	Editar Excluir
65		mista	Guzerá	17/10/2013	Editar Excluir
62		branca	Brangus	17/10/2011	Editar Excluir

Figura 25 - interface de cadastro de animal, aba vacina.

Fonte: Autoria própria.

A Figura 26 é a tela de cadastro de animal na aba de cobertura, nela são inseridos os animais que o animal que está sendo cadastrado cruzou, após o cadastro do cruzamento é possível visualiza-lo na lista de cruzamento, onde podem ser vistos outros cruzamentos realizados. A partir da lista também é possível editar e excluir os cruzamentos.

início Cadastrar Relatorios Comercio Sair

Cadastro de Animal

Animal Pesagem Vacina **Coberturas** Exame Prole

Codigo do Parceiro Data Cobertura Tipo Cruzamento Selezione...

Lista de Coberturas

Codigo	Nascimento	Tipo de Cruzamento	comandos
65	20/10/2013	Assistido	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
65	14/11/2013	Assistido	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
65	01/01/2014	Assistido	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>

Lista de Animais

Codigo	Sexo	Pelagem	Raça	Nascimento	comandos
61		preta	Bonsmara	13/12/2013	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
65		mista	Guzerá	17/10/2013	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
62		branca	Brangus	17/10/2011	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>

Figura 26 - Interface de cadastro de animal, aba cobertura.

Fonte: Aatoria própria.

A Figura 27 é a tela de cadastro de animal na aba de exames, nessa tela é possível cadastrar os exames pelos quais o animal passou, conforme os exames são cadastrados eles vão aparecendo na lista de exames. A lista de exames permite que o usuário tenha acesso aos exames já cadastrados do animal, podendo editar e excluir esses dados.

início Cadastrar Relatorios Comercio Sair

Cadastro de Animal

Animal Pesagem Vacina Coberturas Exame Prole

Exame Selezione... Data Exame: Veterinario Selezione... add Exame

Lista de Exames

Exame	Data Vacinacao	Veterinario	comandos
Não existe nenhuma exame cadastrado			

salvar

Lista de Animais

Codigo	Sexo	Pelagem	Raça	Nascimento	comandos
61		preta	Bonsmara	13/12/2013	Editar Excluir
65		mista	Guzerá	17/10/2013	Editar Excluir
62		branca	Brangus	17/10/2011	Editar Excluir

Figura 27 - Interface cadastro de animal, aba exames.
Fonte: Autoria própria.

A Figura 28 é a tela de cadastro de animal na aba de prole, nessa aba o usuário pode adicionar animais que são filhos do animal que está sendo cadastrado. Após ser inserido o animal filho, os dados dele aparecerão na lista de proles, essa lista traz todos os filhos do animal que está sendo cadastrado. Através da lista de proles é possível editá-las e excluí-las.

início Cadastrar Relatorios Comercio Sair

Cadastro de Animal

Animal Pesagem Vacina Coberturas Exame Prole

Codigo Prole 65 add Filho

Lista de Filhos

Codigo	Sexo	Pelagem	Raca	Nascimento	comandos
61	Femea	preta	Bonsmara	13/12/2013	Editar Excluir
65	Femea	mista	Guzerá	17/10/2013	Editar Excluir

salvar

Lista de Animais

Codigo	Sexo	Pelagem	Raça	Nascimento	comandos
61		preta	Bonsmara	13/12/2013	Editar Excluir
65		mista	Guzerá	17/10/2013	Editar Excluir
62		branca	Brangus	17/10/2011	Editar Excluir

Figura 28 - Interface de cadastro de animal, aba Prole.
Fonte: Autoria própria.

Apesar dos dados da pesagem, vacinação, cobertura, exame e prole serem adicionados nas listas eles ainda não estão salvos, eles somente serão salvos quando o cadastro do animal for efetivado.

Após a efetivação do cadastro do animal, os dados do mesmo aparecem na lista de animais cadastrados no sistema.

A Figura 29 é a tela de cadastro de venda de animais, através dela é possível armazenar informações a respeito de venda de animais e na mesma tela é possível ver as informações de outras vendas, permitindo que o usuário edite ou exclua a venda.

inicio Cadastrar Relatorios Comercio Sair

Cadastro de Venda

Venda Parcelas Animais

Codigo Data Venda Valor da venda

Codigo Comprador Nome Comprador Tipo de Pagamento

Forma de Pagamento Moeda

salvar

Lista de vendas

Codigo	Data	Comprador	Valor	comandos
4	23/12/2013	Fenando	5000.00	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
5	12/10/2013	Diego	7000.00	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
6	12/10/2013	Fenando	8000.00	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>

Figura 29 - Interface Venda de animais.
Fonte: Autoria própria.

A Figura 30 é a tela de cadastro de compra, nela são informados os dados referentes à compra de animais, na mesma tela é possível visualizar os registros de compras já registrados no sistema, podendo o usuário editar ou excluir esses dados.

início Cadastrar ▾ Relatorios ▾ Comercio ▾ Sair

Cadastro de Compra

Venda Parcelas Animais

Codigo Data Compra Valor da compra

Codigo Vendedor Nome Vendedor Tipo de Pagamento

Forma de Pagamento Moeda

salvar

Lista de Compras

Codigo	Data	Comprador	Valor	comandos
4	23/12/2013	Gregorio	5000.00	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
5	12/10/2013	Fabio	7000.00	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
6	12/10/2013	Gregorio	8000.00	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>

Figura 30 – Interface de compra de animais.
Fonte: Autoria própria.

Ambas as telas de cadastro de venda e compra de animais possuem as abas de parcelas Figura 31, onde é possível armazenar informações sobre as parcelas da comercialização, se essa possuir parcela. As parcelas ficam armazenadas na lista de parcelas, onde as mesmas podem ser excluídas, editadas ou o usuário pode informar o pagamento da parcela.

início Cadastrar ▾ Relatorios ▾ Comercio ▾ Sair

Cadastro de Venda

Venda Parcelas Animais

Nº Parcela Numero da Parcela: Valor Parcela

add Parcela

Lista de Parcelas

Parcela	Data da Pagamento	Valor da Parcela	comandos
1	23/12/2013	1000.00	<input type="button" value="Editar"/> <input type="button" value="Excluir"/> <input type="button" value="Pago"/>
2	23/01/2014	1000.00	<input type="button" value="Editar"/> <input type="button" value="Excluir"/> <input type="button" value="Pago"/>
3	23/02/2014	1000.00	<input type="button" value="Editar"/> <input type="button" value="Excluir"/> <input type="button" value="Pago"/>
4	23/03/2014	1000.00	<input type="button" value="Editar"/> <input type="button" value="Excluir"/> <input type="button" value="Pago"/>
5	23/04/2014	1000.00	<input type="button" value="Editar"/> <input type="button" value="Excluir"/> <input type="button" value="Pago"/>

salvar

Lista de vendas

Codigo	Data	Comprador	Valor	comandos
4	23/12/2013	Gregorio	5000.00	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
5	12/10/2013	Fabio	7000.00	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
6	12/10/2013	Gregorio	8000.00	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>

Figura 31 - Interface de compra de animais, aba parcelas.
Fonte: Autoria própria.

As telas de cadastro de venda e compra também possuem a aba de animais, Figura 32, onde são informados os animais que fazem parte da comercialização, nessa mesma aba é possível visualizar a lista de animais que já fazem parte da comercialização, podendo o usuário editar ou excluir esses dados.

inicio Cadastrar Relatorios Comercio Sair

Cadastro de Venda

Venda Parcelas Animais

Codigo Animal add

Lista de Animais na Venda					
Codigo	Sexo	Pelagem	Raca	Nascimento	comandos
65	f	mista	Guzerá	17/10/2013	Editar Excluir
62	f	branca	Brangus	17/10/2011	Editar Excluir

salvar

Lista de vendas				
Codigo	Data	Comprador	Valor	comandos
4	23/12/2013	Gregorio	5000.00	Editar Excluir
5	12/10/2013	Fabio	7000.00	Editar Excluir
6	12/10/2013	Gregorio	8000.00	Editar Excluir

Figura 32 - Interface de compra de animais, aba animais.

Fonte: Autoria própria.

A Figura 33 é a tela de cadastro de proprietário, onde o usuário pode cadastrar os dados de outros proprietários, esses dados são necessários para o cadastro de compra e venda de animais. Os criadores cadastrados podem ser visualizados na lista que fica na mesma página, esses dados podem ser excluídos ou alterados.

início Cadastrar ▾ Relatorios ▾ Comercio ▾ Sair

Cadastro de Criador

Criador **Fazendas**

Codigo

Nome Incrição Estadual UF Documento

CNPJ Estado Cidade

logradouro Bairro Endereço

numero

Lista de Criadores

Codigo	Nome	Documento	Cidade	comandos
18	Gregorio	963.331.172-18	ALECRIM	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
19	Fabio	56.822.416/0001-40	ALAZÃO	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>

Figura 33 – Interface de cadastro de propriedade.

Fonte: Autoria própria.

início Cadastrar ▾ Relatorios ▾ Comercio ▾ Sair

Cadastro de Criador

Criador **Fazendas**

Nome Tamanho Estado

Cidade logradouro Bairro

Endereço numero

Lista de Fazendas

Nome da fazenda	Tamanho	Cidade	Estado	comandos
Fazenda Esperança	2000.00	ALAZÃO	ALAGOAS	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
Fazenda Boa Vista	5000.00	ALECRIM	ALAGOAS	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>

Lista de Criadores

Codigo	Nome	Documento	Cidade	comandos
18	Gregorio	963.331.172-18	ALECRIM	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
19	Fabio	56.822.416/0001-40	ALAZÃO	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>

Figura 34 - Interface de cadastro de propriedade, aba fazendas.

Fonte: Autoria própria.

A Figura 35 é a tela de relatório de comercio, nessa tela o usuário pode realizar pesquisas referentes à compra e venda de animais. Essa tela permite que o

usuário realize pesquisas precisas ou pesquisas mais abrangentes a respeito de uma informação comercial.

início Cadastrar ▾ Relatorios ▾ Comercio ▾ Sair

Relatorio de comercio

Codigo Situação Venda Compra

Codigo Comprador Codigo Animal

Situacao Parcela Quitada Pendente Ambas Data comercio início:

Data comercio Fim: Vencimento Ini

Vencimento Ini Tipo Em dia Atrasada Ambas

Tipo Sintetico Analitico

Figura 35 - Interface de geração de relatório de comércio.
Fonte: Autoria própria.

Após a solicitação de geração do relatório, o sistema disponibilizara ao usuário um relatório em formato de PDF. Um exemplo desse relatório pode ser visto na Figura 36.



Web Pecuária

Relatório de Vendas

Codigo Venda	Data Venda	Valor da venda		Comprador
4	23/12/13 00:00	5000.00		Gregorio

Nº Parcela	Data de vencimento	valor	juros	desconto	Data de recebimento	Status
1	23/12/13 00:00	1000.00	0.00	0.00		Em dia
2	23/01/14 00:00	1000.00	0.00	0.00		Em dia
3	23/02/14 00:00	1000.00	0.00	0.00		Em dia
4	23/03/14 00:00	1000.00	0.00	0.00		Em dia
5	23/04/14 00:00	1000.00	0.00	0.00		Em dia

Figura 36 - Relatório de venda.
Fonte: Autoria própria.

4 CONSIDERAÇÕES FINAIS

Este capítulo apresenta as considerações finais do projeto, apresentando as dificuldades encontradas e o aprendizado adquirido durante o mesmo.

4.1 DIFICULDADES ENCONTRADAS

Durante a execução deste trabalho algumas dificuldades foram encontradas, entre elas a compreensão da tecnologia *JavaServer Faces* utilizada para o desenvolvimento do sistema. Essa dificuldade foi sendo superada no decorrer do projeto, utilizando-se de pesquisas para melhor compreensão dos componentes existentes dentro do framework *JavaServer Faces*.

Outra dificuldade encontrada foi a necessidade da reelaboração do projeto, isso fez com que um período muito longo do tempo que seria utilizado para o desenvolvimento do sistema fosse gasto. Esse replanejamento do projeto foi de total importância, pois sem a documentação e o escopo bem definido o desenvolvimento do projeto poderia falhar.

4.2 APRENDIZADO

Uma grande quantidade de conhecimento foi adquirida durante todo o processo de desenvolvimento do projeto, o conhecimento das ferramentas e linguagem de programação utilizadas para o seu desenvolvimento. Esse conhecimento foi adquirido devido às dificuldades encontradas, fazendo com que fosse necessária a realização de diversas pesquisas para auxiliar no decorrer do projeto.

O principal conhecimento adquirido foi que sem uma boa documentação e sem um escopo de projeto bem definido o desenvolvimento do sistema se torna ainda mais difícil, pois necessita muitas vezes de retrabalho.

REFERÊNCIAS

- ABIEC. **Pecuária Brasileira**. 2011. Associação Brasileira das Indústrias Exportadoras de Carnes. Disponível em: < http://www.abiec.com.br/3_pecuaria.asp >. Acesso em: 20 de abril. 2013.
- ARAÚJO, M. A. **Modelagem de dados com a Visual Paradigm - Do modelo de classes à criação do banco de dados**. Rio de Janeiro. SQL Magazine, n. 42. 2007.
- BRASIL. **Programa nacional de Banda Larga**. Decreto nº 7.175, Brasília, 12 de maio de 2010.
- CAELUM. **Laboratório Java com testes, JSF, web Services e Design Patterns**. Apostila do curso FJ-22. Capítulo 7. Disponível em: < <http://www.caelum.com.br/apostila-java-testes-jsf-web-services-design-patterns/> >. Acesso em: 01 de jan. 2014.
- CAMPOS, F et al. **Avaliação de software agropecuario**. ROCHA, ARC; MALDONADO, JC; WEBER, KC Qualidade de software: teoria e prática. São Paulo: Prentice Hall, p. 208-216.
- CARVALHO, J.G. V. **Ciclo do Scrum Solo**. 2012. Disponível em: < <http://engenhariasoftware.wordpress.com/2012/06/06/scrum-solo/> >. Acesso em: 26 de mai. 2013.
- CETIC. **TIC domicílios 2010**. Centro de Estudos sobre as Tecnologias da Informação e da Comunicação. Núcleo de Informação e Coordenação. São Paulo, 2011.
- COELHO, M. A. **Geografia do Brasil**. Série sinopse. 3 ed. São Paulo: Moderna, 1990.
- CPT SOFTWARE. **Bonanza Gold** (2013). Disponível em: < <http://www.cptsoftwares.com.br/produto/3/2035/bonanza-gold.html> >. Acesso em 10 de dez. 2012.
- DESENVOLVIMENTOAGIL. **Scrum**. 2013. Disponível em: < <http://desenvolvimentoagil.com.br/scrum> >. Acesso em: 20 de mai. 2013.
- EMBRAPA. **Criação de Bovinos de Corte no Estado do Pará: Manejo sanitário**. Dez. 2006. Versão Eletrônica. Disponível em: < http://sistemasdeproducao.cnptia.embrapa.br/FontesHTML/BovinoCorte/BovinoCortePara/paginas/manejo_san.html >. Acesso em: 20 de abril. 2013.
- FOCUS. **Pecuária Bovina no Brasil: Maior produtividades com menor impacto Socioambiental**. 2010
- FORESIGHT. **The future of food and farming: challenges and choices for global sustainability**. The Government Office For Science. London, 2011.

- GONÇAVES, E. **Desenvolvendo aplicações web com jsp,servelets,javaservert faces,hibernate,EJB 3 Persistence e Ajax**. Rio de Janeiro: Editora Ciencia moderna Ltda. 2007.
- LUCAS, E. **Postgresql, muito prazer!** visão geral das funcionalidades do postgresql. SQL Magazine, v. 36, p. 123–142, 2006.
- MARIOTTI, F. S. **Kanban: o ágil adaptativo**. Rio de Janeiro. Engenharia de Software Magazine, n. 45. 2012.
- MENDES, D. R. **Programação Java com Ênfase em Orientação a Objetos**. São Paulo: Novatec. 456p. 2009.
- MICROSOFT Suport. **Description of database normalization basics in Access 2000**. Ago. 2004. Disponível em: < <http://support.microsoft.com/kb/209534/en-us> >. Acesso em: 30 de nov. 2013.
- MILANE, André. **PostgreSQL: Guia do programador**. Novatec, 2008 p -25,26,27,28.
- MOUNTAIN Goat Software. **Scrum Images**. 2005. Disponível em: < <http://www.mountaingoatsoftware.com/agile/scrum/images/> >. Acesso em: 20 de mai. 2013.
- MYSQL workbench. Version 6.0. Disponível para download em: < <http://dev.mysql.com/downloads/tools/workbench/> >. Acesso em: 18 de mai. 2013.
- NETO, Hyberville. **Localização dos maiores rebanhos**. Fev. 2012. Disponível em: < <http://www.scotconsultoria.com.br/noticias/todas-noticias/8733/localizacao-dos-maiores-rebanhos.htm> >. Acesso em: 24 de abr. 2013.
- ORACLE. **O que é a tecnologia Java e por que é necessária?**. Disponível em: < http://www.java.com/pt_BR/download/faq/whatis_java.xml >. Acesso em: 17 de mai. 2013.
- POSTGRESQL. **About**. 2009. Disponível em: <<http://www.postgresql.org/about/>>. Acesso em: 25 de mai. 2013.
- PROCREARE. **Pacote fazenda de sucesso**: Procreate Software para controle Agropecuario (2013). Disponível em: < <http://www.procreate.com.br/infovermelho.htm> >. Acesso em: 10 de dez. 2012.
- RABELO, Lucas. **Planejamento e Gestão na Produção Pecuária**. In: SIMPÓSIO NACIONAL SOBRE PRODUÇÃO E GERENCIAMENTO DA PECUÁRIA DE CORTE, 5. 2012, Belo Horizonte. UFMG – Escola de Veterinária, 2012 p. 4-20.
- SCHWABER, K., SUTHERLAND, J. **Guia do Scrum**, Um guia definitivo para o Scrum: As regras do jogo. 2013.

SIMÕES, A. R. P., SILVA, R. M., OLIVEIRA, M. V. M., CRISTALDO, R. O., BRITO, M. C. B. **Avaliação econômica de três diferentes sistemas de produção de leite na região do Alto Pantanal Sul-mato-grossense.** Universidade Estadual do Mato Grosso do Sul. 27 de nov. 2009.

SOFTFOR. **A3 software para gestão pecuária** (2013). Disponível em: < <http://www.a3pecuaria.com.br/site/website/home/index.htm> >. Acesso em: 11 de dez. 2012.

SVEIBY, Karl Erick. **A nova riqueza das organizações:** gerenciando e avaliando patrimônios de conhecimento. Campus, 2003.

TOFFOLI, G. **iReport Designer Getting Started.** JASPERSOFT CORP. Disponível em: < <http://community.jaspersoft.com/wiki/ireport-designer-getting-started> >. Acesso em: 18 de mai. 2013.

XMIND. Version 3.4.0. Disponível para download em: < <http://www.xmind.net/download/win> >. Acesso em: 18 de mai. 2013.

Apêndice A – Cronograma

O cronograma inicialmente proposto possuía um tempo de duas semanas para cada Sprint, as atividades e o tempo previsto para a sua conclusão podem ser vistos na Tabela 5:

Tabela 5 - Cronograma proposto.

Sprints	Cronograma																								
	01/4	14/4	15/4	28/4	29/4	12/5	13/5	26/5	27/5	09/6	10/6	23/6	24/06	07/7	08/7	21/7	22/7	04/8	05/8	18/8	19/8	01/9	02/9	16/9	
Construção do banco de dados	█	█																							
Cadastro de animais			█	█																					
Cadastro de reprodução					█	█																			
Cadastro de eventos							█	█																	
Cadastro de dados base									█	█															
Agenda, lactação, dieta											█	█													
Função de relatórios													█	█	█	█									
Telas de relatórios																	█	█	█	█					
Testes finais																					█	█			
Ajustes finais																							█	█	

Autor: Autoria Própria.

A Tabela 6 é o cronograma realizado, devido aos imprevistos ocorridos pela falta de um planejamento adequado no início do projeto, o cronograma atrasou 3 meses em relação ao previsto.

Tabela 6 - Cronograma Final.

Atividade	Mês/Semana 2013																2014	
	Mai		Jun		Jul		Ago		Set		Out		Nov		Dez		Jan	
	1º	2º	1º	2º	1º	2º	1º	2º	1º	2º	1º	2º	1º	2º	1º	2º	1º	2º
Replanejamento	■	■	■															
Documentação		■	■															
1º Sprint				■														
Banco de dados				■														
Conexão do Sistema com o banco de dados				■														
2º Sprint					■													
Popular o banco de dados					■													
3º Sprint						■												
Cadastro de usuário						■												
Acesso ao sistema						■												
Recuperação de senha						■												
4º Sprint							■	■										
Cadastro de animal							■	■										
Cadastro de lote de animais								■										
5º Sprint									■	■								
Cadastro vacinação									■	■								
Cadastro de reprodução									■	■								
Cadastro de pesagem										■								
Cadastro de perda											■							
Cadastro de exames												■						
6º Sprint													■					
Cadastro de Produtor													■					
Cadastro de Propriedades													■					
7º Sprint														■	■			
Cadastro de compra														■	■			
Cadastro de venda															■			
8º Sprint																■	■	
Cadastro de eventos																■	■	
Tela Inicial do Sistema																	■	■
9º Spring																	■	
Relatório de controle de animais																	■	
10º Sprint																		■
Relatório de perdas de animal																		■
Relatório de comercialização																		■
Testes				■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Documentação Projeto									■	■	■	■	■	■	■	■	■	■

Autor: Aatoria Própria

