UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMATICA E ESTATISTICA

DANIEL MARAGNO COELHO
JHONATAN EZEQUIEL FACCIN

SOFTWARE PARA GERENCIAMENTO E CONTROLE DA PRODUÇÃO DE LEITE

FLORIANÓPOLIS - SC
06/2017

DANIEL MARAGNO COELHO
JHONATAN EZEQUIEL FACCIN

SOFTWARE PARA GERENCIAMENTO E CONTROLE DA PRODUÇÃO DE LEITE

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Bacharel em Ciência da computação.

Orientador: José Eduardo De Lucca

FLORIANÓPOLIS – SC
06/2017

DANIEL MARAGNO COELHO
JHONATAN EZEQUIEL FACCIN

SOFTWARE PARA GERENCIAMENTO E CONTROLE DA PRODUÇÃO DE LEITE

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação

Aprovada em __/__/__

BANCA EXAMINADORA

_____
Profº. José Eduardo De Lucca
Orientador

_____
Cleberson Charles Colombo Faccin
Membro da Banca

_____
Profº. Jose Francisco Danilo de Guadalupe Correa Fletes
Membro da Banca

RESUMO

A pecuária de leite é uma das atividades mais importantes no agronegócio brasileiro. Sendo praticada principalmente nas propriedades rurais pertencentes à agricultura familiar.

No mercado de *software* brasileiro, existem alguns sistemas disponíveis para o gerenciamento da produção de leite nas propriedades rurais. Observando as más funcionalidades destes sistemas, juntamente com as necessidades dos produtores de leite, além do crescimento da informatização do meio rural. Originou-se a proposta do Ordenha Digital.

Ordenha Digital é um software desenvolvido neste trabalho. Com o intuito de auxiliar os produtores de leite. Em especial os produtores da agricultura familiar, a gerenciarem a atividade.

ABSTRACT

The milk livestok is one of the most importants Brazilian agribusiness activities. Being mainly practiced in family farming properties.

In Brazilian software market there are some available systems to management of milk production in rural properties. Observing the few and poor features of this systems, along with the milk producers' needs, beyond the growth of information in the countryside, originates the proposal of Ordenha Digital.

Ordenha Digital is a software developed in this project in order to help milk producers, especially producers from family farming, to manage your activities.

LISTA DE TELAS

LISTA DE GRÁFICOS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A pecuária no Brasil tem sua origem ligada à exploração do gado trazido durante o período da colonização portuguesa. Inicialmente como força de trabalho e pecuária de corte. Até meados do século XIX o leite tinha papel secundário e a pouca disponibilidade do produto impediu que o consumo se tornasse um hábito naquela época. Com o passar dos anos, diversos desenvolvimentos tecnológicos permitiram novos sistemas de embalagens, tratamentos térmicos, melhores sistemas de transporte, possibilitaram que o produto chega-se até o consumidor em ótimas condições de consumo e com maior durabilidade (ALVES, 2001).

A bovinocultura de leite é uma das atividades mais praticadas no agronegócio brasileiro. Segundo o último censo agropecuário realizado em 2006, existem no Brasil aproximadamente 5,2 milhões de estabelecimentos rurais, dos quais, cerca de 1,3 milhões tem produção de leite, ou seja, um a cada quatro estabelecimentos (MAPA, 2014).

A pecuária de leite tem mostrado um avanço na produção muito significativo. Apresentando um crescimento de mais de 75% no período entre 2000 a 2014 (IBGE, 2014). Porém esse crescimento se deve muito mais pelo aumento no número de vacas ordenhadas do que pelo aumento da produtividade (MAPA, 2014). Observando os dados da produtividade do rebanho brasileiro em 2012, a média de produção por vaca era de 1,42 toneladas de leite ao ano. Esse número é muito abaixo do encontrado em outros países tradicionais na produção de leite. Nos Estados Unidos, por exemplo, a produtividade é de 9,48 toneladas/ano (EMBRAPA, 2017). As principais causas dessa baixa produtividade incluem a utilização de animais sem aptidão ou potencial genético apropriado para a produção de leite, manejo alimentar, sanitário e reprodutivo inadequado, pouca ou nenhuma assistência técnica, além do baixo nível de escolaridade dos produtores, o que dificulta a utilização de tecnologias disponíveis (MAPA, 2014).

Segundo o Portal Brasil, a produção nacional é capaz de fornecer aproximadamente 170 litros de leite por habitante ao ano. Essa quantidade é inferior ao indicado pelos órgãos de saúde nacionais e internacionais que é de 210 litros ao ano por habitante. E com uma estimativa de crescimento da população até 2023 de 216 milhões de habitantes, o governo federal por meio

do Plano Mais Pecuária lançado em 2014, pretende aumentar de forma sustentável a produtividade e a competitividade da pecuária bovina de leite e corte.

1.1 Justificativa

Devido à importância da agropecuária na economia brasileira e com projeções de uma demanda crescente para os próximos anos, levando em conta ainda a baixa produtividade do setor, muito em decorrência do baixo nível de instrução dos produtores, que encontram dificuldades para gerenciar os recursos de suas propriedades, justificando assim a realização deste trabalho.

Sendo assim, o desenvolvimento de um software para a administração de propriedades rurais produtoras de leite, com ênfase na agricultura familiar, tem como intuito auxiliar o administrador com a coleta e organização das informações, gerando históricos de produtividade e lucratividade. Embora existam diversos softwares no mercado, para (DEPONTI 2014) há um descompasso entre os *softwares* e ferramentas de gestão existentes e a aderência por parte dos produtores, tendo em vista a desconexão com as necessidades e habilidades deste público.

1.2 Objetivos

1.2.1 Geral

O desenvolvimento de um sistema capaz de auxiliar a gestão da produção de leite nas propriedades rurais.

1.2.2 Específicos

- Identificar as principais necessidades dos agropecuaristas na gestão de suas atividades de produção.
- Pesquisar ferramentas e *softwares* existentes no mercado voltado à pecuária de leite.

- Conceber um *software* capaz de auxiliar no gerenciamento de algumas das áreas da produção de leite desde criação do rebanho, controle sanitário, saúde animal, controle de produto e estoque.

1.2.3 Metodologia

Inicialmente foi realizada uma pesquisa junto aos órgãos governamentais como a Embrapa e o Ministério da Agricultura Pecuária e Abastecimento, para compreender a dinâmica da pecuária de leite e sua importância na sociedade brasileira. Além de conhecer o perfil dos agropecuaristas, para então fomentar a estrutura teórica e prática do projeto.

Subsequentemente, outra pesquisa foi realizada para identificarmos a existência de ferramentas e *softwares* disponíveis no mercado, quais são suas funcionalidades e que grupo de produtores eles buscam atender.

Com a estrutura teórica definida, desenvolveu-se um modelo de negócio com o intuito de auxiliar no empreendimento do *software,* além de obter uma melhor compreensão do perfil do produtor de leite. Em especial aos produtores que se enquadram na agricultura familiar.

Com base no modelo de negócio e nas pesquisas previamente realizadas, um sistema foi desenvolvido para auxiliar principalmente o pequeno e médio produtor a gerenciar a produção e o fluxo de trabalho dentro da propriedade.

2 FUNDAMENTAÇÃO

2.1 Cenário Atual da Produção de Leite

O Brasil ocupa um lugar de destaque no cenário mundial, sendo o quinto maior produtor de leite de vaca, atrás apenas da União Europeia, Estados Unidos, Índia e China, segundo dados do Departamento de Agricultura dos Estados Unidos (United States Department of Agriculture - USDA). No cenário nacional, a bovinocultura de leite é predominante nas regiões Sul e Sudeste. A região Sul é a maior produtora, responsável por 35,2% do montante produzido, que em 2015 foi de 35 bilhões de litros. É possível observar no Gráfico 1, o crescimento ao longo dos anos e como se distribui a produção nas regiões do país.

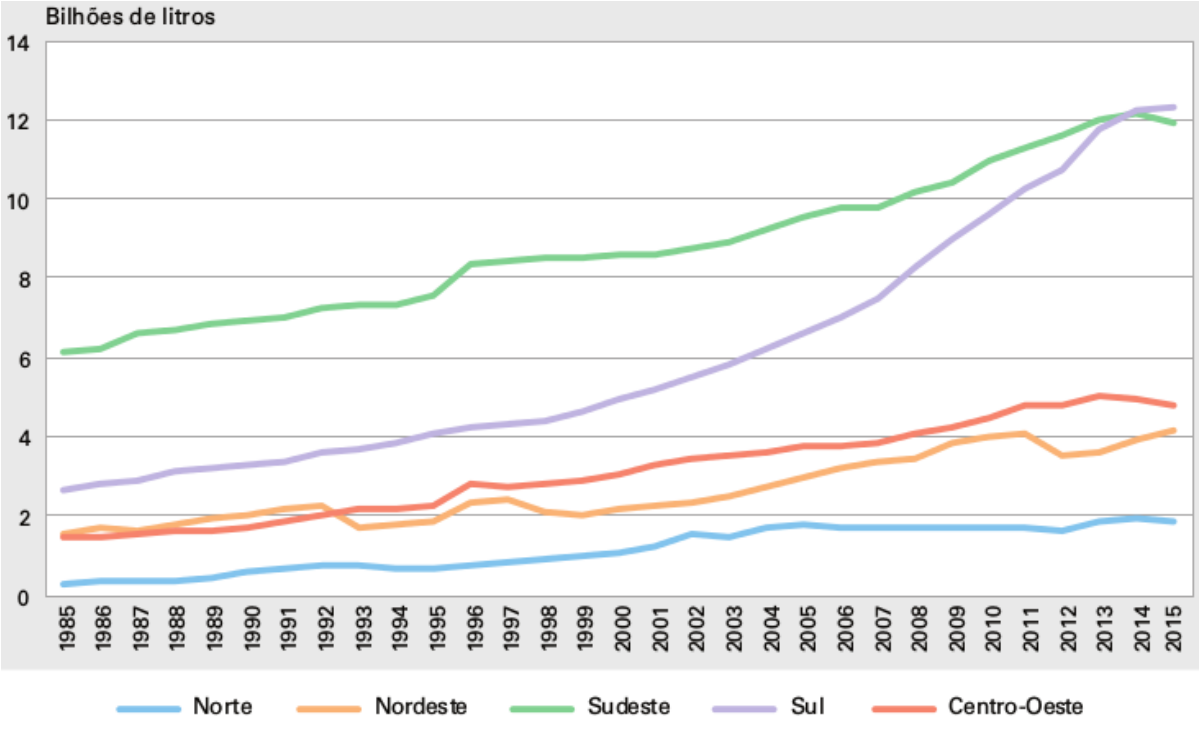


Gráfico 1- Evolução na produção de leite nas regiões do país, entre 1985-2015.
Fonte: IBGE, Diretoria de Pesquisas, Coordenação de Agropecuária, Pesquisa da Pecuária Municipal 1985-2015.

Grande parte dessa quantidade é adquirida pelos estabelecimentos industriais sob algum tipo de inspeção sanitária, seja ela municipal, estadual ou

federal. Podemos observar a diferença de produção e de aquisição ao longo dos anos através do Gráfico-02. Isso reflete a produção não fiscalizada. Essa diferença também pode ser explicada pelo fato de que 31% dos estabelecimentos rurais não vendem ou beneficiam seu produto (ZOCCAL, 2012).

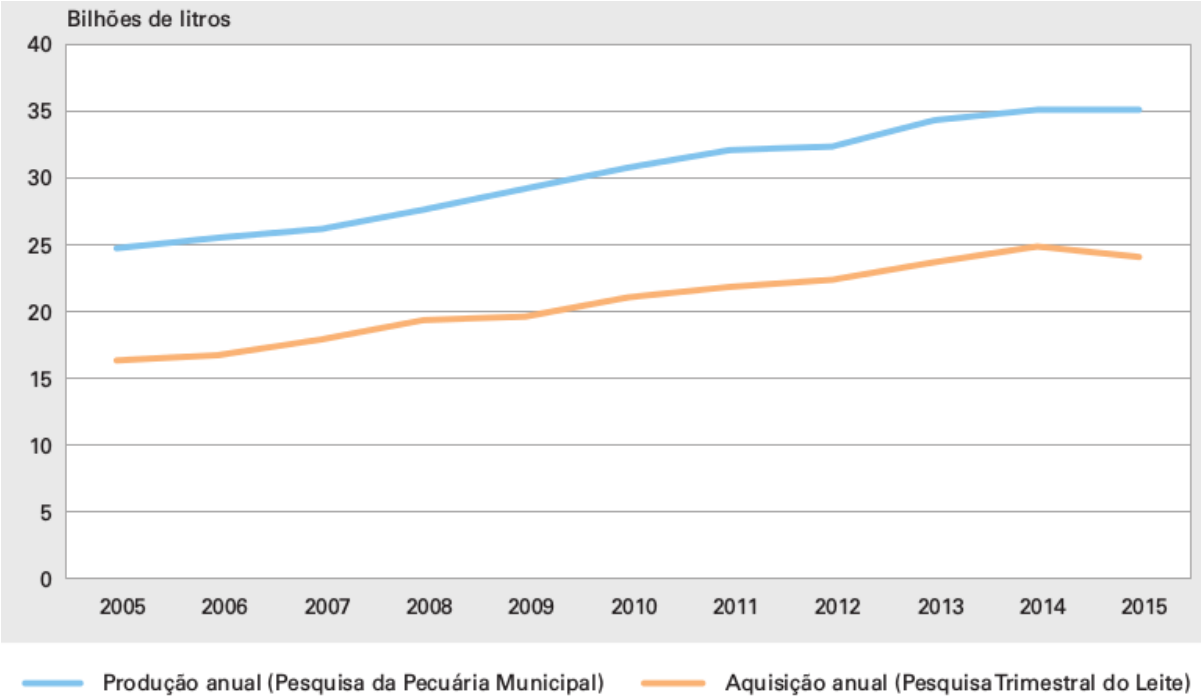


Gráfico 2 - Diferença entre produção e aquisição do leite por estabelecimentos industriais.
Fonte: IBGE: Instituto Brasileiro de Geografia e Estatística - Produção da Pecuária Municipal. Rio de Janeiro, v. 43, p.1-49, 2015.

2.2 Agricultura Familiar e a Produção de Leite

A agricultura familiar, de acordo com a *Food and Agriculture Organization of United Nations* (FAO, 2014) consiste em um "meio de organização das produções agrícola, florestal, pesqueira, pastoril e aquícola gerenciadas e operadas por uma família, tendo como mão de obra o núcleo familiar". Mais especificamente a legislação brasileira define o que é agricultura familiar pela lei nº 11.326 de 24 de julho de 2006.

De acordo com o Portal Brasil cerca de 70% dos alimentos consumidos no Brasil é proveniente dos agricultores familiares, mais especificamente no setor de laticínios é responsável por 58% da oferta do produto no mercado. Pode-se notar através do gráfico 3, que a predominância dos estabelecimentos que se enquadram na agricultura familiar ocorre em todas as regiões do país.

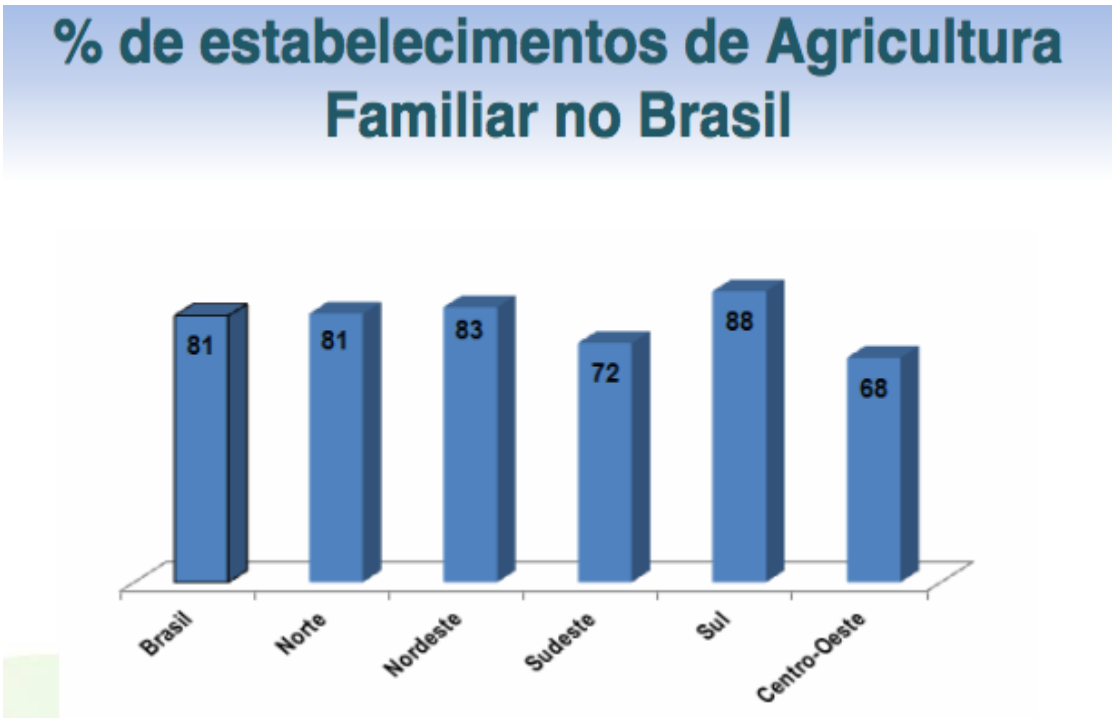


Gráfico 3 - Porcentagem dos estabelecimentos de agricultura familiar no Brasil.
Fonte: Embrapa/MDA/SAF.

É possível considerar que o leite é um dos produtos mais importantes para a agricultura familiar. Isso é notado através da forte relação da agricultura familiar com a produção de leite, através do Gráfico 4.

Gráfico 4 - Porcentagem dos estabelecimentos com leite de agricultura familiar no Brasil.

Fonte: Embrapa/MDA/SAF.

2.3 Perfil do produtor de leite

A predominância de pequenos agricultores na cadeia produtiva de leite pode ser observada pela tabela a seguir, onde mostra que 91,5% dos produtores tem uma produção inferior a 100 litros por dia o que caracteriza propriedades de subsistência sem o uso de tecnologia (ZOOCAL, 2012).

Estrato de produção (litros leite/dia)	Produtores*		Produção**	
	Número	%	Volume	%
Menos de 100	1.236.130	91,5	15.085.220	47
De 100 a 200	73.559	5,4	6.098.280	19
De 201 a 500	32.785	2,4	5.773.319	18
De 201 a 1000	6.109	0,5	2.567.698	8
Mais de 1000	2.226	0,2	2.567.698	8
TOTAL	1.350.809	100,0	32.096.214	100,0

Dados do IBGE/Pesquisa da Pecuária Municipal 2011
Observação: Os dados do Censo Agropecuário se referem a 2006, porém não ocorreram no País fatos relevantes que possam ter provocado mudanças no setor as quais invalidem uma análise sobre os sistemas de produção em uso atualmente.

Tabela 1 – Volume da produção diária de estabelecimentos com leite no Brasil.

Fonte: Disponível em<http://www.sna.agr.br/uploads/AnimalBusiness_09_34.pdf>. Acessado em 27/06/2016

Outro aspecto importante sobre o perfil dos produtores é o baixo nível de escolaridade que é apresentado no gráfico-06. Ainda que os dados do gráfico não referenciem exclusivamente os produtores de leite, e sim sobre dirigentes de quaisquer estabelecimentos rurais do país. Isso nos dá uma perspectiva sobre o grau de escolaridade dos produtores de leite. Embora, mesmo que o produtor tenha um grau educacional baixo, isso não afeta a capacidade operacional. Entretanto, com baixo índice escolar dificilmente proporcionará uma alta capacidade de gestão (TALIARINE, 2015).

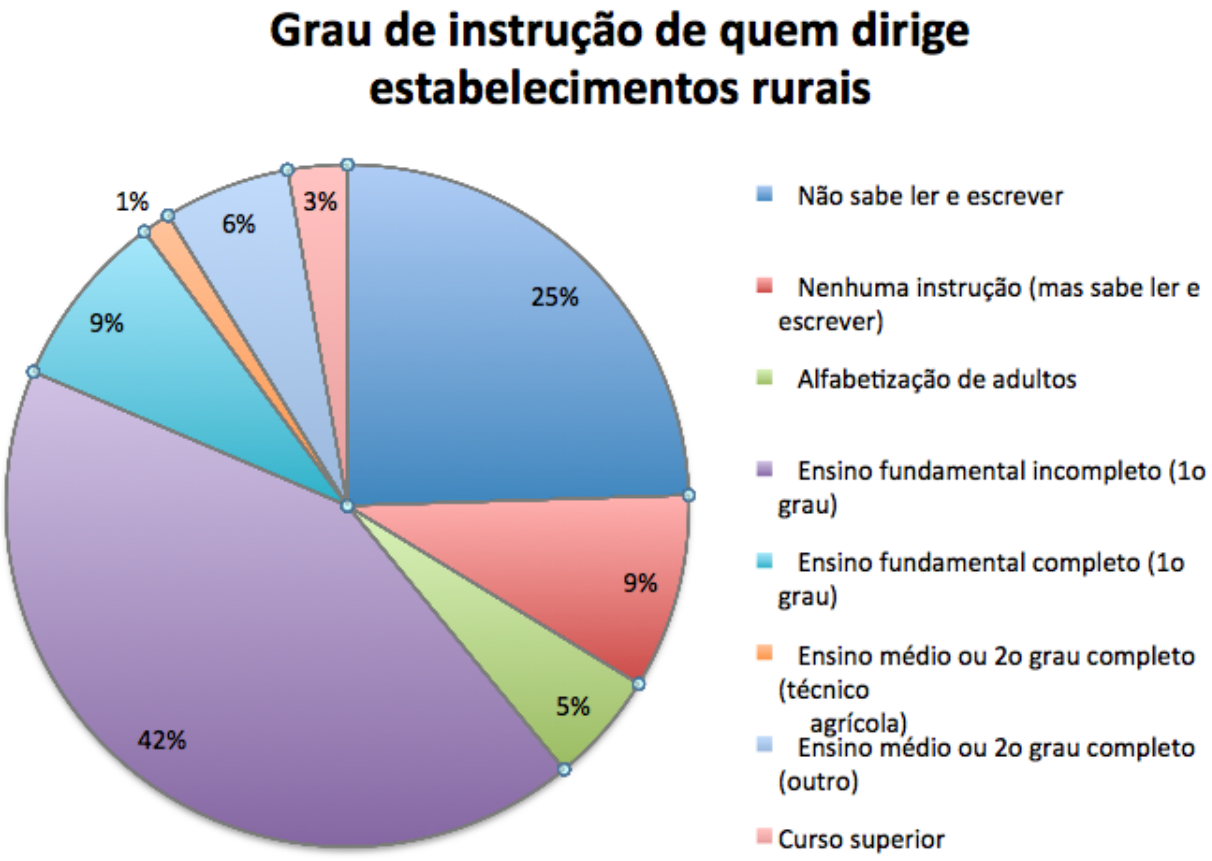


Gráfico 5 - Grau de instrução de dirigentes de estabelecimentos rurais.
Fonte: Elaboração dos autores com base nos dados do IBGE/Censo agropecuário 2006

2.4 Agricultura familiar e a produção de leite em Santa Catarina

No estado de Santa Catarina, a predominância da agricultura familiar é muito expressiva, onde 87% dos estabelecimentos rurais são de agricultores

familiares e 47,8% desses estabelecimentos tem produção de leite de vaca. (IBGE, 2006). Além disso, a agricultura familiar catarinense representa 90% dos estabelecimentos produtores de leite do estado e detém 87% do rebanho produtor de leite. (EPAGRI, 2016).

Apesar de a agricultura catarinense ser baseada em pequenas propriedades rurais, o agricultor diferencia-se dos produtores do restante do país quanto ao grau de escolaridade. Se relacionar o gráfico-06 apresentado anteriormente, com o gráfico-07, que diz respeito ao grau de instrução de dirigentes de propriedades rurais de Santa Catarina, observa-se que embora o grau de escolaridade em ambos seja baixo, existe uma melhora significativa, pois a nível nacional 25% não sabem ler e escrever, enquanto em Santa Catarina esse índice cai para 2%.

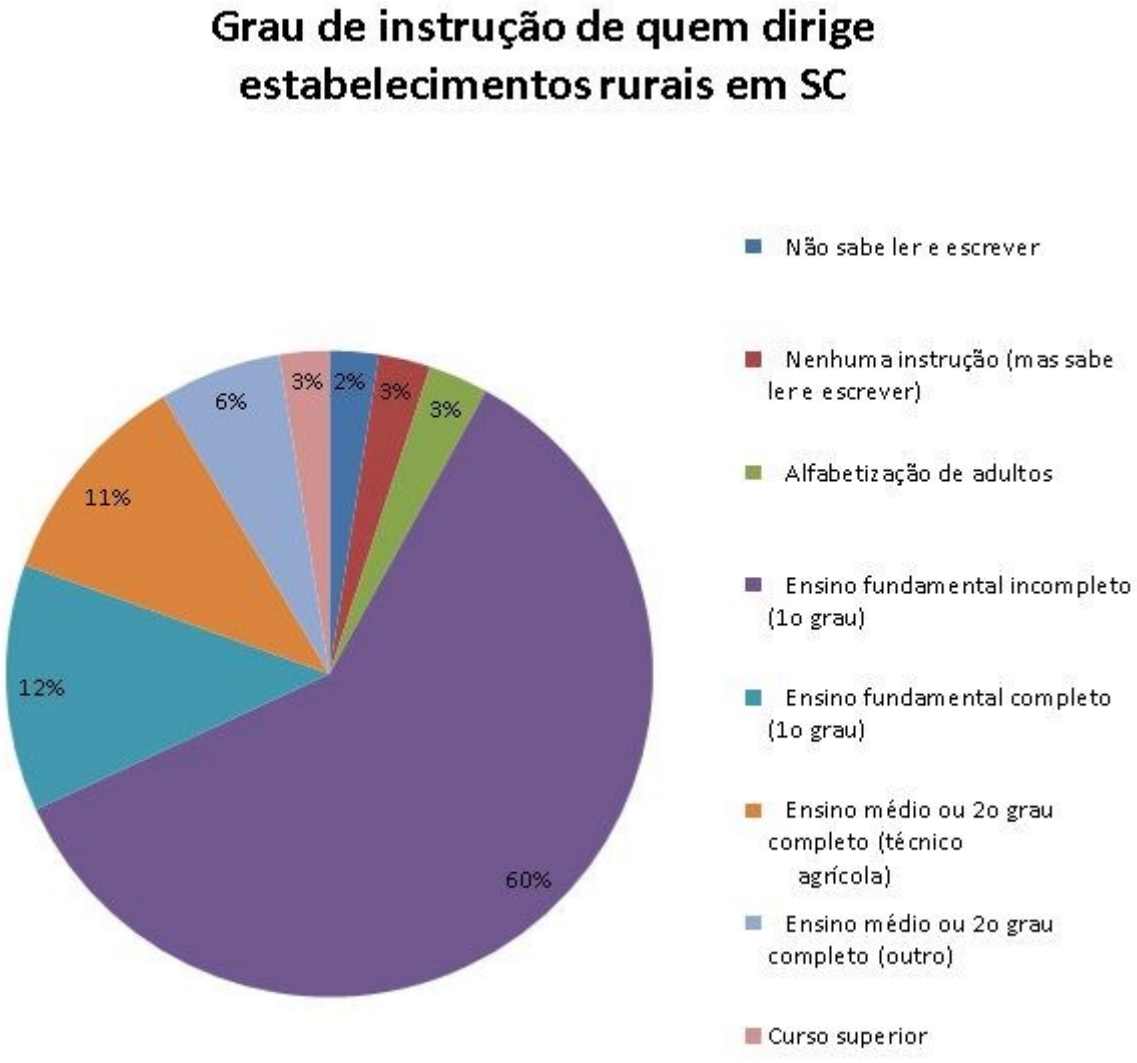


Gráfico 6 - Grau de instrução de dirigentes de estabelecimentos rurais de Santa Catarina

Fonte: Elaboração dos autores com base nos dados do IBGE/Censo agropecuário 2006

3 ESTADO DA ARTE

Foram realizadas pesquisas por *softwares* disponíveis no mercado com características semelhantes às apresentadas neste trabalho. Todos os softwares aqui descritos estão disponíveis para uso independente do custo para o usuário.

Além da identificação de aplicações semelhantes, foram levantadas as características técnicas destes *softwares* e foi realizada a descrição em linhas gerais de suas principais funcionalidades.

3.1 SGFLeite

O SGFLeite é um *software* desenvolvido para plataforma *Windows*, voltado para a gerência da produção de leite. Permite controle do rebanho, controle da produção, reprodução, além do controle sanitário, financeiro e zootécnico da propriedade (SGFLEITE, 2017).

O *software* é comercializado a um preço de R$ 249,90 reais a licença vitalícia, segundo informações do site da empresa referentes a junho de 2017. A licença é vinculada ao CPF ou CNPJ do produtor, e dá o direito de utilizar em um único dispositivo, embora seja possível a utilização em mais de uma máquina mediante pagamento de 30% do valor da licença. Para produtores que desejam adquirir atualizações que são disponibilizadas anualmente, o custo também é 30% do valor da licença vigente (SGFLEITE, 2017).

A imagem a seguir se refere à tela do sistema SGFLeite.

Tela 1: Tela inicial do SGFLeite

Abaixo se encontra uma tabela com características técnicas do *software.*

Tabela 2: Características técnicas do SGFLeite

SGFLeite	
Nome	SGFleite
Desenvolvedor	Lidia Joelma Lopes Pinto
Plataforma	*Windows* 98 ou posterior
Versão	2016
Web site	http://www.sgfleite.com.br/index.html
Licença	Proprietária
Estado de desenvolvimento	Ativo

3.2 4milk

O 4milk é um *software* desenvolvido pela empresa 4Hoofs Tecnologia da Informação LTDA, empresa sediada na cidade de Nova Lima no estado de Minas Gerais. Este *software* permite os manejos reprodutivos, zootécnicos e sanitários, além de gerir a produção das matrizes. O aplicativo também disponibiliza ferramentas para a importação de dados de outros sistemas de

forma gratuita. Também oferece uma plataforma para comércio de animais entre usuários do sistema (4MILK, 2017).

A imagem abaixo é se refere à tela inicial do aplicativo 4milk.

Tela 2: Tela inicial do 4milk

Abaixo se encontra uma tabela com características técnicas do *software.*

Tabela 3: Características técnicas do 4milk

4milk	
Nome	4milk
Desenvolvedor	4Hoofs Tecnologia da Informação LTDA
Plataforma	*Android* 4.2 ou posterior, IOS 7.1 ou posterior
Versão	2.0.3
Web site	http://www.4milk.com.br/Home

Licença	Gratuita
Estado de desenvolvimento	Ativo
Ano de lançamento	2016

3.3 Roda da Reprodução

A Roda da Reprodução é um aplicativo para o sistema Android desenvolvido pela Empresa Brasileira de Pesquisa Agropecuária (Embrapa) que permite gerenciar o manejo reprodutivo e produtivo do rebanho. O aplicativo é disponibilizado de forma gratuita pela empresa.

Abaixo a tela inicial do aplicativo Roda da Reprodução

Tela 3: Tela inicial da Roda da Reprodução

Abaixo a tabela com características técnicas do aplicativo.

Tabela 4: Características técnicas do Roda da Reprodução.

Roda da Reprodução	
Nome	Roda da Reprodução
Desenvolvedor	Empresa Brasileira de Pesquisa Agropecuária
Plataforma	*Android* 4.0.3 ou posterior
Versão	1.0 Rev. 8
Web site	https://www.embrapa.br
Licença	Gratuita
Estado de desenvolvimento	Ativo
Ano de lançamento	2016

3.4 Gisleite

Gisleite é um sistema *web* desenvolvido em parceria entre Embrapa Gado de Leite e a empresa Gemini Sistemas com apoio da Fundação de Amparo à pesquisa de Minas Gerais (FAPEMIG), da Financiadora de Estudos e Projetos (FINEP) e da Prefeitura Municipal de Juiz de Fora (EMBRAPA, 2016). O sistema é oferecido de forma gratuita, embora o suporte e treinamento possua custo que ficam ao cargo da empresa Gemini Sistemas.

As principais funcionalidades do *software* são o gerenciamento do manejo produtivo, reprodutivo, saúde animal, controle financeiro e qualidade do leite.

Abaixo a tela inicial do Gisleite.

Tela 4: Tela inicial do Gisleite.

Abaixo a tabela com características técnicas do Gisleite.

Tabela 5: Características técnicas do Gisleite.

Gisleite	
Nome	Gisleite
Desenvolvedor	Gemini Sistemas
Plataforma	*Web,* indicado o uso dos navegadores Mozila Firefox ou Google Chrome
Versão	2.0
Web site	http://gisleite.cnpgl.embrapa.br/
Licença	Gratuita
Estado de desenvolvimento	Ativo
Ano de lançamento	2012

3.5 Comparativo entre os *softwares* pesquisados

Afim de uma maior compreensão sobre os *softwares* pesquisados foi realizado um levantamento sobre as principais funcionalidades encontradas nesses sistemas. Na tabela abaixo está representado esse comparativo.

Tabela 6: comparativo entre *softwares*

Funcionalidades dos *softwares.*					
Funcionalidades	**Gisleite**	**Roda reprodução**	**4milk**	**SGFLeite**	**Ordenha Digital**
Cadastro de rebanho	Sim	Em parte	Sim	Sim	Sim
Gerenciamento da produção de leite do rebanho	Sim	Não	Sim	Sim	Sim
Gerenciamento da produção de leite individual	Sim	Não	Não	Sim	Não
Controle de eventos reprodutivos	Sim	Sim	Sim	Sim	Sim
Registro de partos	Sim	Sim	Sim	Sim	Sim
Previsão de partos	Sim	Sim	Sim	Sim	Sim
Registro de cruzas	Sim	Sim	Sim	Sim	Sim
Registro de abortos	Sim	Sim	Sim	Sim	Sim
Registro de secagem da vaca	Sim	Sim	Sim	Sim	Não
Controle sanitário	Sim	Não	Sim	Sim	Sim
Possibilidade de vincular mais de uma fazenda ao sistema	Sim	Não	Sim	Sim	Não
Relatórios de produção	Sim	Não	Sim	Sim	Sim
Gerenciamento financeiro	Sim	Não	Não	Sim	Não
Cadastrar fornecedores e cliente da propriedade	Sim	Não	Não	Sim	Não
Cadastro das áreas de plantio	Sim	Não	Não	Em parte	Não
Manejo de animais em áreas de pastagem	Sim	Não	Não	Sim	Não
Gerenciamento da produção de alimentos	Sim	Não	Não	Não	Não
Relatórios zootécnicos	Sim	Não	Sim	Sim	Não
Relatórios econômicos e fluxo de caixa	Sim	Não	Não	Sim	Não

Cadastro e gerenciamento de insumos e estoque	Sim	Não	Não	Sim	Não
Relatório de controle do rebanho	Sim	Não	Sim	Sim	Não
Controle de qualidade do leite	Sim	Não	Não	Não	Não

4 OPORTUNIDADE DE NEGÓCIO

Segundo dados do Centro de Estudos sobre as Tecnologias da Informação e da Comunicação (Cetic). Em 2015, apenas 22% dos domicílios rurais tinham acesso à internet, número significativamente menor aos 56% encontrado na área urbana. Todavia, houve um crescimento muito significativo nos últimos anos, se comparado a 2008, quando o número de domicílios rurais com acesso a internet era de apenas 4% (CETIC, 2015).

Tendo em vista a popularização da internet no campo, aliado à importância econômica da pecuária de leite, principalmente para a agricultura familiar (como descritos nos capítulos iniciais deste estudo), configura-se um cenário propício para o desenvolvimento de um serviço *web* de gestão da produção de leite.

4.1 Modelo de negócio

Com o intuito de desenvolver um sistema voltado a gestão da produção de leite, doravante chamado de Ordenha Digital, elaborou-se um modelo de negócio seguindo os nove componentes do *Business Model Canvas*, proposto por Osterwalder e Pigneur (2011). Essa ferramenta visual tem como objetivo fornecer uma visão holística e maleável do plano de negócio, auxiliando nos processos de criação, diferenciação e inovação, com o intuito de aumentar o número de clientes e potencializar os lucros (SEBRAE, 2015).

Abaixo seguem as descrições dos componentes. O quadro Canvas completo encontra-se disponível no Anexo I.

4.1.1.1 Segmentos de Clientes

Embora o foco do Ordenha Digital seja a agricultura familiar, a proposta do *software* pode abranger todos os produtores de leite com acesso a internet, inclusive cooperativas de leite, que poderiam disponibilizar o sistema para seus cooperados.

4.1.1.2 Proposta de Valor

A proposta de valor apresentada pela Ordenha Digital visa à gestão da produção de leite, mantendo o histórico da produção, além do gerenciamento da mão de obra envolvida no processo.

4.1.1.3 Canais

Os canais de relacionamento com os potenciais clientes são as mídias sociais, telefone, exposição em feiras agrícolas, além do *site* do sistema.

O *site* e as mídias sociais são importantes para a divulgação do sistema e a exposição do trabalho, além de possibilitar que os clientes emitam opiniões sobre o produto.

A exposição em feiras agrícolas de pecuária de leite permite um contato mais próximo a clientes em potencial, também permitindo obter *feedback* instantâneo.

Também o contato e divulgação junto aos extensionistas rurais da Epagri que atuam prestando assistência técnica às famílias produtoras de leite em Santa Catarina.

4.1.1.4 Relacionamento com Clientes

O relacionamento com os clientes é realizado de forma virtual, através dos canais de comunicação da empresa.

4.1.1.5 Fontes de Receita

Um possível modelo de receita seria a cobrança de um valor por um determinado período. O valor cobrado seria com base apenas na quantidade de vacas produtoras de leite.

4.1.1.6 Recursos Principais

Os principais recursos do projeto é a equipe de desenvolvimento, composta pelos autores deste trabalho.

4.1.1.7 Atividades chave

As principais atividades a serem realizadas na organização seriam a manutenção do sistema, correção de erros, suporte ao usuário e o desenvolvimento de novas funcionalidades. Além disso, também é importante a atualização constante das redes sociais com informações sobre a empresa.

4.1.1.8 Parcerias Principais

Antes de o sistema estar comercialmente disponível, alguns produtores de leite, utilizam o *software* a fim de relatar a experiência sobre a usabilidade, além de fornecer sugestões de melhorias e relatar possíveis erros no sistema.

Em um segundo momento, quando o *software* estiver disponível para uso comercial, estes produtores continuarão a utilizar sem nenhum custo.

4.1.1.9 Custos

As principais despesas do projeto são os custos de desenvolvimento do sistema, os custos de manutenção e hospedagem do *site*.

4.2 Mapa da Empatia

Com o propósito de diagnosticar o perfil dos potenciais clientes, além de compreender suas reais necessidades, foi elaborado um mapa da empatia. Que segundo Osterwalder e Pigneur:

> O objetivo é criar um ponto de vista do cliente para questionar continuamente suas suposições sobre o Modelo de Negócios. O perfil do cliente permite dar respostas melhores para questões como: esta Proposta de Valor resolve problemas reais? Ela estaria realmente disposta a pagar por isso? Como ela gostaria de ser atendida? (OSTERWALDER; PIGNEUR, 2011, p. 133)

O mapa da empatia foi desenvolvido junto aos produtores de leite integrantes da agricultura familiar. E ilustra suas visões sobre a pecuária de leite.

Abaixo se encontra a descrições do mapa da empatia. O modelo gráfico se encontra no Anexo II.

4.2.1 O ele pensa e sente?

Embora a remuneração não seja condizente com a quantidade de trabalho que a atividade demanda. Para pequenas propriedades rurais é uma fonte de renda certa e garantida.

A atividade de produção de leite também demanda que os produtores tenham domínio em diversas áreas de conhecimento, para poder planejar, executar e administrar a propriedade.

4.2.2 O que ele vê?

O cliente viu uma melhora na remuneração do setor nos últimos anos. Embora poucas pessoas estejam interessadas em ingressar no setor, a maioria das pessoas que trabalham com pecuária de leite já estão nesse ramo há muito tempo.

4.2.3 O que ele escuta?

Ele escuta que a produção de leite é uma das melhores atividades para a agricultura familiar. Embora ouça de colegas produtores reclamações do baixo lucro, e que é preciso investir e se aperfeiçoar cada vez mais no setor.

4.2.4 O que ele fala e faz?

Ele fala que a pecuária de leite tem um bom retorno financeiro a médio e longo prazo. E tem a vantagem de ser dono do próprio negócio. Também busca trocar experiências com outros produtores com o objetivo de melhorar sua produção.

4.2.5 Qual a sua dor?

A falta de programas governamentais para incentivar a produção de leite e a agricultura de modo geral. Além da instabilidade com relação aos custos da produção, sendo que muitas vezes os ganhos não acompanham.

Ademais, não ter férias, trabalhar todos os dias, inclusive finais de semana e feriados, muitas vezes em horários pouco usuais tornam a atividade maçante.

4.2.6 Quais ganhos e desejos do cliente?

Deseja ter mais acesso as tecnologias que auxiliem a produção e o conforto animal, além do gerenciamento da propriedade, para tornar o trabalho mais administrativo e gerencial e menos braçal.

5 PROJETO

5.1 Sistema proposto

O sistema se divide em quatro módulos, denominados rebanho, ordenha, reprodução e saúde animal. Para ter acesso ao sistema, o cliente precisa efetuar um cadastro informando alguns dados pessoais. Além disso, precisa criar uma senha de acesso, além de fornecer um *e-mail* válido que será o nome de usuário.

No módulo de rebanho, o usuário pode visualizar todos os animais cadastrados no sistema, com suas informações básicas. Ao selecionar um dos animais listados é possível ver todas as suas informações, além de permitir editar ou excluir o animal.

Na parte correspondente ao módulo ordenha. O usuário pode registrar a quantidade produzida, informar a quantidade vendida e o valor recebido por litro de leite comercializado. Isso permite ao sistema gerar históricos de produção, de receitas e de preço médio comercializado. Sendo representados por gráficos mensais.

No módulo de reprodução. É possível visualizar e registrar as inseminações, os partos realizados e as gestações abortadas. Também é possível visualizar históricos de nascimentos e inseminações através de gráficos mensais.

Na parte de saúde animal, encontra-se a o histórico de tratamentos provido aos animais. Além de permitir cadastrar um novo tratamento e o agendamento de tratamentos futuros.

5.2 Estórias de usuário

O desenvolvimento do Ordenha Digital foi baseado nos requisitos coletados na forma de estórias de usuário junto aos agricultores familiares produtores de leite. Os produtores atuam no ramo agrícola no município de Ouro, localizada na região oeste de Santa Catarina.

A tabela 5 representa o *product backlog* para o do módulo de rebanho. O documento completo contendo as estórias de usuário se encontra no anexo III deste trabalho.

Tabela 7: *backlog* do módulo rebanho.

BACKLOG REBANHO				
ID	**Titulo**	**Pri**	**Estória**	**Notas**
1	Visualizar	20	Como usuário eu quero ver a data de nascimento de qualquer animal, qual o pai e a mãe dele.	
2	Foto	18	Como usuário, ao ver as informações de um animal, quero ver a foto dele.	
3	Cadastro	40	Ao cadastrar um novo animal informar: nome, a data de nascimento, progenitor e progenitora, sexo, foto, tipo do animal e brinco.	Data nasc, sexo e brinco são campos obrigatórios. Tipo do animal é novilha, vaca de leite, bezerro e reprodutor.

4	Tela	16	Listar os animais cadastrados, permitir pesquisar animal pelo brinco. Ordenar por tipo de animal para facilitar a visualização.	
5	Histórico de cruza	12	Como usuário, quero ver todas às cruzas e partos já realizados de uma vaca.	
6	Histórico de tratamento	10	Como usuário, quero ver todos os tratamentos de um animal já realizou.	

5.3 Funcionalidades gerais do sistema

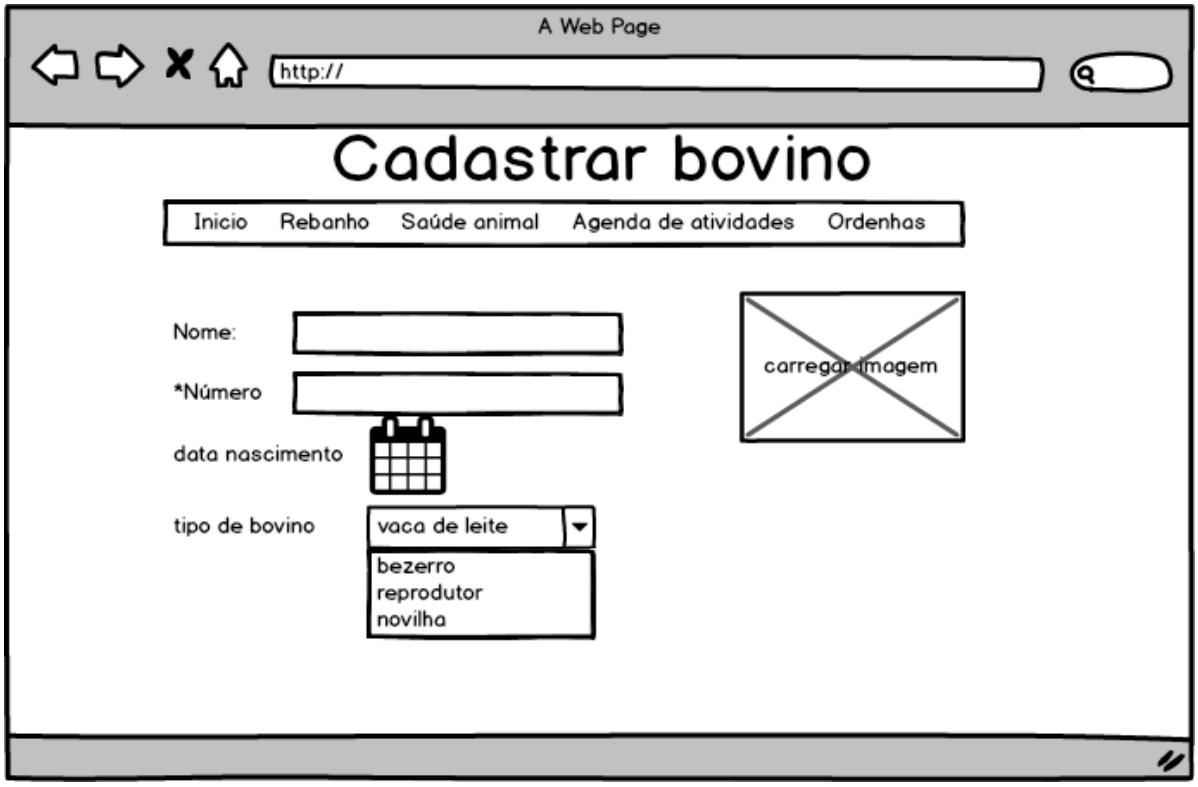
A partir das estórias de usuário, juntamente com as pesquisas feitas em outros *softwares,* como descritas no capitulo 3. Conclui-se que o Ordenha Digital deve ter as seguintes características:

- Cadastrar o rebanho.
- Controle reprodutivo estimando datas de parto.
- Histórico de partos realizados.
- Registrar ordenha.
- Manter histórico de produção e receitas.
- Manter histórico de tratamentos do rebanho.
- Agendar tratamento dos animais.

5.4 Protótipos de tela.

Após a coleta das estórias de usuário, foram desenvolvidos protótipos de tela visando aperfeiçoar o desenvolvimento da interface gráfica do sistema. Os protótipos foram desenvolvidos com auxilio da ferramenta Balsamiq Mockups.

Na tela abaixo se encontra um protótipo da tela de cadastro de animal. Os demais protótipos se encontram no anexo IV.



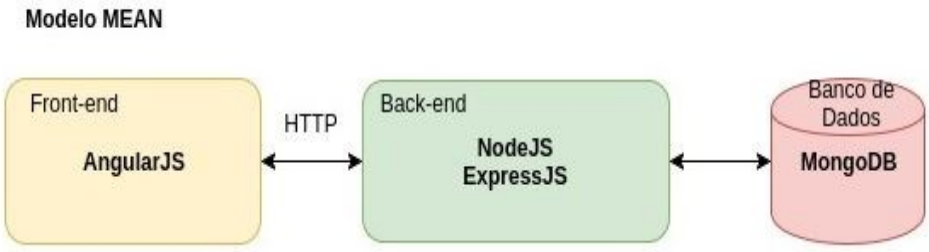
Tela 5: protótipo da tela de cadastro de animal.

6 DESENVOLVIMENTO

O Ordenha Digital foi desenvolvido como uma plataforma *online*, possibilitando aos produtores utilizarem o sistema via diversos aparelhos que não só um único *smartphone*, podendo estes acessar seus registros e dados de qualquer dispositivo conectado à rede. Desta maneira o usuário poderá utilizar seu *smartphone* para registrar um parto diretamente no local de tal e depois visualizar seus dados com mais conforto em seu *desktop* dentro de casa, por exemplo. Esta experiência só é possível devido à expansão do acesso à internet no meio rural, como citado anteriormente no capítulo 4. Também pensando no futuro a centralização dos dados, consequência de um sistema *online*, possibilita que o Ordenha Digital possa emitir relatórios gerais de produção de leite de uma determinada região, ou de uma cooperativa, bem como disponibilizar um módulo de troca de informação entre produtores.

6.1 Arquitetura geral do sistema

O Ordenha Digital foi desenvolvido utilizando a pilha de ferramentas JavaScript para desenvolvimento de aplicações *web* MEAN (acrônimo para *MongoDB, ExpressJS, AngularJS e NodeJS*), num modelo MVC (*Model, View e Controller*).



Tela 6: Modelo geral

Fonte: Elaboração dos autores

6.2 Banco de dados

O banco de dados utilizado no Ordenha Digital é o MongoDB, seguindo a arquitetura da pilha de desenvolvimento *web* MEAN, como citado anteriormente neste capítulo.

De acordo com *Mean Full stack JavaScript* para aplicações *web* com MongoDB, Express, Angular e Node (ALMEIDA, 2015), MongoDB permite armazenar e reaver um formato muito similar ao JSON (*JavaScript Object Notation*) e é um banco de dados fracamente tipado, deixando as responsabilidades de validação nas mãos dos desenvolvedores. Isto é uma vantagem, pois deixa o esquema do banco de dados muito maleável, controlado totalmente pelo *Model* da aplicação, liberando o desenvolvedor para que ele foque mais no negócio. O Ordenha Digital utilizou o módulo Mongoose do NodeJS para acesso e controle do banco de dados.

Mongoose é um módulo responsável pela comunicação com o banco de dados, MongoDB. É baseado em esquemas e permite modelares os dados da

aplicação. Possui um sistema de conversão de tipos, validação de dados, criação de consultas e *hooks* para a lógica de negócios.

6.3 Back-end

O *back-end* foi escrito em JavaScript, aproveitando o ecossistema em torno da linguagem que já é muito difundida e segundo (ALMEIDA, 2015) continua a evoluir, através da plataforma NodeJS. Segundo (ALMEIDA, 2015) NodeJS age como um *core* da aplicação e disponibiliza diversos módulos escritos pela própria comunidade de desenvolvedores NodeJS através de seu gerenciador de pacotes *npm*.

O módulo responsável por receber e responder as requisições vindas do *front-end*, como sugere a pilha MEAN, é o ExpressJS. "*Express*, criado em 2009 por TJ Holowaychuk, é um *framework web light-weight* que ajuda na organização de sua aplicação *web* na arquitetura MVC." (ALMEIDA, 2015). Juntamente com o ExpressJS outro módulo, Body-parser é responsável por lidar com o *stream* de dados vindos ao corpo das requisições HTTP do tipo *POST* e *PUT*. Ele atua como um *middleware* entre as requisições que chegam ao servidor e o efetivo tratamento delas pelo *back-end*, facilitando a leitura do corpo das requisições pelo mesmo.

O esquema do servidor consiste em *Models*, *Controllers* e *Routes*.

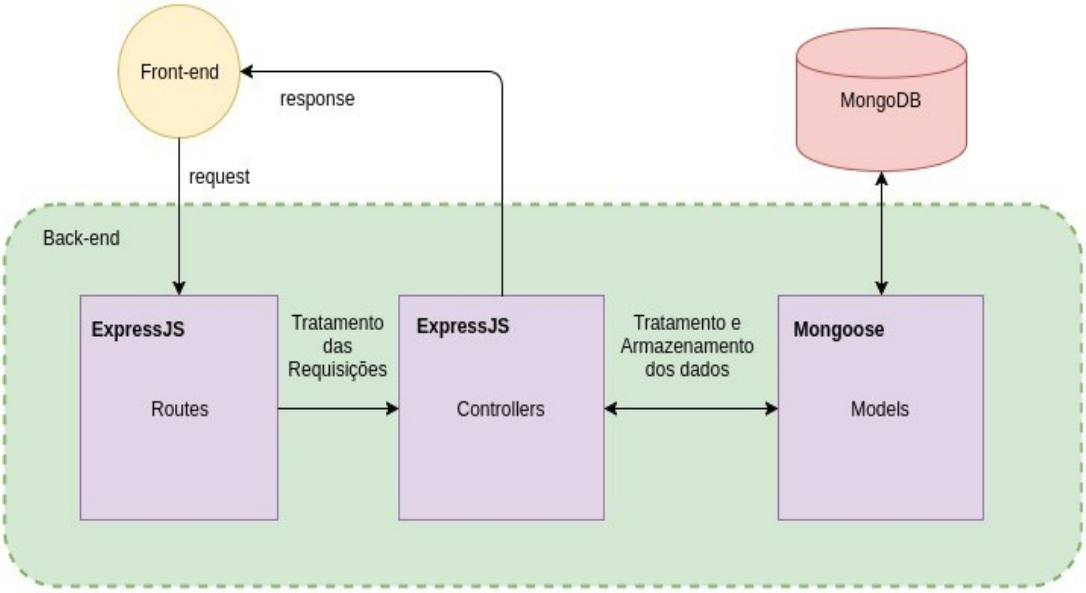
As *Routes* são como o próprio nome sugere as rotas de comunicação entre o *front-end* e o *back-end*, nelas está contido as URLs e os verbos HTTP correspondentes a cada requisição. Nos *Routes* também são instanciados os *Controllers*, do qual terão suas instâncias atreladas a cada uma dessas rotas.

Os *Controllers*, por sua vez, são responsáveis por tratar o conteúdo das requisições e dar uma resposta ao *front-end*, são nos *Controllers* que cada *Model* é instanciado, para que os tratamentos das requisições se comuniquem com o banco de dados.

Os *Models* são *schemas* do Mongoose (módulo *npm* para comunicação com o MongoDB), cada Model representa uma coleção de dados no MongoDB, neles estão contidos os nomes, tipos e atribuições de cada campo do

documento que será armazenado. São eles que fazem a comunicação direta com o banco de dados e a instância de cada um contém métodos para tal.

No Ordenha Digital cada segmento do sistema (autenticação, usuário, rebanho, ordenha, reprodução e saúde animal) possui sua trilha de *Model*, *Controller* e *Route* específica (exceto autenticação, que possui somente *Route* e *Controller*, pois não necessita de um esquema no banco de dados).



Tela 7: Modelo back-end

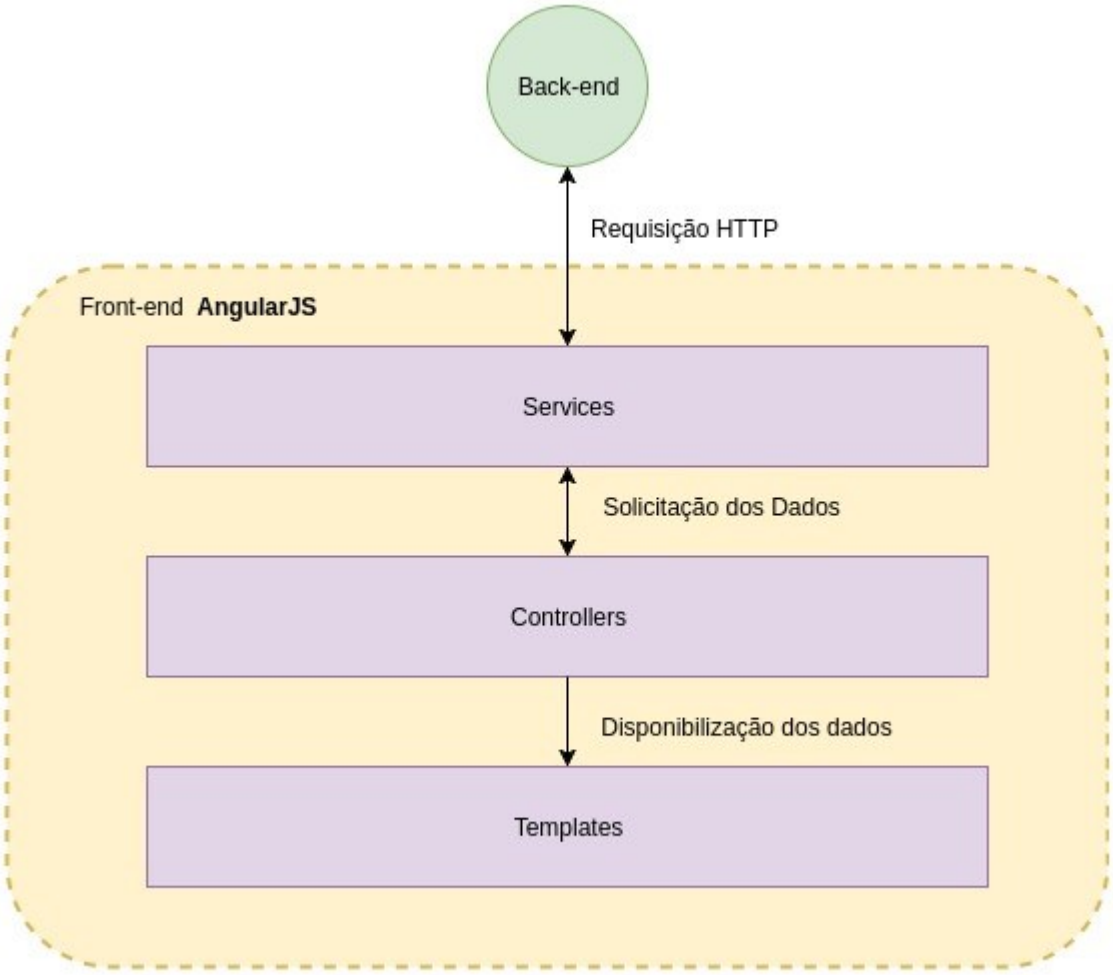
Fonte: Elaboração dos autores

6.4 Front-end

No *front-end* a ferramenta da pilha de desenvolvimento *web* MEAN, utilizado neste sistema, é o AngularJS.

“AngularJS é um *framework* MVC no lado do cliente voltado para *Single Page Web Applications* (SPWA) criado pela Google e liberado ao público em 2009” (ALMEIDA. 2015). Ainda segundo (ALMEIDA. 2015) AngularJS facilita o recebimento de dados e execução da lógica diretamente no cliente, diferente da maneira tradicional na qual dado e lógica são processados no lado do servidor, e é o responsável por interfaces dinâmicas sem manipulação do DOM.

A engenharia do cliente consiste em *Controllers*, *Services* e *Templates*. Desta maneira os *Controllers* solicitam dados aos *Services*, que por sua vez se comunicam com o servidor através de requisições HTTP, e disponibilizam estes dados para os *Templates*, que renderizam estes dados na tela. Cada *template* possui um *Controller* específico e os *Services* foram programados um para cada segmento do sistema (autenticação, usuário, rebanho, ordenha, reprodução e saúde animal).



Tela 8: Modelo front-end

Fonte: Elaboração dos autores

6.5 Interface com o usuário

A interface com o usuário foi desenvolvida utilizando Bootstrap, um framework para *design* de interfaces de *websites* e aplicações *web* que utiliza HTML, CSS e JavaScript como ferramentas.

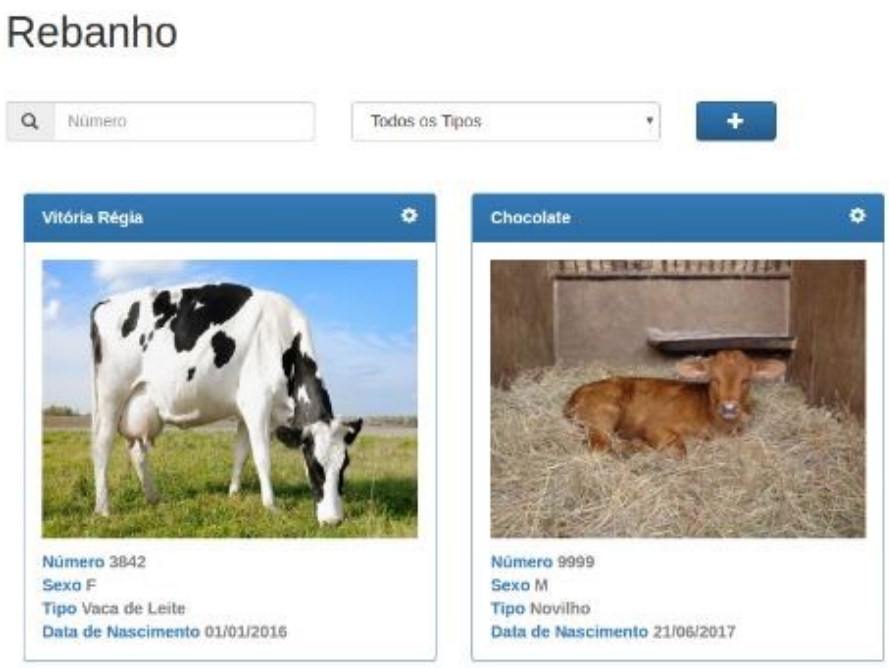
A interface é dividida em quatro módulos, Rebanho, Ordenha, Reprodução e Saúde Animal. Além disso, usa *design* responsivo, ou seja, se adapta para o uso em diferentes tamanhos de telas para se ajustar em diversos dispositivos como em *desktops*, *smartphones* e *tablets*.

A tela a seguir mostra a tela inicial do sistema, logo após o usuário acessar o sistema. Nela estão contidos os dados de usuário e número de animais no rebanho, ao lado esquerdo, bem como os eventos pendentes na semana, lado direito.

Tela 9: Tela inicial do sistema.

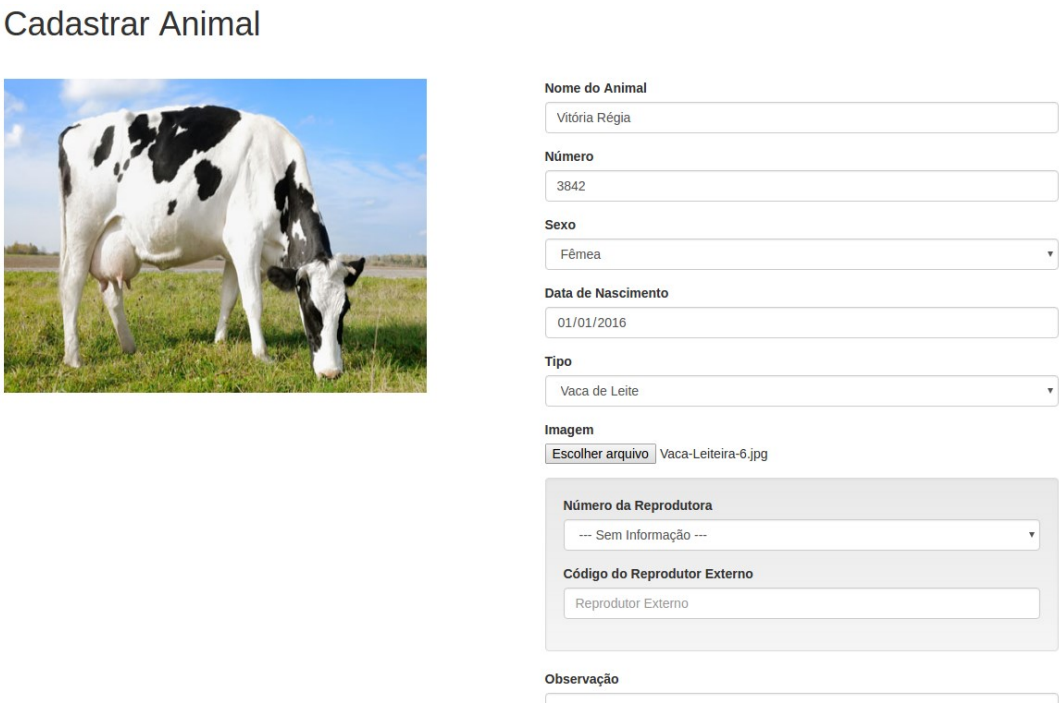
As próximas telas são as telas de rebanho.

Na tela principal consegue-se obter uma visão geral do rebanho, esta tela contém um filtro por tipo de animal e número do cadastro e traz as informações básicas de cada animal.



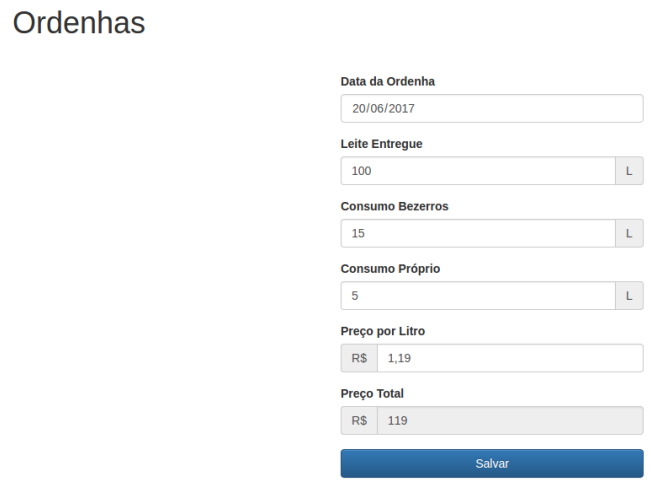
Tela 10:Tela inicial do módulo rebanho.

Na tela de cadastro de rebanho estão contidos os dados do animal a ser cadastrado. Nome, número, sexo, data de nascimento, tipo, imagem, dados do reprodutor e uma observação que pode conter a raça do animal, por exemplo, dentre outros dados relevantes para cada produtor.



Tela 11:Cadastrar novo animal.

Após o módulo de rebanho temos o módulo de ordenha, neste módulo é possível cadastrar a quantidade de leite entregue ao distribuidor, por dia, e o preço praticado por litro em cada entrega. O sistema calcula o preço médio por litro, a quantidade de leite entregue e a receita que o produtor obteve com a venda do leite, ambos por mês, e disponibiliza esta informação via gráfico, para interfaces no padrão *desktop*, e via tabela, para interfaces nos padrões *smartphone* e *tablet*.

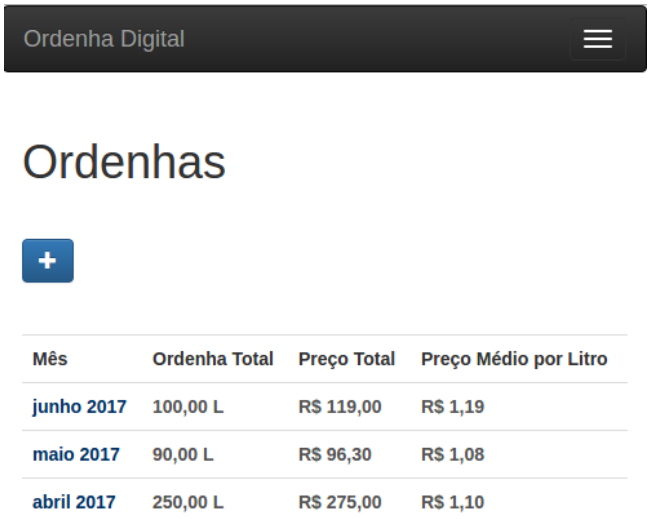


Tela 12:Cadastrar nova ordenha.

Tela 13:Relatório de ordenhas realizadas no mês.

Tela 14:Relatório de recitas mensais.

Tela 15: Relatório de preço comercializado

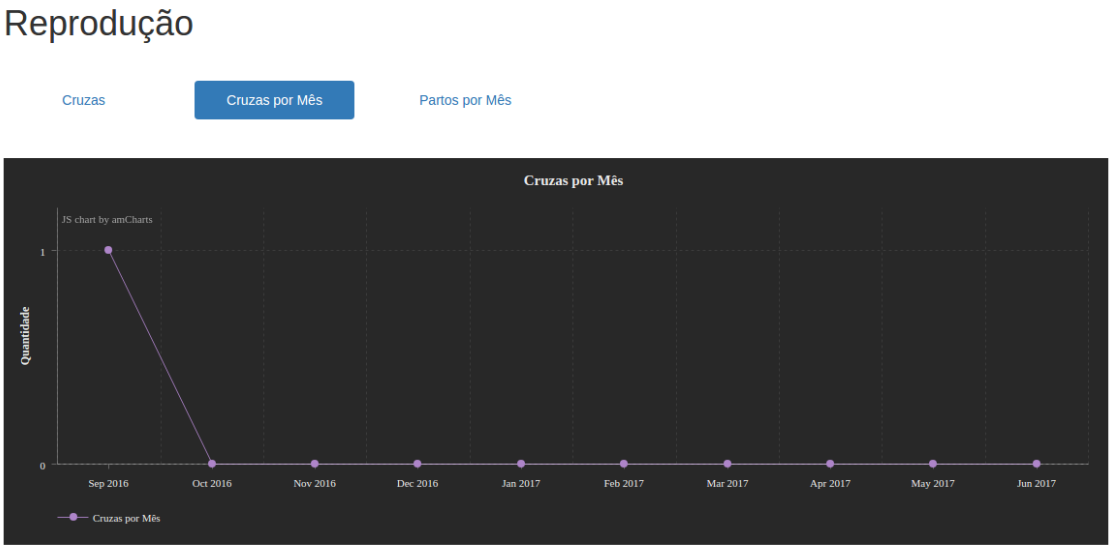


Tela 16:Relatório das ordenhas em forma de lista para *smartphone.*

O módulo seguinte é o de reprodução, nele cadastramos uma nova cruza informando o nome da matriz reprodutora, a data da cruza e informações sobre o reprodutor externo, como código, nome e uma observação a cargo do usuário.

Além disso, é possível visualizar as cruzas já cadastrada através de tabela, com os *status* de “Em Gestação”, “Parto Realizado” e “Gestação Abortada”. A partir da data da cruza o sistema calcula 9 meses exatos, período de gestação do bovino, permitindo que esta data seja alterada no dia que o parto for realizado. O sistema também traz em formato de gráficos o número de cruzas e o número de partos realizados, ambos por mês.

Tela 17:Tela inicial do módulo reprodução.

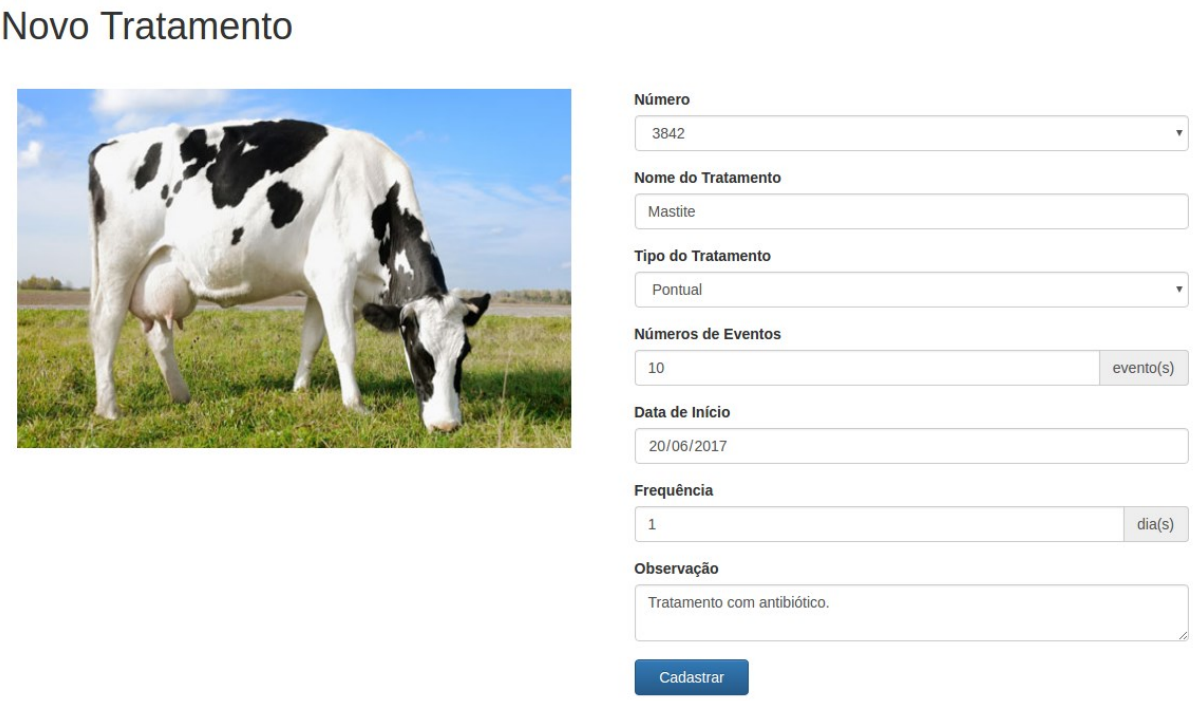


Tela 18:Relatório de cruzas realizadas.

O último módulo é o de saúde animal, é nele que estão contidos os tratamentos de cada animal do rebanho.

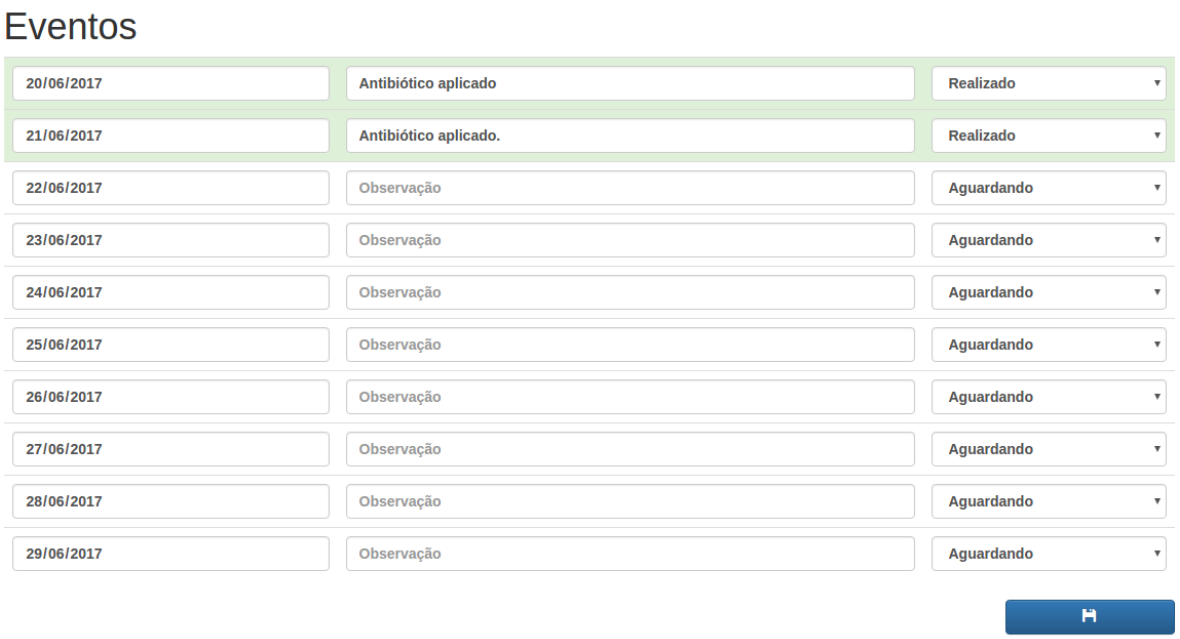
Este módulo é subdividido em tratamentos contínuos e tratamentos pontuais. Sendo o primeiro tratamentos que durarão um longo período ou a vida inteira do animal, como tratamento de carrapatos, por exemplo, e o segundo tratamentos que tem uma duração menor, geralmente para tratar doenças que podem aparecer.

No cadastro do tratamento é informado o tipo do tratamento bem como seu nome, o número do animal a ser tratado, o número de eventos (somente em caso de tratamento contínuo) a data de início, a frequência com os quais os eventos deste tratamento acontecerão e uma observação sobre tal que fica a carga do produtor.



Tela 19:Registrar tratamento.

Estes tratamentos geram eventos, cada qual com sua data pré-calculada pelo sistema.



Tela 20:Lista de tratamentos pré-agendados ou realizados.

7 CONCLUSÃO

O desenvolvimento do Ordenha Digital mostrou-se bastante plausível, pois existe uma demanda de mercado muito grande a ser explorado. Isso é percebido nos *softwares* pesquisados, sendo todos eles relativamente novos no mercado. Muito em virtude da popularização da internet no campo, que apresentou um amplo crescimento nos últimos anos.

Além disso, as projeções de crescimento da pecuária de leite para os próximos anos são significativas. Sendo a busca pela melhora da produtividade que é muito baixa no Brasil, um dos fatores que podem contribuir para o uso de *softwares* ou outras ferramentas de gestão.

7.1 Trabalhos futuros

O sistema desenvolvido neste trabalho apresenta diversas possibilidades de melhorias. A seguir algumas sugestões que poderiam ser realizadas de forma a contribuir a melhoria do Ordenha Digital.

- Desenvolvimento de uma plataforma de comércio entre os produtores, possibilitando a comercialização de seu rebanho.
- Desenvolver uma funcionalidade que consista no cadastro de várias fazendas vinculadas a uma mesma conta de usuário. Sendo útil para cooperativas acompanharem as propriedades de seus cooperados.

8 REFERÊNCIAS

ALEXANDER, Osterwalder; PIGNEUR, Yves. **Business Model Generation**: Inovação em Modelos de Negócios: um manual para visionários, inovadores e revolucionários. Rio de Janeiro: Alta Books, 2011

ALMEIDA, Flávio. **Mean Full stack JavaScript para aplicações web com MongoDB, Express, Angular e Node**. São Paulo: Casa do Código, 2015.

ALVES, Daniela Rodrigues (Ed.). INDUSTRIALIZAÇÃO E COMERCIALIZAÇÃO DO LEITE DE CONSUMO NO BRASIL. In: MADALENA, Fernando Enrique; MATOS, Leovegildo Lopes de; HOLANDA JÚNIOR, Evandro Vasconcelos (Ed.). **Produção de leite e sociedade: uma análise crítica da cadeia do leite no Brasil.** Belo Horizonte: Fepmvz, 2001. p. 75-83.

Comitê Gestor da Internet. **TIC DOMICÍLIOS 2015: Pesquisa Sobre o uso Das Tecnologias de Informação e Comunicação nos Domicílios Brasileiros**. São Paulo: no Brasil, 2016. Disponível em: <http://cetic.br/pesquisa/domicilios/publicacoes>. Acesso em 15 abril de 2017.

Department of Agriculture. **Cows milk production and consumption: summary for selected countries**. In: ESTADOS UNIDOS. PSD online production, supply and distribution. Washington, DC: USDA, 2015. Disponível em: <http://apps.fas.usda.gov/psdonline/>. Acesso em: 01/01/2016.

DEPONTI, C. M. **As "Agruras" da gestão da propriedade rural pela agricultura familiar.** REDES – Rev. Des. Regional, Santa Cruz do Sul, v. 19, Ed. especial, p. 9.24, 2014. Documento eletrônico disponível em: <http://online.unisc.br/seer/index.php/redes/article/view/5150/3555>. Acesso em: 12 maio 2016.

Embrapa: Empresa brasileira de pesquisa e agropecuária. **Leite em números. 2017.** Disponível em: <http://www.cileite.com.br/content/leite-em-n%C3%BAmeros-produ%C3%A7%C3%A3o>. Disponível em Acesso em 02/10/2016.

EMBRAPA. **GisLeite - Guia do Usuário.** 2016. Disponível em: <http://gisleite.cnpgl.embrapa.br/documentos/guia_usuario_gisleite.pdf>. Acesso em: 22 de maio de 2017.

Embrapa Gado de Leite e SEAPA-MG 2016. **Leite em números**. Disponível em <http://www.cileite.com.br/content/leite-em-n%C3%BAmeros-produ%C3%A7%C3%A3o> Acessado em 02/10/2016.

EPAGRI. **Números da Agropecuária Catarinense**. Florianópolis: Estado de Santa Catarina, 2015. 70 p. Disponível em: <http://docweb.epagri.sc.gov.br/website_cepa/publicacoes/Numeros_agropecuarios_2015.pdf>. Acesso em: 06 jul. 2016.

FAO: Food and Agriculture Organization of United Nations. **O que é agricultura familiar?** Documento eletrônico disponível em: <http://www.fao.org/family-farming-2014/home/what-is-family-farming/pt/> Acesso em: 12/06/2016.

IBGE - Instituto Brasileiro de Geografia e Estatística**. CENSO AGROPECUÁRIO 2006: Brasil, Grandes Regiões e Unidades da Federação.** Rio de Janeiro: 2006. Disponível em: <http://biblioteca.ibge.gov.br/visualizacao/periodicos/51/agro_2006.pdf>. Acesso em: 02 set. 2016.

_____Instituto Brasileiro de Geografia e Estatística - **Produção da Pecuária Municipal**. Rio de Janeiro, v. 42, p.1-39, 2014.

_____Instituto Brasileiro de Geografia e Estatística - **Produção da Pecuária Municipal**. Rio de Janeiro, v. 28, p.1-28, 2000.

_____:Instituto Brasileiro de Geografia e Estatística - Produção da Pecuária Municipal. Rio de Janeiro, v. 43, p.1-49, 2015

MAPA - Ministério da Agricultura, Pecuária e Abastecimento. **Plano Mais Pecuária.** 2014. Disponível em: <http://www.agricultura.gov.br/arq_editor/file/Ministerio/Publicacao_v2.pdf> Acesso em 01/10/2016

_____. Ministério da Agricultura, Pecuária e Abastecimento. **Plano Mais Pecuária.** 2014. Disponível em:

<http://www.agricultura.gov.br/arq_editor/file/Ministerio/Publicacao_v2.pdf> Acesso em 01/10/2016

_____. Pecuária e Abastecimento. Leite 100 - **Programa de Fortalecimento da Atividade leiteira na Agricultura Familiar**. Disponível em: <http://www.agricultura.gov.br/arq_editor/file/camaras_setoriais/Leite_e_derivados/36RO/App_EMBRAPA_Leite.pdf> Acesso em 18 de junho de 2015.

Portal Brasil: **Governo quer ampliar produção de leite em 40% em 10 anos**. Disponível em: <http://www.brasil.gov.br/economia-e-emprego/2014/02/governo-quer-aumentar-producao-de-leite-em-40-em-10-anos> Acessado em 15/09/2016

_____. **Agricultura familiar produz 70% dos alimentos consumidos por brasileiro**. Documento eletrônico disponível em:<http://www.brasil.gov.br/economia-e-emprego/2015/07/agricultura-familiar-produz-70-dos-alimentos-consumidos-por-brasileiro>. Acessado em 28/08/2016.

Portal Embrapa. Disponível em: <https://www.embrapa.br/busca-de-produtos-processos-e-servicos/-/produto-servico/1430/gisleite---gestao-de-sistemas-de-producao-de-leite> Acesso em 20 de maio de 2017.

_____. Disponível em: < https://www.embrapa.br/busca-de-noticias/-/noticia/15562960/gestao-do-rebanho-leiteiro-ganha-aplicativo>. Acesso em 24 maio de 2017.

Portal Gisleite. Disponível em: <http://gisleite.cnpgl.embrapa.br/lista.php?type=producao_leite> Acesso em 10 de novembro de 2017.

_____. Disponível em:<http://gisleite.cnpgl.embrapa.br> Acesso em 20 de maio de 2017.

Portal 4milk. Disponível em: <http://www.4milk.com.br/Home> Acesso em 10 de março de 2017.

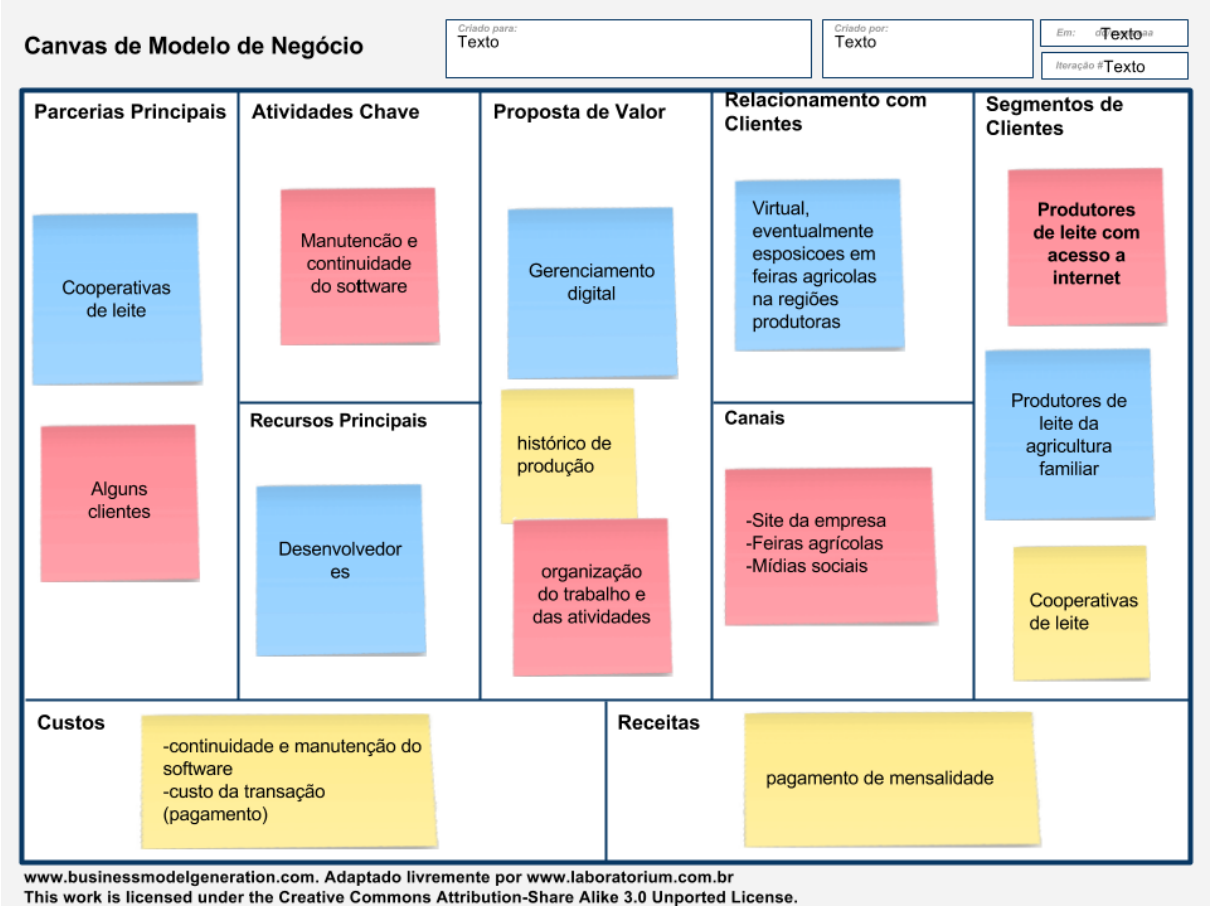
SEBRAE. Canvas: **Melhoria na Competitividade**. Disponível em: <http://www.sebrae.com.br/sites/PortalSebrae/bis/quadro-de-modelo-de-negocios-para-criar-recriar-e-inovar,a6df0cc7f4217410VgnVCM2000003c74010aRCRD>. Acesso em: 06 de junho de 2017.

TALIARINE, Adriana. **A importância da gestão no agronegócio brasileiro**. Revista Perspectiva em Gestão, Educação E Tecnologia v.4 n.8, julho-dezembro/2015.

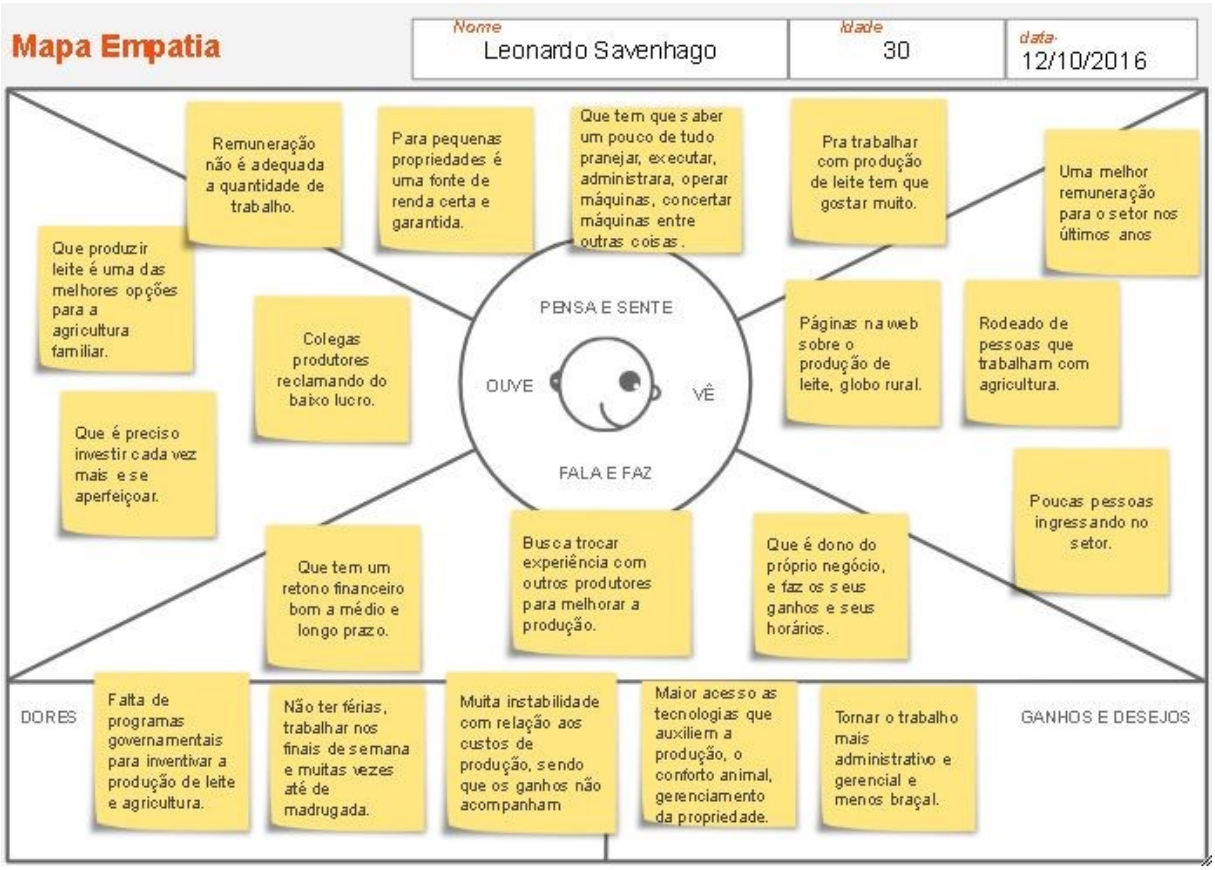
ZOOCAL, R. **Produtores de leite: quantos são?** 2012. Disponível em: <http://www.repileite.com.br/forum/topics/produtor-de-leite-quantos-s-o>. Acesso em: 12/06/2016.

ZOCCAL, Rosangela; DUSI, Geraldo. **Modelo ideal para produção de leite no Brasil**. Documento eletrônico disponível em: <http://www.sna.agr.br/uploads/AnimalBusiness_09_34.pdf>

ANEXO I – CANVAS



ANEXO II – MAPA DA EMPATIA



ANEXO III – ESTÓRIAS DE USUÁRIO

BACKLOG SAÚDE ANIAMAL				
ID	**Titulo**	**Pri**	**Estória**	**Notas**
1	Agendar	15	Quero ser lembrado de tratamentos agendados. Além de poder agendar tratamentos futuros.	Avisar na tela inicial
2	Histórico		Quero visualizar todas as doenças e tratamentos que um animal já teve	
3	Tratamento		Quero registrar os tratamentos realizados e as doenças tratadas.	
4	Tela Saúde animal		Na tela de saúde animal, separar os que são pontuais (exceção) e os contínuos.	
5	Tela de tratamento		O usuário pode marca a opção de tratamento realizado ou não	

BACKLOG REPRODUÇÃO				
ID	**Titulo**	**Pri**	**Estória**	**Notas**
1	Inseminação	15	Como usuário, quero saber quais vacas foram inseminadas e em quais datas. Para monitorar retorno de cio e saber a data de parto.	
2	Inseminação	12	Como usuário, quando inseminar uma vaca desejo registrar o nome do reprodutor.	Tanto reprodutor interno quanto externo
3	Reprodutor	11	O reprodutor tem um código fornecido quando o sêmen é comprado. Que identificar qual o touro e de qual fazenda procede.	Permitir que este código fosse inserido na forma de identificador.
4	Parto	20	Como usuário, quero saber a data do	

			parto de cada vaca.	

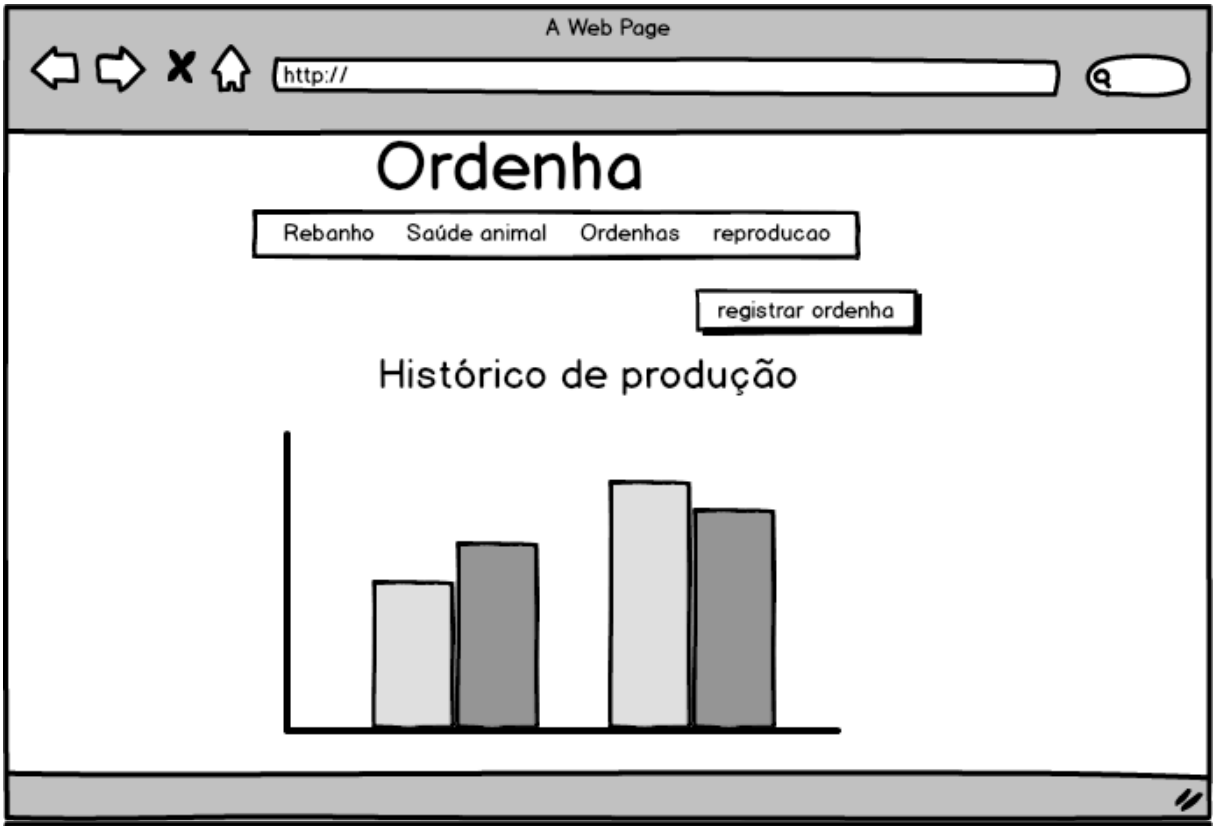
BACKLOG REBANHO				
ID	**Titulo**	**Pri**	**Estória**	**Notas**
1	Visualizar	20	Como usuário eu quero ver a data de nascimento de qualquer animal, qual o pai e a mãe dele.	
2	Foto		Como usuário, ao ver as informações de um animal, quero ver a foto dele.	
3	Cadastro		Ao cadastrar um novo animal informar: nome, a data de nascimento, progenitor e progenitora, sexo, foto, tipo do animal e brinco.	Data nasc, sexo e brinco são campos obrigatórios. Tipo do animal é novilha, vaca de leite, bezerro e reprodutor.
4	Tela		Listar os animais cadastrados, permitir pesquisar animal pelo brinco. Ordenar por tipo de animal para facilitar a visualização.	
5	Histórico de cruza		Como usuário, quero ver todas às cruzas e partos já realizados de uma vaca.	
6	Histórico de tratamento		Como usuário, quero ver todos os tratamentos de um animal já realizou.	

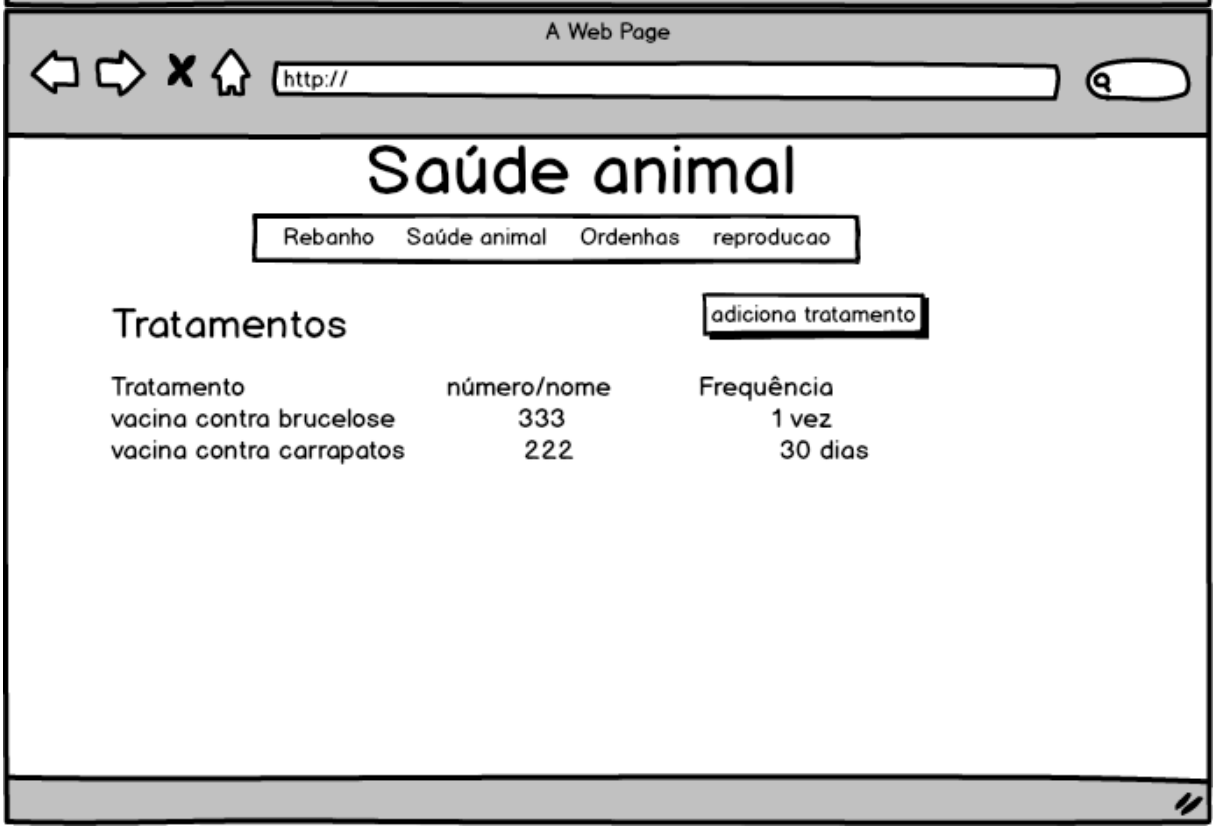
.

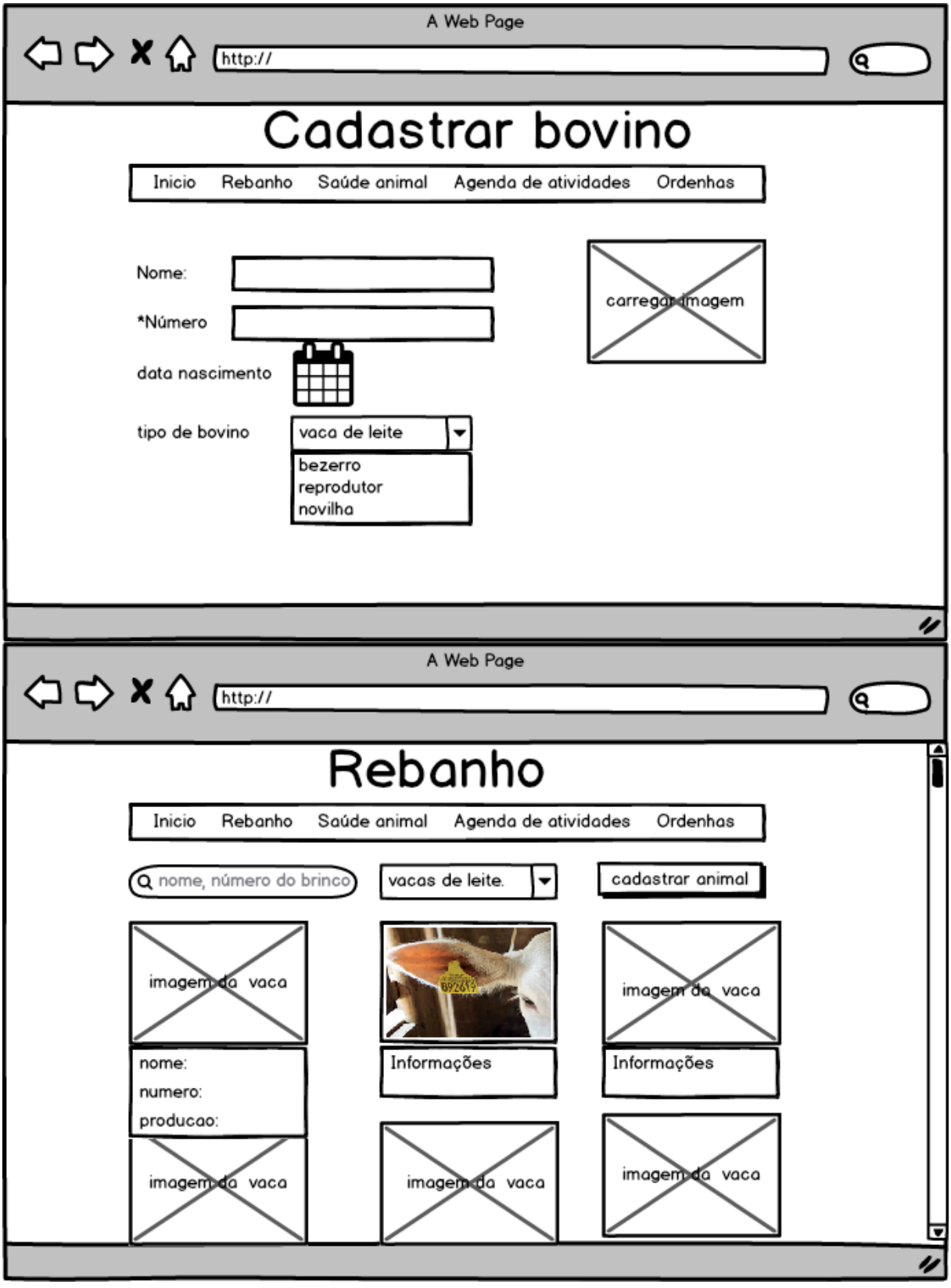
BACKLOG ORDENHA				
ID	**Titulo**	**Pri**	**Estória**	**Notas**
1	Produção	20	Como usuário, quero ver a quantidade de leite produzido.	
2	Compara		Como usuário, quero comparativos	

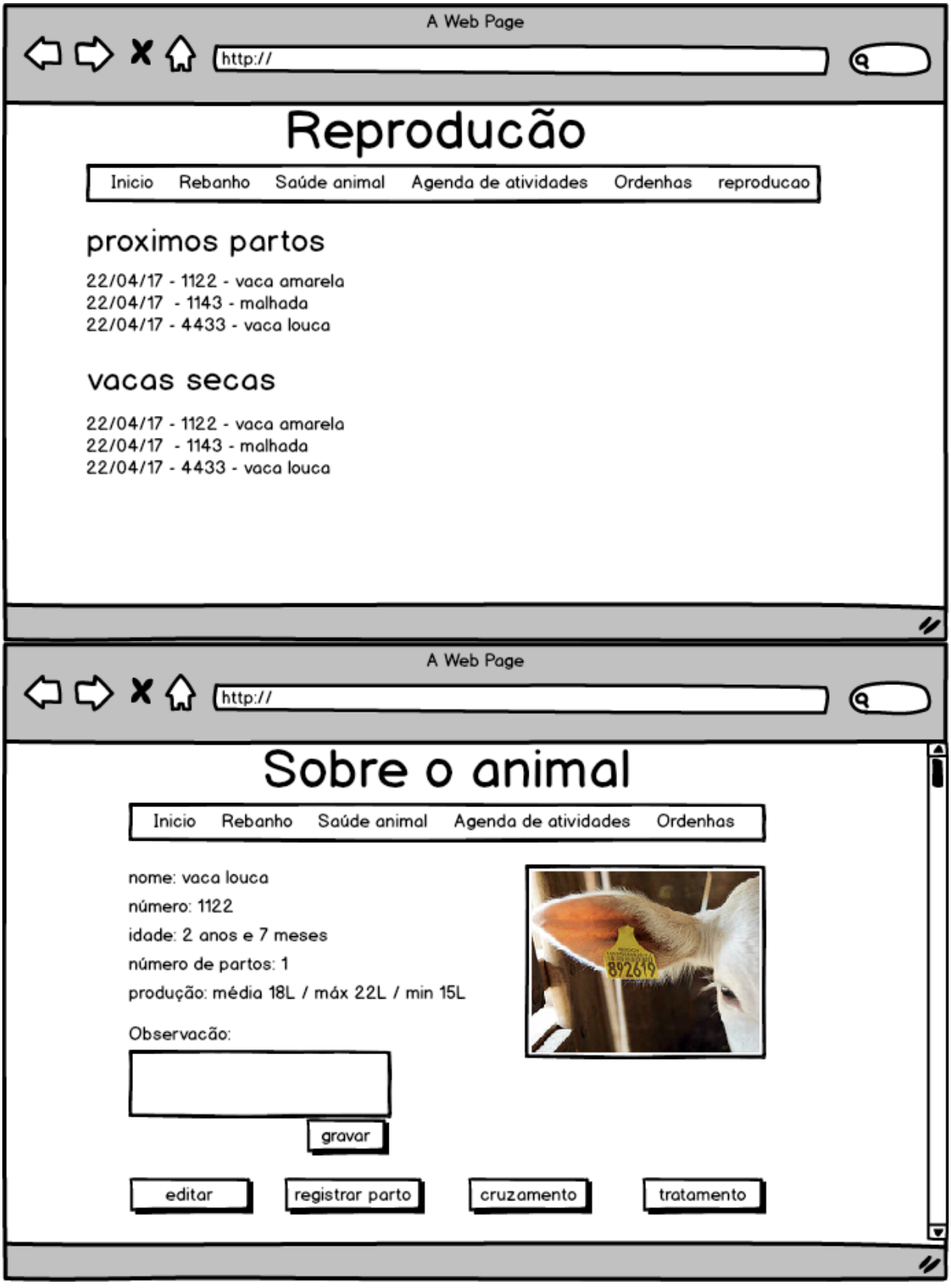
	tivos		de produção mensais.	
3	Histórico		Como usuário, quero saber a quantidade de leite produzido e a quantidade vendida.	
4	Registro		Ao registrar uma ordenha informar data, quantidade vendida, usada para consumo próprio, e na alimentação de bezerros.	

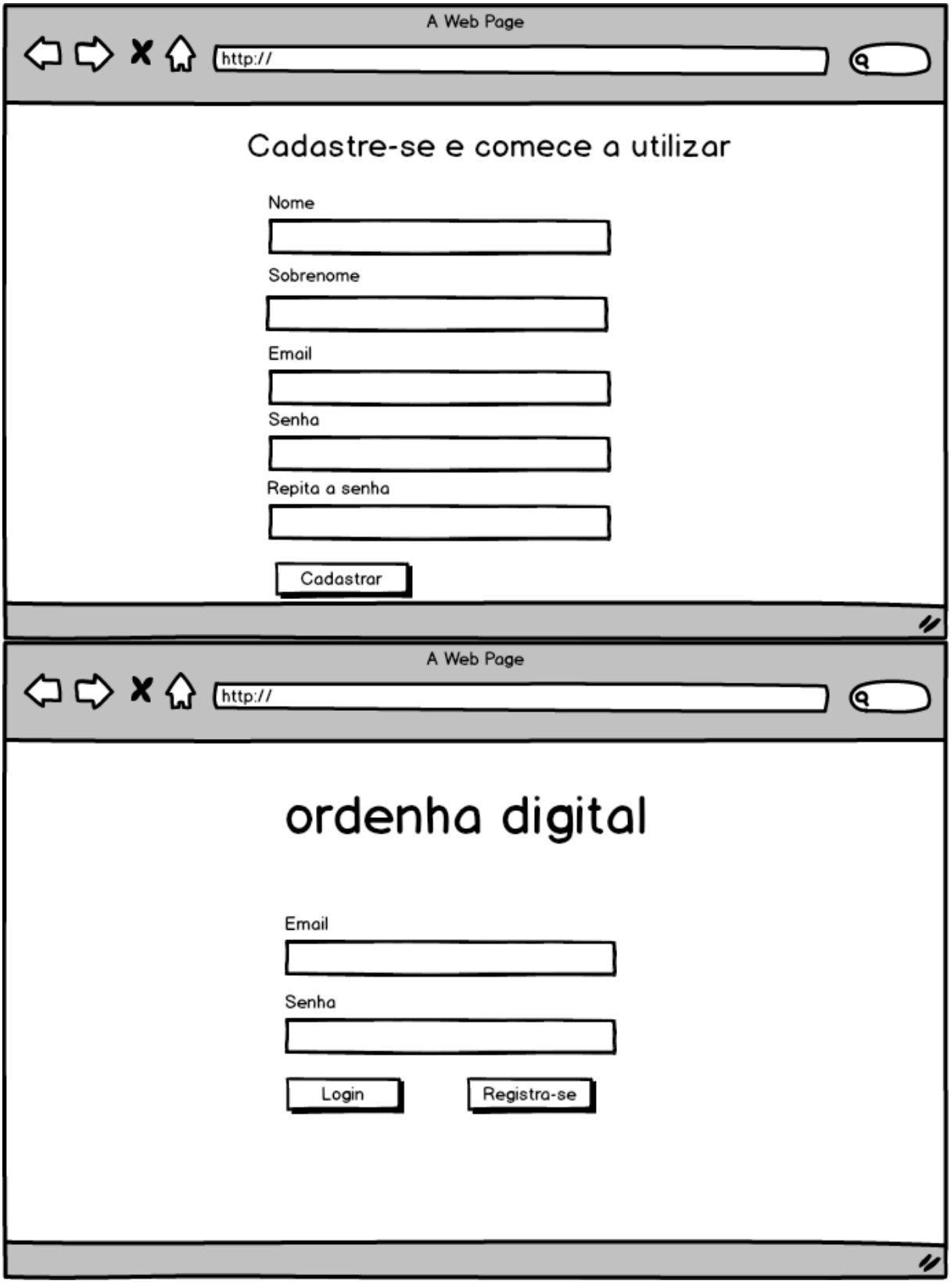
ANEXO IV – PROTÓTIPOS DE TELA






A Web Page
http://
Cadastrar bovino
Inicio
Rebanho
Saúde animal
Agenda de atividades
Ordenhas
Nome:
*Número
data nascimento
tipo de bovino
vaca de leite
bezerro
reprodutor
novilha
carregar imagem
A Web Page
http://
Rebanho
Inicio
Rebanho
Saúde animal
Agenda de atividades
Ordenhas
nome, número do brinco
vacas de leite.
cadastrar animal
imagem da vaca
nome:
numero:
producao:
imagem da vaca
Informações
imagem da vaca
Informações
imagem da vaca
imagem da vaca


A Web Page
http://
Reprodução
Inicio Rebanho Saúde animal Agenda de atividades Ordenhas reproducao
proximos partos
22/04/17 - 1122 - vaca amarela
22/04/17 - 1143 - malhada
22/04/17 - 4433 - vaca louca
vacas secas
22/04/17 - 1122 - vaca amarela
22/04/17 - 1143 - malhada
22/04/17 - 4433 - vaca louca
A Web Page
http://
Sobre o animal
Inicio Rebanho Saúde animal Agenda de atividades Ordenhas
nome: vaca louca
número: 1122
idade: 2 anos e 7 meses
número de partos: 1
produção: média 18L / máx 22L / min 15L
Observacão:
gravar
editar
registrar parto
cruzamento
tratamento


A Web Page
http://
Cadastre-se e comece a utilizar
Nome
Sobrenome
Email
Senha
Repita a senha
Cadastrar
A Web Page
http://
ordenha digital
Email
Senha
Login
Registra-se


A Web Page
http://www.leiteleite.com
jhonfaccin@gmail.com
Ordenha Digital
rebanho
ordenha
atividades
saúde animal

APENDICE I – ARTIGO

Software para Gerenciamento e Controle da Produção de Leite

Daniel M. Coelho, Jhonatan E. Faccin

Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)

Florianópolis – SC– Brasil

danielcoelho.esk@gmail.com, jhonfaccin@gmail.com

Abstract. *The milk livestok is one of the most importants Brazilian agribusiness activities. Being mainly practiced in family farming properties.*

In Brazilian software market there are some available systems to management of milk production in rural properties. Observing the few and poor features of this systems, along with the milk producers' needs, beyond the growth of information in the countryside, originates the proposal of Ordenha Digital.

Ordenha Digital is a software developed in this project in order to help milk producers, especially producers from family farming, to manage your activities.

Resumo. *A pecuária de leite é uma das atividades mais importantes no agronegócio brasileiro. Sendo praticada principalmente nas propriedades rurais pertencentes à agricultura familiar.*

No mercado de software brasileiro, existem alguns sistemas disponíveis para o gerenciamento da produção de leite nas propriedades rurais. Observando as más funcionalidades destes sistemas, juntamente com as necessidades dos produtores de leite, além do crescimento da informatização do meio rural. Originou-se a proposta do Ordenha Digital.

Ordenha Digital é um software desenvolvido neste trabalho. Com o intuito de auxiliar os produtores de leite. Em especial os produtores da agricultura familiar, a gerenciarem a atividade.

1. Introdução

A pecuária no Brasil tem sua origem ligada à exploração do gado trazido durante o período da colonização portuguesa. Inicialmente como força de trabalho e pecuária de corte. Até meados do século XIX o leite tinha papel secundário e a pouca disponibilidade do produto impediu que o consumo se tornasse um hábito naquela época. Com o passar dos anos, diversos desenvolvimentos tecnológicos permitiram novos sistemas de embalagens, tratamentos térmicos, melhores sistemas de transporte, possibilitaram que o produto chega-se até o consumidor em ótimas condições de consumo e com maior durabilidade (ALVES, 2001).

A bovinocultura de leite é uma das atividades mais praticadas no agronegócio brasileiro. Segundo o último censo agropecuário realizado em 2006, existem no Brasil

aproximadamente 5,2 milhões de estabelecimentos rurais, dos quais, cerca de 1,3 milhões tem produção de leite, ou seja, um a cada quatro estabelecimentos (MAPA, 2014).

A pecuária de leite tem mostrado um avanço na produção muito significativo. Apresentando um crescimento de mais de 75% no período entre 2000 a 2014 (IBGE, 2014). Porém esse crescimento se deve muito mais pelo aumento no número de vacas ordenhadas do que pelo aumento da produtividade (MAPA, 2014). Observando os dados da produtividade do rebanho brasileiro em 2012, a média de produção por vaca era de 1,42 toneladas de leite ao ano. Esse número é muito abaixo do encontrado em outros países tradicionais na produção de leite. Nos Estados Unidos, por exemplo, a produtividade é de 9,48 toneladas/ano (EMBRAPA, 2017). As principais causas dessa baixa produtividade incluem a utilização de animais sem aptidão ou potencial genético apropriado para a produção de leite, manejo alimentar, sanitário e reprodutivo inadequado, pouca ou nenhuma assistência técnica, além do baixo nível de escolaridade dos produtores, o que dificulta a utilização de tecnologias disponíveis (MAPA, 2014).

Segundo o Portal Brasil, a produção nacional é capaz de fornecer aproximadamente 170 litros de leite por habitante ao ano. Essa quantidade é inferior ao indicado pelos órgãos de saúde nacionais e internacionais que é de 210 litros ao ano por habitante. E com uma estimativa de crescimento da população até 2023 de 216 milhões de habitantes, o governo federal por meio do Plano Mais Pecuária lançado em 2014, pretende aumentar de forma sustentável a produtividade e a competitividade da pecuária bovina de leite e corte.

Sendo assim, o desenvolvimento de um software para a administração de propriedades rurais produtoras de leite, com ênfase na agricultura familiar, tem como intuito auxiliar o administrador com a coleta e organização das informações, gerando históricos de produtividade e lucratividade. Embora existam diversos softwares no mercado, para (DEPONTI 2014) há um descompasso entre os softwares e ferramentas de gestão existentes e a aderência por parte dos produtores, tendo em vista a desconexão com as necessidades e habilidades deste público.

2. Fundamentação Técnica

O Brasil ocupa um lugar de destaque no cenário mundial, sendo o quinto maior produtor de leite de vaca, atrás apenas da União Europeia, Estados Unidos, Índia e China, segundo dados do Departamento de Agricultura dos Estados Unidos (*United States Department of Agriculture* - USDA). No cenário nacional, a bovinocultura de leite é predominante nas regiões Sul e Sudeste. A região Sul é a maior produtora, responsável por 35,2% do montante produzido, que em 2015 foi de 35 bilhões de litros. É possível observar no Gráfico 1, o crescimento ao longo dos anos e como se distribui a produção nas regiões do país.

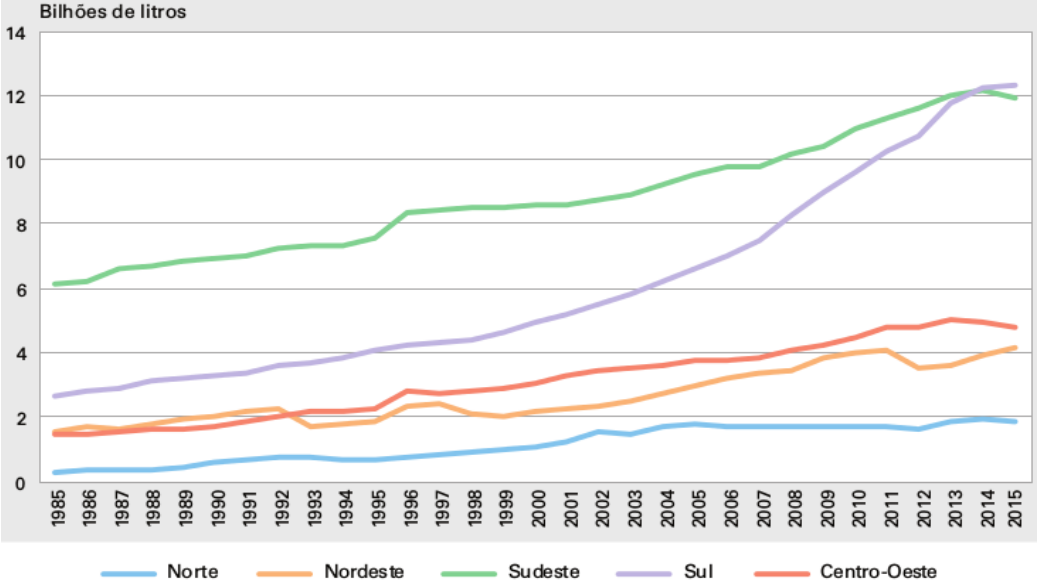


Gráfico 7: Evolução na produção de leite nas regiões dos pais, entre 1985-2015

De acordo com o Portal Brasil cerca de 70% dos alimentos consumidos no Brasil é proveniente dos agricultores familiares, mais especificamente no setor de laticínios é responsável por 58% da oferta do produto no mercado.

3. Estado da Arte

Foram realizadas pesquisas na web por *softwares* disponíveis no mercado com características semelhantes às apresentadas neste trabalho. Todos os softwares aqui descritos estão disponíveis para uso independente do custo para o usuário. Os seguintes *softwares* foram identificados: Gisleite, 4milk, Roda da Reprodução, SGFLeite.

O SGFLeite é um *software* desenvolvido para plataforma *Windows*, voltado para a gerência da produção de leite. Permite controle do rebanho, controle da produção, reprodução, além do controle sanitário, financeiro e zootécnico da propriedade (SGFLEITE, 2017).

O 4milk é um *software* desenvolvido pela empresa 4Hoofs Tecnologia da Informação LTDA, empresa sediada na cidade de Nova Lima no estado de Minas Gerais. Este software permite os manejos reprodutivos, zootécnicos e sanitários, além de gerir a produção das matrizes. O aplicativo também disponibiliza ferramentas para a importação de dados de outros sistemas de forma gratuita. Também oferece uma plataforma para comércio de animais entre usuários do sistema (4MILK, 2017).

A Roda da Reprodução é um aplicativo para o sistema Android desenvolvido pela Empresa Brasileira de Pesquisa Agropecuária (Embrapa) que permite gerenciar o manejo reprodutivo e produtivo do rebanho. O aplicativo é disponibilizado de forma gratuita pela empresa.

Gisleite é um sistema *web* desenvolvido em parceria entre Embrapa Gado de Leite e a empresa Gemini Sistemas com apoio da Fundação de Amparo à pesquisa de Minas Gerais (FAPEMIG), da Financiadora de Estudos e Projetos (FINEP) e da Prefeitura Municipal de Juiz de Fora (Guia de usuário Gisleite). O sistema é oferecido

de forma gratuita, embora o suporte e treinamento possua custo que ficam ao cargo da empresa Gemini Sistemas.

4. Oportunidade de Negócio

Segundo dados do Centro de Estudos sobre as Tecnologias da Informação e da Comunicação (Cetic). Em 2015, apenas 22% dos domicílios rurais tinham acesso à internet. Numero significativamente menor aos 56% encontrado na área urbana. Todavia, houve um crescimento muito significativo nos últimos anos, se comparado a 2008, quando o número de domicílios rurais com acesso a internet era de apenas 4% (CETIC, 2015).

Tendo em vista a popularização da internet no campo, aliado a importância econômica da pecuária de leite, principalmente para a agricultura familiar. Como descritos nos capítulos iniciais deste estudo. Configura-se um cenário propício para o desenvolvimento de um serviço web de gestão da produção de leite.

5. Projeto

O sistema se divide em quatro módulos, denominados rebanho, ordenha, reprodução e saúde animal. Para ter acesso ao sistema, o cliente precisa efetuar um cadastro informando alguns dados pessoais. Além disso, precisa criar uma senha de acesso, além de fornecer um e-mail válido que será o nome de usuário.

No módulo de rebanho, o usuário pode visualizar todos os animais cadastrados no sistema, com suas informações básicas. Ao selecionar um dos animais listados é possível ver todas as suas informações, além de permitir editar ou excluir o animal.

Na parte correspondente ao módulo ordenha. O usuário pode registrar a quantidade produzida, informar a quantidade vendida e o valor recebido por litro de leite comercializado. Isso permite ao sistema gerar históricos de produção, de receitas e de preço médio comercializado. Sendo representados por gráficos mensais.

No módulo de reprodução. É possível visualizar e registrar as inseminações, os partos realizados e as gestações abortadas. Também é possível visualizar históricos de nascimentos e inseminações através de gráficos mensais.

Na parte de saúde animal, encontra-se a o histórico de tratamentos provido aos animais. Além de permitir cadastrar um novo tratamento e o agendamento de tratamentos futuros.

6. Desenvolvimento

O Ordenha Digital foi desenvolvido como uma plataforma online, possibilitando aos produtores utilizarem o sistema via diversos aparelhos que não só um único smartphone, podendo estes acessar seus registros e dados de qualquer dispositivo conectado à rede. Desta maneira o usuário poderá utilizar seu smartphone para registrar um parto diretamente no local de tal e depois visualizar seus dados com mais conforto em seu desktop dentro de casa, por exemplo. Esta experiência só é possível devido à expansão do acesso à internet no meio rural, como citado anteriormente **neste artigo**. Também pensando no futuro a centralização dos dados, consequência de um sistema online, possibilita que o Ordenha Digital possa emitir relatórios gerais de produção de

leite de uma determinada região, ou de uma cooperativa, bem como disponibilizar um módulo de troca de informação entre produtores.

7. Conclusão

O desenvolvimento do Ordenha Digital mostrou-se bastante plausível, pois existe uma demanda de mercado muito grande a ser explorado. Isso é percebido nos softwares pesquisados, sendo todos eles relativamente novos no mercado. Muito em virtude da popularização da internet no campo, que apresentou um amplo crescimento nos últimos anos.

Além disso, as projeções de crescimento da pecuária de leite para os próximos anos são significativas. Sendo a busca pela melhora da produtividade que é muito baixa no Brasil, um dos fatores que podem contribuir para o uso de softwares ou outras ferramentas de gestão.

References

Comitê Gestor da Internet. TIC DOMICÍLIOS 2015: Pesquisa Sobre o uso Das Tecnologias de Informação e Comunicação nos Domicílios Brasileiros. São Paulo: no Brasil, 2016. Disponível em: <http://cetic.br/pesquisa/domicilios/publicacoes>. Acesso em 15 abril de 2017

Department of Agriculture. **Cows milk production and consumption: summary for selected countries**. In: ESTADOS UNIDOS. PSD online production, supply and distribution. Washington, DC: USDA, 2015. Disponível em: <http://apps.fas.usda.gov/psdonline/>. Acesso em: 01/01/2016.

Embrapa: Empresa brasileira de pesquisa e agropecuária. **Leite em números**. 2017. Disponível em: <http://www.cileite.com.br/content/leite-em-n%C3%BAmeros-produ%C3%A7%C3%A3o>. Disponível em Acesso em 02/10/2016

EMBRAPA. GisLeite - Guia do Usuário. 2016. Disponível em: <http://gisleite.cnpgl.embrapa.br/documentos/guia_usuario_gisleite.pdf>. Acesso em: 22 de maio de 2017.

EMBRAPA. GisLeite - Guia do Usuário. 2016. Disponível em: <http://gisleite.cnpgl.embrapa.br/documentos/guia_usuario_gisleite.pdf>. Acesso em: 22 de maio de 2017.

IBGE - Instituto Brasileiro de Geografia e Estatística. CENSO AGROPECUÁRIO 2006: Brasil, Grandes Regiões e Unidades da Federação. Rio de Janeiro: 2006. Disponível em: <http://biblioteca.ibge.gov.br/visualizacao/periodicos/51/agro_2006.pdf>. Acesso em: 02 set. 2016.

_____Instituto Brasileiro de Geografia e Estatística - Produção da Pecuária Municipal. Rio de Janeiro, v. 42, p.1-39, 2014

_____Instituto Brasileiro de Geografia e Estatística - Produção da Pecuária Municipal. Rio de Janeiro, v. 28, p.1-28, 2000.

MAPA - Ministério da Agricultura, Pecuária e Abastecimento. Plano Mais Pecuária. 2014. Disponível em: <http://www.agricultura.gov.br/arq_editor/file/Ministerio/Publicacao_v2.pdf> Acesso em 01/10/2016

Portal Brasil: Governo quer ampliar produção de leite em 40% em 10 anos. Disponível em: <http://www.brasil.gov.br/economia-e-emprego/2014/02/governo-quer-aumentar-producao-de-leite-em-40-em-10-anos> Acessado em 15/09/2016

Portal Gisleite. Disponível em: <http://gisleite.cnpgl.embrapa.br/lista.php?type=producao_leite> Acesso em 10 de novembro de 2017

Portal Embrapa. Disponível em: <https://www.embrapa.br/busca-de-produtos-processos-e-servicos/-/produto-servico/1430/gisleite---gestao-de-sistemas-de-producao-de-leite> Acesso em 20 de maio de 2017.

Portal 4milk. Disponível em: <http://www.4milk.com.br/Home> Acesso em 10 de março de 2017.

APENDICE II – CÓDIGO FONTE DO SISTEMA

./bowerrc

```
{
        "directory": "public/vendor"
}
```

.gitignore

```
node_modules/
public/vendor/
```

bower.json

```
{
 "name": "ordenha-digital",
 "description": "UFSC, TCC 2017.",
 "main": "server.js",
 "authors": [
   "Daniel Maragno & Jhon Faccin"
 ],
 "license": "ISC",
 "homepage": "",
 "ignore": [
   "**/.*",
   "node_modules",
   "bower_components",
   "test",
   "tests"
 ],
 "dependencies": {
   "angular": "1.3",
   "bootstrap": "^3.3.7",
   "angular-ui-router": "^0.4.2",
   "angular-ui-router-styles": "^1.1.0",
   "moment": "^2.17.1",
   "amcharts": "^3.15.2",
   "angular-locale-pt-br": "^1.3.15"
 },
 "resolutions": {
   "angular-ui-router": "^0.2.13",
   "angular": "1.3"
 }
}
```

package.json

```
{
 "name": "ordenha-digital",
 "version": "1.0.0",
 "description": "UFSC, TCC 2017.",
 "main": "index.js",
 "scripts": {
   "start": "node server.js",
   "postinstall": "./node_modules/bower/bin/bower install"
 },
 "author": "Daniel Maragno & Jhon Faccin",
 "license": "ISC",
 "dependencies": {
   "bcryptjs": "^2.4.3",
   "body-parser": "^1.16.1",
   "bower": "^1.8.0",
   "consign": "^0.1.2",
   "express": "^4.14.1",
   "jsonwebtoken": "^7.3.0",
   "mongoose": "^4.8.2"
 }
}
```

server.js

```
"use strict";

var http = require('http');
var app = require('./config/express')();

// Mongoose (MongoDB) connect
var uri = process.env.MONGODB_URI;
if(!uri)
        uri = 'mongodb://localhost:27015/ordenha-digital';

require('./config/database.js')(uri);

http.createServer(app).listen(app.get('port'), function() {
        console.log("Orgenha Digital operando na porta " + app.get('port'));
});
```

utils/app/user.js

```
"use strict";


// Load Modules
var bcrypt = require('bcryptjs');
var jwt = require('jsonwebtoken');
```

```
var fs = require('fs');

//
//        Suport Functions
//

// Read secret key for generate token
var readTokenSecret = function() {
   return String(fs.readFileSync(__dirname + "/token-secret"));
};


//
//        Password Operations
//

var geraSalt = function() {
        var saltRounds      = 14;
        var salt            = bcrypt.genSaltSync(saltRounds);
        return salt;
};

exports.generatePasswordHash = function(password) {
        var salt            = geraSalt();
        var bcryptHash      =  bcrypt.hashSync(password, salt);
        return bcryptHash;
};

exports.comparePasswordHash = function(plainPw, hashPw){
        var compare = bcrypt.compareSync(plainPw, hashPw);
        return compare;
};

// Token
exports.generateToken = function(user){
        // Read secret
        var secret     = readTokenSecret();
        // Generate token by User with email
        var token      = jwt.sign({ "_id": user._id, "email": user.email, "loginTime": new
Date()}, secret);

        return token;
}
```

app/controllers/models

animal.js

```
"use strict";
```

```
var mongoose = require('mongoose');

module.exports = function(){

	var schema = mongoose.Schema({

		numero: {
			type: String,
			required: true,
			index: {
				unique: false
			}
		},

		nome: {
			type: String
		},

		sexo: {
			type: String,
			enum: ["M","F"]
		},


		dataNasc: {
			type: Date
		},

		tipoBovino : {
			type: String
		},

		imagem: {
			type: String
		},

		// Observação qualquer (como Raça, Tipo de Pelo, etc...)
		observacao: {
			type: String
		},

		// Parents Info

		// Reprodutor do Rebanho
		parentNumero: {
			type: String
		},

		// Doador do semem (animal de fora do rebanho)
```

```
                parentReprodutorExterno: {
                        type: String
                },

                // Owner of animal
                email: {
                        type: String,
                        required: true,
                        index: {
                                unique: false
                        }
                },

                timeStamp: {
                        type: Date,
                        required: true
                },

                // Check if animal is alive or in the farm
                active: {
                        type: Boolean,
                        default: true
                }

        });

        // Return schema as User model
        return mongoose.model('Animal', schema);
}
```

cruza.js

```
"use strict";

var mongoose = require('mongoose');

module.exports = function(){

        var schema = mongoose.Schema({

                // Infos about Reprodutor cow from Rebanho

                numero: {
                        type: String,
                        required: true
                },

                // Infos abaouts Reprodutor
                //        - Reprodutor is, usually, an outsider from Rebanho.
```

```
        codReprodutor: {
                type: String
        },

        nomeReprodutor: {
                type: String
        },

        observacao: {
                type: String
        },

        // Infos about cruza

        dataCruza: {
                type: Date,
                required: true
        },

        // If status == 'open' then Provável data do Parto
        // If status == 'parto' then Data real do Parto
        dataParto: {
                type: Date
        },

        status: {
                type: String,
                enum: [ "open", "abort", "parto"]
        },

        // Owner of cruza
        email: {
                type: String,
                required: true,
                index: {
                        unique: false
                }
        },

        timeStamp: {
                type: Date,
                required: true
        },


});

// Return schema as User model
return mongoose.model('Cruza', schema);
```

```
}
```

ordenha.js

```
"use strict";

var mongoose = require('mongoose');

module.exports = function(){

	var schema = mongoose.Schema({

		// Owner of ordenha
		email: {
			type: String,
			required: true,
			index: {
				unique: false
			}
		},

		timeStamp: {
			type: Date,
			required: true
		},

		// Quantidade em litros
		qtd: {
			type: Number,
			required: true
		},

		// Quantidade em litro consumida pelos bezerros
		qtd_bezerro: {
			type: Number,
			required: true
		},

		// Quantidade em litro para consumo próprio
		qtd_proprio: {
			type: Number,
			required: true
		},

		// Preço total
		preco_total: {
			type: Number,
			required: true
		},
```

```
                // Preço por litro
                preco_litro: {
                        type: Number,
                        required: true
                }

        });

        // Return schema as User model
        return mongoose.model('Ordenha', schema);
}
```

saudeAnimal.js

```
"use strict";

var mongoose = require('mongoose');

module.exports = function(){

        var schema = mongoose.Schema({

                // Identificador do Animal
                email: {
                        type: String,
                        required: true,
                        index: {
                                unique: false
                        }
                },

                numero: {
                        type: String,
                        required: true,
                        index: {
                                unique: false
                        }
                },


                // Info da SaudeAnimal
                nomeTratamento: {
                        type: String,
                        required: true,
                        index: {
                                unique: true
                        }
```

```
                },

                tipoTratamento: {
                        type: String,
                        enum: ["pontual", "continuo"]
                        // Pontual são tratamentos com inicio e fim, Continuo são os que
não acabam
                },

                timeStampInicio: {
                        type: Date
                },

                timeStampFim: {
                        type: Date
                },

                // Frequencia do tratamento em dias
                frequencia: {
                        type: Number
                },

                eventos: [{
                        timeStamp:   { type: Date },
                        status:      { type: String, enum: ["ok", "open", "abort"] },
                        observacao: { type: String }
                }],

                observacao: {
                        type: String
                }




        });

        // Return schema as User model
        return mongoose.model('SaudeAnimal', schema);
}
```

user.js

```
"use strict";

var mongoose = require('mongoose');

module.exports = function(){
```

```
	var schema = mongoose.Schema({

		timeStamp: {
			type: Date,
			required: true
		},

		email: {
			type: String,
			required: true,
			index: {
				unique: true
			}
		},

		passwd: {
			type: String,
			required: true
		},

		name: {
			type: String,
			required: true
		},

		cpf: {
			type: String,
			match: /^[0-9]{11}$/,
		},

		// Control keys

		token: {
			type: String
		},

		isActive: {
			type: Boolean,
			required: true
		},

	});

	// Return schema as User model
	return mongoose.model('User', schema);
}
```

app/controllers

animal.js

```
"use strict";

module.exports = function(app){

	// Model instance (MongoDB collection access)
	var Animal = app.models.animal;


	var controller = {};

	controller.CreateAnimal = function(req, res){

		// Animal owner
		var email = req.body.user.email;

		var animal = req.body.animal;

		animal["email"] = email;


		storeAnimal(animal, function(err){
			if(err) res.status(400).send({"message": err});
			else    res.status(200).send({"message": "Animal cadastrado com
sucesso!"});
		});


	};


	controller.ListAnimals = function(req, res){

		// Animals owner
		var email = req.body.user.email;

		Animal
			.find({"email": email, "active": true}, {'__v':0, '_id':0, 'active':0,
'email':0})
			.sort({"numero": 1})
			.exec(function(err, animals){
				if(err) res.status(500).send({"message": "DB error
internal"});
				else    res.status(200).send(animals);
			});

	};

	controller.ListSpecificAnimal = function(req, res){

		// Animals owner
```

```
		var email = req.body.user.email;
		// Numero from URL
		var numero   = req.params.numero;

		Animal
			.find({'numero': numero, 'email': email, 'active': true}, {'__v':0,
'_id':0, 'active':0, 'email':0})
			.exec(function(err, animal){
				if(err)
					res.status(500).send({"message": "DB error
internal"});
				else if(animal.length < 1)
					res.status(404).send({"message": "Animal not
found!"});
				else
					res.status(200).send(animal[0]);
			});

	};

	controller.EditAnimal = function(req, res){

		// Animal owner
		var email      = req.body.user.email;
		// Numero from URL
		var numero   = req.params.numero;

		// Config what is updated
		var animal    = req.body.animal;
		animal['email']  = email;
		animal['numero'] = numero;

		flagActiveToFalse(email, numero, function(err){
			if(err)
				res.status(500).send({"message": "Flag to false error"})
			else{
				storeAnimal(animal, function(err){
					if(err)
						res.status(500).send({"message": "Store new
animal error"});
					else
						res.status(200).send({"message": "Animal
editado com sucesso!"});
				});
			}
		});

	};

	controller.DeleteAnimal = function(req, res){
```

```
		// Animal owner
		var email	= req.body.user.email;
		// Numero from URL
		var numero	= req.params.numero;

		flagActiveToFalse(email, numero, function(err){
			if(err) res.status(500).send({"message": "DB error internal"});
			else	res.status(200).send({"message": "Animal removido com
sucesso!"})
		});

	};

	//
	// Utils functions
	//

	function storeAnimal(animal, callback){

		// Set timeStamp
		animal["timeStamp"] = new Date();

		var animal_coll = new Animal(animal);

		animal_coll.save(function(err){
			callback(err);
		});
	};

	function flagActiveToFalse(email, numero, callback){
		Animal.update(
			{"email": email, "numero": numero, "active": true},
			{"$set": {"active": false}},
			function(err){
				callback(err);
			}
		);
	};

	controller.callVacaDeLeite = function(req, res){
		// Animal owner
		var email	= req.body.user.email;

		Animal
			.find({'email': email, 'tipoBovino': 'Vaca de Leite'}, {'__v':0, '_id':0,
'active':0, 'email':0})
			.sort({'numero': 1})
			.exec(function(err, animals){
				if(err) res.status(500).send({"message": "DB error
internal"});
				else	res.status(200).send(animals);
```

```
                });
        };


        return controller;

}

cruza.js

"use strict";

module.exports = function(app){

        // Model instance (MongoDB collection access)
        var Cruza = app.models.cruza;


        var controller = {};


        controller.listCruzas = function(req, res){

                // Owner
                var email = req.body.user.email;

                Cruza
                        .find({'email': email})
                        .sort({'dataCruza': -1})
                        .exec(function(err, cruzas){
                                if(err)  res.status(500).send({'message': 'Erro ao listar
Cruzas'});
                                else    res.status(200).send(cruzas);
                        });
        };

        controller.listCruzasForSpecificAnimal = function(req, res){
                // Owner
                var email  = req.body.user.email;
                var numero = req.params.numero;

                Cruza
                        .find({'email': email, 'numero': numero})
                        .sort({'timeStamp': -1})
                        .exec(function(err, cruzas){
                                if(err)  res.status(500).send({'message': 'Erro ao listar
Cruzas'});
                                else    res.status(200).send(cruzas);
                        });

        };
```

```
	controller.createCruza = function(req, res){

		// Owner
		var email = req.body.user.email;

		req.body.cruza.email = email;

		var cruza = new Cruza(req.body.cruza);

		cruza.timeStamp = new Date();
		cruza.status  = "open";

		// Calc Parto date
		cruza.dataParto = new Date(cruza.dataCruza);
		cruza.dataParto.setMonth(cruza.dataParto.getMonth()+9);

		cruza.save(function(err){
			if(err) res.status(400).send({'message': 'Erro ao cadastrar
Cruza!'});
			else    res.status(200).send({'message': 'Cruza cadastrada com
sucesso!'});
		})
	};

	controller.editCruza = function(req, res){

		// Owner
		var email = req.body.user.email;
		var id = req.param('id');

		var cruza = req.body.cruza;

		Cruza.update(
			{'email': email, '_id': id},
			{'$set': cruza},
			function(err){
				if(err)  res.status(400).send({'message': 'Erro ao editar
Cruza'});
				else    res.status(200).send({'message': 'Cruza editada
com sucesso!'})
			}
		);
	};


	controller.deleteCruza = function(req, res){

		// Owner
		var email = req.body.user.email;
		var id = req.param('id');
```

```
		Cruza.remove({'email': email, '_id': id}, function(err){
			if(err) res.status(500).send({'message': 'Erro ao remover
Cruza!'});
			else	res.status(200).send({'message': 'Cruza removida com
sucesso!'});
		});

	};

	controller.cruzaByMonth = function(req, res){
		// Owner
		var email  = req.body.user.email;

		Cruza.aggregate(
			{
				"$match": {'email': email},
			},
			{
				"$group": {
					"_id": {"year": {"$year": "$dataCruza"}, "month":
{"$month": "$dataCruza"} },
					"qtd": {"$sum": 1}
				}
			},
			{
				"$sort": {"_id.year": 1, "_id.month": 1}
			},
			function(err, cruzas){
				if(err){
					console.log(err);
					res.status(500).end();
				}
				else	res.status(200).send(cruzas);
			}
		)
	};

	controller.partoByMonth = function(req, res){
		// Owner
		var email  = req.body.user.email;

		Cruza.aggregate(
			{
				"$match": {'email': email, 'status': 'parto'},
			},
			{
				"$group": {
					"_id": {"year": {"$year": "$dataParto"}, "month":
{"$month": "$dataParto"} },
					"qtd": {"$sum": 1}
```

```
                        }
                    },
                    {
                        "$sort": {"_id.year": 1, "_id.month": 1}
                    },
                    function(err, partos){
                        if(err){
                            console.log(err);
                            res.status(500).end();
                        }
                        else    res.status(200).send(partos);
                    }
                )
        };

        return controller;

}
```

ordenha.js

```
"use strict";

module.exports = function(app){

        // Model instance (MongoDB collection access)
        var Ordenha = app.models.ordenha;

        var controller = {};

        controller.createOrdenha = function(req, res){

                var email = req.body.user.email;
                var ordenha = req.body.ordenha;

                // Atualiza Ordenha
                if('_id' in ordenha){
                        Ordenha.update({'email': email, '_id': ordenha._id}, {"$set":
ordenha}, function(err){
                        if(err) res.status(500).send({'message': 'Erro ao cadastrar
ordenha! Por favor, tente novamente mais tarde.'});
                        else    res.status(200).send({'message': "Ordenha
atualizada com sucesso!"});
                        });
                }
                // Nova Ordenha
                else{
                        ordenha.email = email;

                        ordenha = new Ordenha(ordenha);
                        ordenha.save(function(err){
```

```
                        if(err) res.status(500).send({'message': 'Erro ao
cadastrar ordenha! Por favor, tente novamente mais tarde.'});
                        else    res.status(200).send({'message': "Nova Ordenha
cadastrada com sucesso!"});
                });
        }

};

controller.deleteOrdenha = function(req, res){

        var email = req.body.user.email;

        var id = req.param('id');

        Ordenha.remove({'_id': id, 'email': email}, function(err){
                if(err) res.status(500).send({'message': 'Falha ao remover
Ordenha! Por favor, tente novamente mais tarde.'});
                else    res.status(200).send({'message': 'Ordenha removida com
sucesso!'});
        });

};

controller.listOrdenhasByUser = function(req, res){

        var email = req.body.user.email;

        Ordenha.find({'email': email}, {}, {'sort': {'timeStamp': -1}}, function(err,
ordenhas){
                if(err){
                        console.log(err);
                        res.status(500).send({'message': 'Erro no servidor! Por
favor, tente novamente mais tarde.'});
                }
                else
                        res.status(200).send(ordenhas);
        });

};

controller.ordenhasByMonth = function(req, res){

        var email = req.body.user.email;

        Ordenha.aggregate(
                {
                        "$match": {
                                "email": email
                        }
                },
```

```
                {
                        "$group": {
                                "_id": { "year": {"$year": "$timeStamp"}, "month":
{"$month": "$timeStamp"} },
                                "qtd": {"$sum": "$qtd"},
                                "preco_total": {"$sum": "$preco_total"},
                                "avg_preco_litro": {"$avg": "$preco_litro"}
                        }
                },
                {
                        "$sort": {"_id.year": 1, "_id.month": 1}
                },
                function(err, ordenhas){
                        if(err){
                                console.log(err);
                                res.status(500).end();
                        }
                        else res.status(200).send(ordenhas);
                }
        )

};

controller.ordenhaByTimeStamp = function(req, res){

        var email      = req.body.user.email;
        var timeStamp = new Date(req.param("timeStamp"));

        Ordenha.find({"email": email, "timeStamp":
timeStamp}).exec(function(err, ordenhas){
                if(err){
                        console.log(err);
                        res.status(400).send({"message": "Request Inválido!"})
                }
                else
                        res.status(200).send(ordenhas);
        });
};

controller.findOrdenhasWeek = function(req, res){
        var email      = req.body.user.email;

        // Calc Date
        var week6 = new Date();
        week6.setHours(0,0,0,0);
        var week0 = new Date( week6.setDate(week6.getDate() -
week6.getDay()) );
        week6.setDate(week6.getDate() + 7);

        // Find Ordenhas
        Ordenha
```

```
                .find({"email": email, 'timeStamp': {"$gte": week0, "$lt":
week6}})
                .sort({"numero": 1, "timeStamp": 1})
                .exec(
                        function(err, ordenhas){
                                if(err){
                                        console.log(err);
                                        res.status(500).send({"message": "Algo deu
errado!"});
                                }
                                else
                                        res.status(200).send(ordenhas);
                        }
                );

        };

        controller.upsertMultiOrdenhas = function(req, res){
                var email      = req.body.user.email;
                var ordenhas = req.body.ordenhas;

                // create bulk obj
                var bulk = Ordenha.collection.initializeOrderedBulkOp();

                // Prepare ordenhas att
                for(var i=0, ordenha, id; ordenha=ordenhas[i++]; ){
                        ordenha.timeStamp = new Date(ordenha.timeStamp);
                        ordenha.timeStamp.setHours(0,0,0,0);
                        // ordenha.timeStamp.setHours(ordenha.timeStamp.getHours()-
3);

                        bulk
                                .find({'email':email, 'numero': ordenha.numero,
'timeStamp': ordenha.timeStamp})
                                .upsert()
                                .updateOne(ordenha)
                }

                // Execute bulk upserts
                bulk.execute(function(err, result){
                        if(err){
                                console.log(err);
                                res.status(500).send({"message": "Algo deu errado!"});
                        }
                        else{
                                console.log(result);
                                res.status(200).send({"message": "Ordenhas cadastradas
com sucesso!"});
                        }
                });
```

```
	};

	return controller;

}
```

saudeAnimal.js

```
"use strict";

module.exports = function(app){

	// Model instance (MongoDB collection access)
	var SaudeAnimal = app.models.saudeAnimal;


	var controller = {};

	controller.createTratamento = function(req, res){

		// Owner
		var email = req.body.user.email;

		var saudeAnimal = new SaudeAnimal(req.body.tratamento);
		saudeAnimal.email = email;

		saudeAnimal.save(function(err){
			if(err){
				console.log(err);
				res.status(400).send({"message": "Algo deu errado."});
			}
			else
				res.status(200).send({"message": "Tratamento cadastrado
com sucesso!"})
		});
	};

	controller.listTratamentoBasicInfo = function(req, res){

		// Owner
		var email = req.body.user.email;

		SaudeAnimal.find(
			{"email": email},
			{
				"_id": 1,
				"numero": 1,
				"nomeTratamento": 1,
				"tipoTratamento": 1,
				"timeStampInicio": 1,
				"timeStampFim": 1,
```

```
                        "frequencia": 1
                },
                function(err, tratamentos){
                        if(err){
                                console.log(err);
                                res.status(500).end();
                        }
                        else
                                res.status(200).send(tratamentos);
                }
        );
};

controller.getTratamentoSpecific = function(req, res){

        // Owner
        var email       = req.body.user.email;
        var id          = req.params.id;

        SaudeAnimal.find(
                {"email": email, "_id": id},
                function(err, tratamento){
                        if(err){
                                console.log(err);
                                res.status(500).end();
                        }
                        else
                                res.status(200).send(tratamento);
                }
        );
};

controller.updateEvent = function(req, res){

        // Owner
        var email               = req.body.user.email;
        var tratamento          = req.body.tratamento;

        // Check contínuo
        if(tratamento.tipoTratamento == "continuo"){
                var hoje = new Date();
                hoje.setHours(0,0,0,0);
                var dataUltimoEvento = new
Date(tratamento.eventos[tratamento.eventos.length-1].timeStamp);
                dataUltimoEvento.setHours(0,0,0,0);

                if(dataUltimoEvento<=hoje){

        dataUltimoEvento.setDate(dataUltimoEvento.getDate()+parseInt(tratamento.fr
equencia));
                        tratamento.eventos.push({
```

```
                                "timeStamp": dataUltimoEvento,
                                "status": "open",
                                "observacao": ""
                        });
                }
        }

        SaudeAnimal.update(
                {"email": email, "_id": tratamento._id},
                {"$set": {"eventos": tratamento.eventos}},
                function(err){
                        if(err){
                                console.log(err);
                                res.status(400).send({"message": "!"});
                        }
                        else
                                res.status(200).send({"message": "Evento
atualizado com sucesso!"});
                }
        );
};

controller.deleteTratamentoSpecific = function(req, res){
        // Owner
        var email       = req.body.user.email;
        var id          = req.params.id;

        SaudeAnimal.remove({"email": email, "_id": id}, function(err){
                if(err){
                        console.log(err);
                        res.status(400).send({"message": "!"});
                }
                else
                        res.status(200).send({"message": "Tratamento removido
com sucesso!"});
        });
};

controller.getTratamentoBySpecifcAnimal = function(req, res){
        // Owner
        var email       = req.body.user.email;
        var numero      = req.params.numero;

        SaudeAnimal.find(
                {'email': email, 'numero': numero},
                {
                        "_id": 1,
                        "numero": 1,
                        "nomeTratamento": 1,
                        "tipoTratamento": 1,
                        "timeStampInicio": 1,
```

```
                        "timeStampFim": 1,
                        "frequencia": 1
                },
                {'sort': {'tipoTratamento': 1, 'timeStampInicio': -1}},
                function(err, tratamentos){
                        if(err){
                                console.log(err);
                                res.status(500).send({"message": "!"});
                        }
                        else
                                res.status(200).send(tratamentos);
                }
        );

};

controller.proximosTratamentosAbertos = function(req, res){

        // Email
        var email       = req.body.user.email;
        var days        = parseInt(req.params.days);

        var limitDate = new Date();
        limitDate.setHours(0,0,0,0);
        limitDate.setDate(limitDate.getDate() + days);

        SaudeAnimal.aggregate(
                {"$unwind": "$eventos"},
                {"$match": {"email": email, "eventos.status": "open",
"eventos.timeStamp": {"$lt": limitDate }} },
                {"$sort": {"eventos.timeStamp": 1}},
                function(err, eventos){
                        if(err){
                                console.log(err);
                                res.status(500).end();
                        }
                        else res.status(200).send(eventos);
                }
        );
}

return controller;

}
```

session.js

```
"use strict";

var utils = require('../../utils/app/user.js');

module.exports = function(app){
```

```
	// Model instance (MongoDB collection access)
	var User = app.models.user;

	var controller = {};

	controller.login = function(req, res){

		var data = req.body.data;

		User.findOne({"email": data.email, "isActive": true}, function(err, user){

			if(err) res.status(500).send({"message": "DB error internal"});

			else if(!user) res.status(400).send({"message": "Email ou senha
inválidos!"});

			else if(utils.comparePasswordHash(data.passwd, user.passwd)){

				var token = utils.generateToken(user);

				User.update({"email": data.email, "isActive": true}, {"$set":
{"token": token}}, function(err){
					if(err) res.status(500).send({"message": "DB error
internal"});
					else    res.status(200).send({"token": token});
				});

			}

			else res.status(400).send({"message": "Email ou senha
inválidos!"});

		});

	};


	controller.logout = function(req, res){
		// Ask for a x-access-token
		var token = req.headers['x-access-token'];

		User.update({"token": token, "isActive": true}, {"$unset": {"token": ""}},
function(err){
			if(err) res.status(500).send({"message": "DB error internal"});
			else    res.status(200).send({"message": "Logout realizado com
sucesso!"})
		});

	};
```

```
	controller.checkLogin = function(req, res, next){
		// Ask for a x-access-token
		var token = req.headers['x-access-token'];

		User.findOne({"token": token, "isActive": true}, function(err, user){
			if(err)  res.status(500).send({"message": "DB error internal"});
			else if(!user) res.status(401).send({"message": "Faça login antes
de utilizar a API"});
			else {
				req.body.user = user;
				next();
			}
		});

	};


	return controller;

}

user.js

"use strict";

var utils = require('../../utils/app/user.js');

module.exports = function(app){

	// Model instance (MongoDB collection access)
	var User = app.models.user;


	var controller = {};

	controller.createUser = function(req, res) {
		var data = req.body.data;

		if ('passwd' in data){
			data.passwd	=
utils.generatePasswordHash(data.passwd);
			data.timeStamp	= new Date();
			data.isActive = true;
			console.log(data);

			var user = new User(data);
			user.save(function(err, userRes){
				if(!err){
					res.status(200).send({"message": "Usuário
registrado com sucesso!"});
					console.log(userRes);
```

```
                        }
                        else{
                                res.status(400).send({"message": "Este email já
existe em nossa base!"});
                                console.log(err)
                        }

                });

        }
        else
                res.status(400).end();
};

controller.getUser = function(req, res){
        var email = req.body.user.email;

        User.findOne({'email': email},{'__v':0, '_id':0, 'passwd':0, 'isActive':0,
'token': 0}, function(err, user){
                if(err)
                        res.status(400).send({"message": "Bad Request"});
                else if(!user)
                        res.status(404).send({"message": "User Not Found"})
                else
                        res.status(200).send({"user": user});

        });

};

return controller;

}
```

app/routes

animal.js

```
"use strict";

module.exports = function(app){

        var controller= app.controllers.animal;
        var session   = app.controllers.session;


        app.route('/animal')

                // Create new user
                .post(session.checkLogin, controller.CreateAnimal)
```

```
            // List Animals
            .get(session.checkLogin, controller.ListAnimals);


        app.route('/animal/:numero')

            .get(session.checkLogin, controller.ListSpecificAnimal)

            // Edit animal
            .put(session.checkLogin, controller.EditAnimal)

            // Delete animal
            .delete(session.checkLogin, controller.DeleteAnimal)

        app.route('/animal-de-leite')

            // Get All Vacas De Leite
            .get(session.checkLogin, controller.callVacaDeLeite)

};
```

cruza.js

```
"use strict";

module.exports = function(app){

        var controller= app.controllers.cruza;
        var session           = app.controllers.session;

        app.route('/cruza')

            // List Cruzas from an  user
            .get(session.checkLogin, controller.listCruzas)

            // Create new Cruza
            .post(session.checkLogin, controller.createCruza)

            // Edit cruza
            .put(session.checkLogin, controller.editCruza)

            // Delete Cruza
            .delete(session.checkLogin, controller.deleteCruza);

        app.route('/cruza/:numero')
            // List Cruzas from specific Animal
            .get(session.checkLogin, controller.listCruzasForSpecificAnimal);


        app.route('/cruza/cruza/bymonth')
```

```
                    // Number of cruzas by month
                    .get(session.checkLogin, controller.cruzaByMonth);


        app.route('/cruza/parto/bymonth')

                    // Number of cruzas by month
                    .get(session.checkLogin, controller.partoByMonth);

};
```

ordenha.js

```
"use strict";

module.exports = function(app){

        var controller= app.controllers.ordenha;
        var session   = app.controllers.session;


        app.route('/ordenha')

                // Create new ordenha
                .post(session.checkLogin, controller.createOrdenha)

                // Delete Ordenha
                .delete(session.checkLogin, controller.deleteOrdenha)

                // List Animals
                .get(session.checkLogin, controller.listOrdenhasByUser);


        app.route('/ordenhas-by-month')

                // List qtd of Ordenhas by month
                .get(session.checkLogin, controller.ordenhasByMonth);

        app.route('/ordenha-by-timestamp')

                // Get ordenha by timeStamp
                .get(session.checkLogin, controller.ordenhaByTimeStamp);

        app.route('/ordenha-multi')

                // Get ordenhas from the wekk
                .get(session.checkLogin, controller.findOrdenhasWeek)

                // Upsert multi Ordenha documents
                .post(session.checkLogin, controller.upsertMultiOrdenhas);
```

```
};
```

saudeAnimal.js

```
"use strict";

module.exports = function(app){

	var controller= app.controllers.saudeAnimal;
	var session   = app.controllers.session;


	app.route('/tratamento')

		// List Tratamentos
		.get(session.checkLogin, controller.listTratamentoBasicInfo)

		// Create Tratamento
		.post(session.checkLogin, controller.createTratamento);

	app.route('/tratamento/:id')

		// Get specific Tratamento
		.get(session.checkLogin, controller.getTratamentoSpecific)

		// Get specific Tratamento
		.delete(session.checkLogin, controller.deleteTratamentoSpecific);

	app.route('/tratamento-animal/:numero')

		// Get tratamentos from a Specific animal
		.get(session.checkLogin, controller.getTratamentoBySpecifcAnimal)

	app.route('/tratamento-eventos')

		// Update tratamento specific event
		.put(session.checkLogin, controller.updateEvent);

	app.route('/tratamento-eventos/:days')

		// Get Tratamentos with satus open $lt today + days
		.get(session.checkLogin, controller.proximosTratamentosAbertos)


};
```

session.js

```
"use strict";
```

```
module.exports = function(app){

	var controller= app.controllers.session;


	app.route('/login')

		// Login
		.post(controller.login);


	app.route('/logout')

		//Logout
		.get(controller.checkLogin, controller.logout);

};
```

user.js

```
"use strict";

module.exports = function(app){

	var controller= app.controllers.user;
	var session		= app.controllers.session;

	app.route('/user')

		// Get an existent user
		.get(session.checkLogin, controller.getUser)

		// Create new user
		.post(controller.createUser);

};
```

config/

database.js

```
"use strict";

var mongoose = require('mongoose');

module.exports = function(uri){

	mongoose.connect(uri);

	mongoose.connection.on('connected', function(){
```

```
		console.log("Mongoose! ON as " + uri);
	});

	mongoose.connection.on('disconnected', function(){
		console.log('Mongoose! DISCONNECTED');
	});

	mongoose.connection.on('error', function(error){
		console.log('Mongoose! Connection ERROR: \n' + error);
	});

	// Ensures that mongoose closes its connection when process is finished
	process.on('SIGINT', function(){
		mongoose.connection.close(function(){
			console.log('Mongoose! DESCONECTADO by the end of
application');
			process.exit(0);
		});
	});

}
```

express.js

```
"use strict";

var express   = require('express');
var consign   = require('consign');
var bodyParser      = require('body-parser');

module.exports = function(){

	var app = express();

	// Config PORT
	var port = process.env.PORT;
	if(!port)
		port = 3000;

	// Sets
	app.set('port', port);
	app.set('views', './app/views');

	// Middleware
	app.use(express.static('./public'));
	app.use(bodyParser.urlencoded({extended: true,limit: '5mb'}));
	app.use(bodyParser.json({limit: '5mb'}));

	// Loading MVC
	// consign({cwd: 'app'})
	consign({cwd: process.cwd()+"/app"})
```

```
            .include('models')
            .then('controllers')
            .then('routes')
            .into(app);


    return app;
}
```

public/app/animal

AnimalController.js

```
(function (){
    "use strict";

    function AnimalController($scope, $state, AuthService, RebanhoService){

        var tokenId = AuthService.getTokenId();

        initialize();
        function initialize(){
            if(!tokenId) $state.go('login');

            $scope.alertDanger = false;
            $scope.alertSuccess = false;

            $scope.numero = $state.params.numero;
            callAnimal($scope.numero);

            // Init image size
            $scope.inputImagem = {"result": "", "size": 0};

        };

        function callAnimal(numero){
            RebanhoService.callAnimal(numero)
                .then(
                    function(res){
                        var animal = res.data;
                        handleCRUD(animal);
                    },
                    function(res){
                        $state.go('rebanho');
                    }
                );
        };

        function handleCRUD(animal){
            console.log(animal);
            $scope.animal = animal;
```

```
                $scope.animal.dataNasc = new Date(animal.dataNasc);
            };


            $scope.deleteAnimal = function(){
                var animal = $scope.animal;

                if(confirm("Tem certeza que deseja deletar " + animal.nome + ",
Número " + animal.numero+"?")){
                    RebanhoService.deleteAnimal(animal.numero)
                        .then(
                            function(res){
                                alert(res.data.message);
                                $state.go('rebanho');
                            },
                            function(res){
                                $scope.alertDanger = true;
                                $scope.alertResMessage =
res.data.message;
                            }
                        )
                }
            };

        };

        angular.module('ordenhaDigital').controller('AnimalController',
AnimalController);

})();
```

animal.html

```
<div class="main-container">

        <div class="container container-title">
            <div class="row">
                <div class="col-sm-12">
                    <h1>Animal {{numero}}</h1>
                </div>
            </div>
        </div>

        <div class="container crud">
            <div class="row">
                <div class="col-md-5">
                    <img class="media-object" src="{{animal.imagem}}"
alt="No Image" height="340" width="100%">
                </div>

                <div class="col-md-6 col-md-offset-1">
```

```
<form name="formAnimal">

    <!-- Nome -->
    <div class="form-group">
        <div class="row">
            <div class="col-md-12">
                <label>Nome do Animal</label>
                <input class="form-control"
type="text" placeholder="Nome do Animal" ng-model="animal.nome" required>
            </div>
        </div>
    </div>

    <!-- Data Nasc -->
    <div class="form-group">
        <div class="row">
            <div class="col-md-12">
                <label>Data de
Nascimento</label>
                <input class="form-control"
type="date" ng-model="animal.dataNasc" required>
            </div>
        </div>
    </div>

    <!-- Tipo do Bovino -->
    <div class="form-group">
        <div class="row">
            <div class="col-md-12">
                <label>Tipo</label>
                <input class="form-control"
type="text" placeholder="Tipo" ng-model="animal.tipoBovino" required>
            </div>
        </div>
    </div>


    <!-- ALERTS -->
    <div class="row">
        <div class="col-md-12">
            <div class="alert alert-success" ng-
show="alertSuccess">
                <button type="button"
class="close" ng-click="alertSuccess=false"><span>&times;</span></button>

    <strong>{{alertResMessage}}</strong>
            </div>
        </div>
    </div>

    <div class="row">
```

```
                                        <div class="col-md-12">
                                                <div class="alert alert-danger" ng-
show="alertDanger">
                                                        <button type="button"
class="close" ng-click="alertDanger=false"><span&times;</span></button>
                                                        {{alertResMessage}}
                                                </div>
                                        </div>
                                </div>
                                <!-- ALERTS END -->

                                <!-- Submit Button -->
                                <div class="form-group">
                                        <div class="row">
                                                <div class="col-md-3">
                                                        <button class="btn btn-danger
form-control" ng-click="deleteAnimal()">Deletar</button>
                                                </div>
                                        </div>
                                </div>

                        </form>

                </div>
        </div>
</div>
```

home/HomeController.js

```
(function (){
        "use strict";

        function HomeController($scope, $state, AuthService, UserService,
SaudeAnimalService, RebanhoService){

                var tokenId = AuthService.getTokenId();

                initialize();
                function initialize(){
                        if(!tokenId) $state.go('login');

                        UserService.callUser().then(handleUser);

        SaudeAnimalService.proximosTratamentosAbertos(7).then(handleAtividades);
                        RebanhoService.callAnimals().then(handleAnimals);
                };

                function handleUser(res){
                        var user = res.data.user;
```

```
                $scope.user = user;
            };

            function handleAtividades(res){
                var eventos = res.data;
                $scope.eventos = eventos;
            };

            function handleAnimals(res){
                var animals = res.data;
                console.log(animals);
                $scope.animals = animals;
            }

        };

        angular.module('ordenhaDigital').controller('HomeController', HomeController);

})();
```

home/home.html

```
<div class="main-container">
        <div class="container container-title">
            <div class="row">
                <div class="col-sm-12">
                    <h1>{{user.name}}</h1>
                </div>
            </div>
        </div>


        <div class="container">

            <div class="row">

                <!-- User Area -->
                <div class="col-md-4 canvas">

                    <div class="form-group">
                        <div class="row">
                            <div class="col-md-12">
                                <label>Email</label>
                                <div class="form-
info">{{user.email}}</div>
                            </div>
                        </div>
                    </div>

                    <div class="form-group">
                        <div class="row">
```

```
                                        <div class="col-md-12">
                                                <label>Cadastro</label>
                                                <div class="form-
info">{{user.timeStamp | date: "dd MMMM y"}}</div>
                                        </div>
                                </div>
                        </div>

                        <div class="form-group">
                                <div class="row">
                                        <div class="col-md-12">
                                                <label>Rebanho Total</label>
                                                <div class="form-
info">{{animals.length}} animais</div>
                                        </div>
                                </div>
                        </div>

                </div>

                <!-- Eventos Area -->
                <div class="col-md-8 canvas">

                        <div class="row">
                                <div class="col-sm-12">
                                        <h3>Eventos Pendentes</h3>
                                </div>
                        </div>

                        <div class="row">
                                <div class="col-md-12 eventosArea">
                                        <table class="table" ng-
show="eventos.length > 0">
                                                <tr
                                                        ng-repeat="e in eventos"
                                                        ng-click="$state.go('saude-
animal-tratamento', {'id': e._id});"
                                                        style="cursor: pointer;">
                                                        <td>{{e.nomeTratamento}}</td>
                                                        <td
class="numero">{{e.numero}}</td>
                                                        <td>{{e.observacao}}</td>
                                                        <td>{{e.eventos.timeStamp |
date: "dd MMMM y"}}</td>
                                                </tr>
                                        </table>
                                        <div ng-show="eventos.length < 1">

                                        </div>
                                </div>
                        </div>
```

```
                </div>

            </div>

        </div>

</div>
```

login/LoginController.js

```
(function (){
    "use strict";

    function LoginController($scope, $state, AuthService){

        var tokenId = AuthService.getTokenId();

        initialize();
        function initialize(){
            if(tokenId) $state.go('home');
        };


        $scope.doLogin = function(form){

            AuthService.login(form)
                        .then(
                            function(res){
                                var token = res.data.token;
                                AuthService.setTokenId(token);
                                $state.go('home');
                            },
                            function(res){
                                $scope.alertResMessage =
res.data.message;
                                $scope.alertRes = true;
                            }
                        );

        }

        $scope.registrar = function(){
            $state.go('register');
        }

    };

    angular.module('ordenhaDigital').controller('LoginController', LoginController);
```

```
})();
```

login/login.html

```
<div class="container">
	<div class="col-md-6 col-md-offset-3">
		<div class="panel">
			<div class="panel-heading">
				<h2>Bem Vindo ao Ordenha Digital</h2>
			</div>
			<div class="panel-body">
				<form name="formLogin">

					<div class="form-group">
						<div class="row">
							<div class="col-md-12">
								<input class="form-control"
type="email" name="email" placeholder="Email" ng-model="form.email" ng-
pattern="/^[^\s@]+@[^\s@]+\.[^\s@]{2,}$/" required>
							</div>
						</div>
					</div>

					<div class="form-group">
						<div class="row">
							<div class="col-md-12">
								<input class="form-control"
type="password" name="passwd" placeholder="Senha" ng-model="form.passwd"
required>
							</div>
						</div>
					</div>

					<div class="row">
						<div class="col-sm-12">
							<div class="alert alert-danger" ng-
show="alertRes">
								<button type="button"
class="close" ng-click="alertRes=false"><span>&times;</span></button>

	<strong>{{alertResMessage}}</strong>
							</div>
						</div>
					</div>

					<div class="form-group">
						<div class="row">
							<div class="col-sm-6">
								<button type="submit"
class="form-control btn btn-primary" ng-click="doLogin(form)">Login</button>
							</div>
```

```
                                        <div class="col-sm-6">
                                                <button class="form-control btn
btn-primary" ng-click="registrar()">Registrar-se</input>
                                        </div>
                                </div>
                        </div>

                        <div class="row namerights">
                                <div class="col-sm-12">
                                        <small>2017 - Ordenha
Digital</small>
                                </div>
                        </div>

                </form>
        </div>
    </div>
</div>
```

nav/NavController.js

```
(function (){
    "use strict";

    function NavController($scope, $state, $rootScope, AuthService){

        // Coloca botões
        $rootScope.$on('$stateChangeStart', function(){
            if(AuthService.getTokenId()){
                $scope.logado = true;
            }
            else{
                $scope.logado = false;
            }

        });


        $scope.doLogout = function(){
            AuthService.logout()
                        .then(
                            function(){
                                AuthService.setTokenId("");
                                $state.go('login');
                            },
                            function(){
                                AuthService.setTokenId("");
                                $state.go('login');
                            }
                        )
```

```
		}

		$scope.goTo = function(stateName){
			$state.go(stateName);
		}

	};

	angular.module('ordenhaDigital').controller('NavController', NavController);

})();
```

ordenha/OrdenhaCadastroController.js

```
(function (){
	"use strict";

	function OrdenhaCadastroController($scope, $state, AuthService,
OrdenhaService){

		var tokenId = AuthService.getTokenId();

		initialize();
		function initialize(){
			if(!tokenId) $state.go('login');

			$scope.alertSuccess = false;
			$scope.alertError = false;
			$scope.alertResMessage = "";


			var startDate = new Date();
			startDate.setHours(0,0,0,0);
			setForm(startDate);
		};

		function setForm(timeStamp){
			OrdenhaService.ordenhaByTimeStamp(timeStamp)
				.then(function(res){
					// If Ordenha exists
					if(res.data.length){
						$scope.form = res.data[0];
						$scope.form.timeStamp = new
Date($scope.form.timeStamp);
					}
					// Else Cadastra nova
					else{
						$scope.form = {
							"timeStamp": timeStamp,
							"qtd": 0.0,
							"qtd_bezerro": 0.0,
```

```
                                        "qtd_proprio": 0.0,
                                        "preco_litro": 0.0,
                                        "preco_total": 0.0
                                    };
                                }

                            });
                };

                $scope.attPrecoTotal = function(){
                        $scope.form.preco_total = $scope.form.preco_litro *
$scope.form.qtd;
                };

                $scope.handleTimeStampChange = function(){
                        setForm($scope.form.timeStamp);
                };

                $scope.salvaOrdenha = function(){
                        OrdenhaService.storeOrdenha($scope.form)
                                .then(
                                        // Status 200
                                        function(res){
                                                $scope.alertSuccess = true;
                                                $scope.alertResMessage =
res.data.message;
                                        },
                                        // Status 400
                                        function(res){
                                                $scope.alertError = true;
                                                $scope.alertResMessage =
res.data.message;
                                        }
                                )
                };

        };

        angular.module('ordenhaDigital').controller('OrdenhaCadastroController',
OrdenhaCadastroController);

})();
```

ordenha/OrdenhaController.js

```
(function (){
        "use strict";

        function OrdenhaController($scope, $state, AuthService, OrdenhaService){

                var tokenId = AuthService.getTokenId();
```

```
        initialize();
        function initialize(){
            if(!tokenId) $state.go('login');

            $scope.tab = 'qtd';

            OrdenhaService.callOrdenhas().then(formatOrdenhas);

    OrdenhaService.callOrdenhasByMonth().then(formatOrdenhasByMonth);
        };

        $scope.cadastraOrdenha = function(){
            $state.go('ordenha-cadastra');
        };

        function formatOrdenhas(res){
            var ordenhas = res.data;
            $scope.ordenhas = ordenhas;
        };

        function formatOrdenhasByMonth(res){
            var data = res.data;
            var dataProvider = generateDataProvider(data);
            $scope.dataProvider = dataProvider;
            console.log(dataProvider)

            chartQtd(dataProvider);
            chartPrecoTotal(dataProvider);
            chartAvgPrecoLitro(dataProvider);

        };

        function generateDataProvider(data){
            var dataProvider = [];

            if(data.length > 0){
                var firstNode = data[0]['_id']
                var inicialDate = new Date(firstNode.year+"-
"+firstNode.month+"-00:00:00");
                var finalDate = new Date()

                // Create auxDict
                var auxDict = {};
                while(inicialDate < finalDate){
                    var key =  inicialDate.getFullYear() + "-" + ("0" +
((inicialDate).getMonth() + 1)).slice(-2);
                    auxDict[key] = {
                        'timeStamp':
moment(inicialDate).format("MMM YYYY"),
                        'mes': new Date(inicialDate),
```

```
                                        'qtd': 0,
                                        'preco_total': 0,
                                        'avg_preco_litro': 0
                                    };

                                    inicialDate.setMonth(inicialDate.getMonth() + 1);
                                }

                                // put qtds into auxDict
                                for(var i=0, d; d=data[i++];){
                                    var key = d['_id'].year+"-"+("0" +
d['_id'].month).slice(-2);
                                    auxDict[key].qtd = d.qtd;
                                    auxDict[key].preco_total = d.preco_total;
                                    auxDict[key].avg_preco_litro = d.avg_preco_litro;
                                }

                                var keys = Object.keys(auxDict);
                                keys.sort()

                                for(var i=0,k; k = keys[i++];)
                                    dataProvider.push(auxDict[k]);
                            }

                            return dataProvider;
                        }

                        function chartQtd(dataProvider){

                            var chart = AmCharts.makeChart("chartByMonth",
                                {
                                    "type": "serial",
                                    "categoryField": "timeStamp",
                                    // "startDuration": 1,
                                    "theme": "dark",
                                    "categoryAxis": {
                                        "gridPosition": "start"
                                    },
                                    "chartCursor": {
                                    "fullWidth":true,
                                    "valueLineEabled":true,
                                    "valueLineBalloonEnabled":true,
                                    "valueLineAlpha":0.5,
                                    "cursorAlpha":0
                                  },
                                    "trendLines": [],
                                    "graphs": [
                                        {
                                            "balloonText": "[[qtd]] L",
                                            "bullet": "round",
                                            "id": "AmGraph-1",
```

```
                                "title": "Ordenha",
                                "valueField": "qtd"
                        }
                ],
                "guides": [],
                "valueAxes": [
                        {
                                "id": "ValueAxis-1",
                                "title": "Quantidade (L)"
                        }
                ],
                "allLabels": [],
                "balloon": {},
                "legend": {
                        "enabled": true,
                        "useGraphSettings": true
                },
                "titles": [
                        {
                                "id": "Title-1",
                                "size": 15,
                                "text": "Ordenha por Mês"
                        }
                ],
                "dataProvider": dataProvider,
        }
    );
}; // chartQtd()

function chartPrecoTotal(dataProvider){
    AmCharts.makeChart("chartPrecoTotal",
        {
            "type": "serial",
            "categoryField": "timeStamp",
            // "startDuration": 1,
            "theme": "dark",
            "categoryAxis": {
                "gridPosition": "start"
            },
            "chartCursor": {
              "fullWidth":true,
              "valueLineEabled":true,
              "valueLineBalloonEnabled":true,
              "valueLineAlpha":0.5,
              "cursorAlpha":0
            },
            "trendLines": [],
            "graphs": [
                {
                    // "balloonText": "R$ [[preco_total]]",
                    "bullet": "round",
```

```
                                        "id": "AmGraph-1",
                                        "title": "Ordenha",
                                        "valueField": "preco_total",
                                        "balloonFunction":
function(graphDataItem, graph){
                                                return "R$ "+
graphDataItem.values.value.toFixed(2);
                                        }
                                }
                        ],
                        "guides": [],
                        "valueAxes": [
                                {
                                        "id": "ValueAxis-1",
                                        "title": "Preço Total (R$)"
                                }
                        ],
                        "allLabels": [],
                        "balloon": {},
                        "legend": {
                                "enabled": true,
                                "useGraphSettings": true
                        },
                        "titles": [
                                {
                                        "id": "Title-1",
                                        "size": 15,
                                        "text": "Receita por Mês"
                                }
                        ],
                        "dataProvider": dataProvider
                }
        );
};

function chartAvgPrecoLitro(dataProvider){
        AmCharts.makeChart("chartAvgPrecoLitro",
                {
                        "type": "serial",
                        "categoryField": "timeStamp",
                        // "startDuration": 1,
                        "theme": "dark",
                        "categoryAxis": {
                                "gridPosition": "start"
                        },
                        "chartCursor": {
                    "fullWidth":true,
                    "valueLineEabled":true,
                    "valueLineBalloonEnabled":true,
                    "valueLineAlpha":0.5,
                    "cursorAlpha":0
```

```
                },
                "trendLines": [],
                "graphs": [
                        {
                                "bullet": "round",
                                "id": "AmGraph-1",
                                "title": "Ordenha",
                                "valueField": "avg_preco_litro",
                                "balloonFunction":
function(graphDataItem, graph){
                                        return "R$ "+
graphDataItem.values.value.toFixed(2);
                                }
                        }
                ],
                "guides": [],
                "valueAxes": [
                        {
                                "id": "ValueAxis-1",
                                "title": "Preço Médio Por Litro (R$)"
                        }
                ],
                "allLabels": [],
                "balloon": {},
                "legend": {
                        "enabled": true,
                        "useGraphSettings": true
                },
                "titles": [
                        {
                                "id": "Title-1",
                                "size": 15,
                                "text": "Preço Médio Mensal por Litro"
                        }
                ],
                "dataProvider": dataProvider
        }
    );
};

$scope.removeOrdenha = function(id, index){
    if(confirm("Tem certeza que deseja remove a ordenha
"+index+"?"))
        OrdenhaService.removeOrdenha(id)
            .then(
                //success
                function(res){
                    alert(res.data.message);
                    $state.reload();
                },
                //error
```

```
                                        function(res){
                                                alert(res.data.message);
                                        }
                                )
                };
        };

        angular.module('ordenhaDigital').controller('OrdenhaController',
OrdenhaController);

})();
```

ordenha/cadastro.html

```
<div class="main-container">

        <div class="container container-title">
                <div class="row">
                        <div class="col-sm-12">
                                <h1>Ordenhas</h1>
                        </div>
                </div>
        </div>

        <div class="container">
                <div class="row">
                        <div class="col-md-4 col-md-offset-4">

                                <form name="formOrdenha">

                                        <!-- TimeStamp -->
                                        <div class="form-group">
                                                <div class="row">
                                                        <div class="col-md-12">
                                                                <label>Data da
Ordenha</label>
                                                                <input class="form-control"
type="date" ng-model="form.timeStamp" ng-change="handleTimeStampChange()"
required>
                                                        </div>
                                                </div>
                                        </div>

                                        <!--  Qtd -->
                                        <div class="form-group">
                                                <div class="row">
                                                        <div class="col-md-12">
                                                                <label>Leite Entregue</label>
                                                                <div class="input-group">
                                                                        <input class="form-
control" type="number" ng-model="form.qtd" ng-change="attPrecoTotal()" required>
```

```
                                        <div class="input-group-addon">L</div>
                                    </div>
                                </div>
                            </div>

                            <!-- Qtd Bezerro -->
                            <div class="form-group">
                                <div class="row">
                                    <div class="col-md-12">
                                        <label>Consumo Bezerros</label>
                                        <div class="input-group">
                                            <input class="form-control" type="number" ng-model="form.qtd_bezerro" required>
                                            <div class="input-group-addon">L</div>
                                        </div>
                                    </div>
                                </div>
                            </div>

                            <!-- Qtd Próprio -->
                            <div class="form-group">
                                <div class="row">
                                    <div class="col-md-12">
                                        <label>Consumo Próprio</label>
                                        <div class="input-group">
                                            <input class="form-control" type="number" ng-model="form.qtd_proprio" required>
                                            <div class="input-group-addon">L</div>
                                        </div>
                                    </div>
                                </div>
                            </div>

                            <!-- Preço por Litro -->
                            <div class="form-group">
                                <div class="row">
                                    <div class="col-md-12">
                                        <label>Preço por Litro</label>
                                        <div class="input-group">
                                            <div class="input-group-addon">R$</div>
                                            <input
                                                class="form-control"
                                                type="number"
```

```
                                                    step="0.01"
                                                    ng-
model="form.preco_litro"
                                                    ng-
change="attPrecoTotal()"
                                                    required>
                                                </div>
                                            </div>
                                        </div>
                                    </div>

                                    <!--  Qtd -->
                                    <div class="form-group">
                                        <div class="row">
                                            <div class="col-md-12">
                                                <label>Preço Total</label>
                                                <div class="input-group">
                                                    <div class="input-group-
addon">R$</div>
                                                    <input class="form-
control" type="number" step="0.01" ng-model="form.preco_total" required readonly>
                                                </div>
                                            </div>
                                        </div>
                                    </div>

                                    <!-- ALERTS -->
                                    <div class="row">
                                        <div class="col-md-12">
                                            <div class="alert alert-success" ng-
show="alertSuccess">
                                                <button type="button"
class="close" ng-click="alertSuccess=false"><span>&times;</span></button>

        <strong>{{alertResMessage}}</strong>
                                            </div>
                                        </div>
                                    </div>

                                    <div class="row">
                                        <div class="col-md-12">
                                            <div class="alert alert-danger" ng-
show="alertError">
                                                <button type="button"
class="close" ng-click="alertError=false"><span>&times;</span></button>
                                                {{alertResMessage}}
                                            </div>
                                        </div>
                                    </div>
                                    <!-- ALERTS END -->
```

```
                                <!-- Submit Button -->
                                <div class="form-group">
                                        <div class="row">
                                                <div class="col-md-12">
                                                        <button class="btn btn-primary
form-control" ng-click="salvaOrdenha()">Salvar</button>
                                                </div>
                                        </div>
                                </div>

                        </form>

                </div>
        </div>
</div>



</div>
```

ordenha/ordenha.html

```
<div class="main-container">

        <div class="container container-title">
                <div class="row">
                        <div class="col-sm-12">
                                <h1>Ordenhas</h1>
                        </div>
                </div>
        </div>
        <div class="container">
                <div class="row">
                        <!-- Cadastrar Nova Ordenha -->
                        <div class="col-md-2 col-md-offset-10">
                                <button class="btn btn-primary col-md-12" title="Cadastrar
Ordenha" ng-click="cadastraOrdenha()">
                                        <span class="glyphicon glyphicon-plus"></span>
                                </button>
                        </div>
                </div>
        </div>

        <div class="onlyDesktop">
                <div class="container">
                        <div class="row">
                                <!-- Nav Pills -->
                                <div class="col-md-11">
                                        <ul class="nav nav-pills">
                                                <li ng-class="tab == 'qtd' ? 'active' : ""><a
data-toggle="tab" ng-click="tab = 'qtd'">Ordenhas por Mês</a></li>
```

```
                                        <li ng-class="tab == 'preco_total' ? 'active' : ""><a data-toggle="tab" ng-click="tab = 'preco_total'">Receita por Mês</a></li>
                                        <li ng-class="tab == 'avg_preco_litro' ? 'active' : ""><a data-toggle="tab" ng-click="tab = 'avg_preco_litro'">Preço Médio Mensal por Litro</a></li>
                                    </ul>
                                </div>
                            </div>
                        </div>

                        <div class="container" ng-show="tab == 'qtd'">
                            <div class="row">
                                <div class="col-md-12">
                                    <div id="chartByMonth" style="width: 100%; height: 400px; background-color: #282828;" ></div>
                                </div>
                            </div>
                        </div>

                        <div class="container" ng-show="tab == 'preco_total'">
                            <div class="row">
                                <div class="col-md-12">
                                    <div id="chartPrecoTotal" style="width: 100%; height: 400px; background-color: #282828;" ></div>
                                </div>
                            </div>
                        </div>

                        <div class="container" ng-show="tab == 'avg_preco_litro'">
                            <div class="row">
                                <div class="col-md-12">
                                    <div id="chartAvgPrecoLitro" style="width: 100%; height: 400px; background-color: #282828;" ></div>
                                </div>
                            </div>
                        </div>

                    </div>


                    <!-- Mobile table -->
                    <div class="onlyMobile">
                        <div class="container">
                            <div class="row">
                                <div class="col-md-12">

                                    <table class="table table-hover">
                                        <tr>
                                            <th>Mês</th>
                                            <th>Ordenha Total</th>
                                            <th>Preço Total</th>
```

```
                                        <th>Preço Médio por Litro</th>
                                    </tr>
                                    <tr ng-repeat="mes in dataProvider |
orderBy:'mes':true">
                                        <td class="date">{{mes.mes | date:
"MMMM y" }}</td>
                                        <td class="qtd">{{mes.qtd | number:2}}
L</td>
                                        <td class="total_preco">R$
{{mes.preco_total | number:2}}</td>
                                        <td class="avg_preco_litro">R$
{{mes.avg_preco_litro | number:2}}</td>
                                    </tr>
                                </table>

                            </div>
                        </div>
                    </div>
                </div>


</div>
```

rebanho/RebanhoCadastroController.js

```
(function (){
    "use strict";

    function RebanhoCadastroController($scope, $state, AuthService,
RebanhoService){

        var tokenId = AuthService.getTokenId();

        initialize();
        function initialize(){
            if(!tokenId) $state.go('login');

            $scope.alertDanger = false;
            $scope.alertSuccess = false;

            $scope.animalTypes = RebanhoService.animalTypes();
            RebanhoService.callAnimals().then(function(res){
$scope.animals = res.data; })

            // Init image size
            $scope.inputImagem = {"result": "", "size": 0};

            // Fill fields with Parents params
            $scope.form = $state.params;
```

```
		};

		$scope.cadastraAnimal = function(){
			if($scope.formAnimal.$valid){

				var animal = $scope.form;

				animal.imagem = $scope.imagem;

				RebanhoService.cadastraAnimal(animal)
					.then(
						function(res){
							$scope.alertSuccess = true;
							$scope.alertResMessage =
res.data.message;
						},
						function(res){
							$scope.alertDanger = true;
							$scope.alertResMessage =
res.data.message;
						}
					);

			}
		};

		$scope.$watch('inputImagem.result', function(){
			if($scope.inputImagem.size < 500000 &&
$scope.inputImagem.size > 0){
				$scope.imagem = $scope.inputImagem.result;
			}
			else if($scope.inputImagem.size >= 500000){
				$scope.alertDanger = true;
				$scope.alertResMessage = "Desculpe! A Imagem deve ter
tamanho máximo de 500KB.";
			}
		});

	};

	angular.module('ordenhaDigital').controller('RebanhoCadastroController',
RebanhoCadastroController);

})();
```

rebanho/RebanhoController.js

```
(function (){
	"use strict";

	function RebanhoController($scope, $state, AuthService, RebanhoService){
```

```
        var tokenId = AuthService.getTokenId();

        initialize();
        function initialize(){
            if(!tokenId) $state.go('login');

            $scope.tipoBovino = 'all';
            $scope.animalTypes = RebanhoService.animalTypes();
            $scope.animais = [];

            callAnimals();

        };

        function callAnimals(){
            RebanhoService.callAnimals()
                .then(function(res){
                    $scope.animais = res.data;

                    angular.forEach($scope.animais, function(animal){
                        animal.dataNasc  =
moment(animal.dataNasc).format("DD/MM/YYYY");
                        animal.timeStamp =
moment(animal.timeStamp).format("DD/MM/YYYY");
                    })

                });
        };

        $scope.cadastrarAnimal = function(){
            $state.go('rebanho-cadastro')
        };

        $scope.edit = function(numero){
            $state.go('rebanho-edit', {'numero': numero})
        };

        $scope.fichaAnimal = function(numero){
            $state.go('animal', {'numero': numero})
        };

    };

    angular.module('ordenhaDigital').controller('RebanhoController',
RebanhoController);

})();
```

rebanho/RebanhoEditController.js

```
(function (){
	"use strict";

	function RebanhoEditController($scope, $state, AuthService,
RebanhoService, ReproducaoService, SaudeAnimalService){

		var tokenId = AuthService.getTokenId();

		initialize();
		function initialize(){
			if(!tokenId) $state.go('login');

			$scope.tab = $state.params.tab;

			$scope.alertDanger = false;
			$scope.alertSuccess = false;

			$scope.numero = $state.params.numero;
			callAnimal($scope.numero);

			$scope.animalTypes = RebanhoService.animalTypes();

			// Init image size
			$scope.inputImagem = {"result": "", "size": 0};

			// Call Cruzas, and Tratamentos

	ReproducaoService.callCruzasForSpecificAnimal($state.params.numero).then
(handleCruzas);

	SaudeAnimalService.getTratamentosByAnimal($state.params.numero).then(h
andleTratamentos);

			$scope.statusLabel = {};
			var statusList = ReproducaoService.getStatus();
			for(var i=0; i<statusList.length; i++){
				$scope.statusLabel[statusList[i].status] =
statusList[i].statusName;
			}

		};

		function callAnimal(numero){
			RebanhoService.callAnimal(numero)
				.then(
					function(res){
						var animal = res.data;
						handleCRUD(animal);
					},
					function(res){
						$state.go('rebanho');
```

```
                        }
                    );
            };

            function handleCRUD(animal){

                $scope.form = animal;

                $scope.form.dataNasc = new Date(animal.dataNasc);
                $scope.inputImagem.result = animal.imagem;
            };

            $scope.$watch('inputImagem.result', function(){
                if($scope.inputImagem.size < 500000 &&
$scope.inputImagem.size > 0){
                    $scope.form.imagem = $scope.inputImagem.result;
                }
            });


            function handleCruzas(res){
                $scope.cruzaOpenTotal = 0;
                $scope.cruzaPartoTotal = 0;
                $scope.cruzaAbortTotal = 0;

                var i;
                for(i in res.data){
                    res.data[i].dataCruza = new Date(res.data[i].dataCruza);
                    res.data[i].dataParto = new Date(res.data[i].dataParto);

                    switch(res.data[i].status){
                        case "open": $scope.cruzaOpenTotal++; break;
                        case "parto": $scope.cruzaPartoTotal++; break;
                        case "abort": $scope.cruzaAbortTotal++; break;
                    }
                }

                $scope.cruzas = res.data;
                // console.log(res.data);
            };


            $scope.editAnimal = function(){
                var animal = $scope.form;


                RebanhoService.editAnimal(animal.numero, animal)
                    .then(
                        function(res){
                            $scope.alertSuccess = true;
```

```
                                        $scope.alertResMessage =
res.data.message;
                                },
                                function(res){
                                        $scope.alertDanger = true;
                                        $scope.alertResMessage =
res.data.message;
                                }
                        );
        };

        $scope.deleteAnimal = function(){
                var animal = $scope.form;

                if(confirm("Tem certeza que deseja deletar " + animal.nome + ",
Número " + animal.numero+"?")){
                        RebanhoService.deleteAnimal(animal.numero)
                                .then(
                                        function(res){
                                                alert(res.data.message);
                                                $state.go('rebanho');
                                        },
                                        function(res){
                                                $scope.alertDanger = true;
                                                $scope.alertResMessage =
res.data.message;
                                        }
                                )
                }
        };

        function handleTratamentos(res){
                var tratamentos = res.data;
                console.log(tratamentos);

                $scope.tratamentosContinuos = [];
                $scope.tratamentosPontuais = [];

                for(var i=0, tratamento; tratamento=tratamentos[i++]; ){
                        if(tratamento.tipoTratamento == "pontual")
                                $scope.tratamentosPontuais.push(tratamento)
                        else if(tratamento.tipoTratamento == "continuo")
                                $scope.tratamentosContinuos.push(tratamento);
                }
        };

};

angular.module('ordenhaDigital').controller('RebanhoEditController',
RebanhoEditController);
```

```
})();
```

rebanho/cadastro.html

```
<div class="main-container">

	<div class="container container-title">
		<div class="row">
			<div class="col-sm-12">
				<h1>Cadastrar Animal</h1>
			</div>
		</div>
	</div>

	<div class="container crud">
		<div class="row">
			<div class="col-md-5">
				<img class="media-object" src="{{imagem}}" alt=""
height="340" width="100%">
			</div>

			<div class="col-md-6 col-md-offset-1">
				<form name="formAnimal">

					<!-- Nome -->
					<div class="form-group">
						<div class="row">
							<div class="col-md-12">
								<label>Nome do Animal</label>
								<input class="form-control"
type="text" placeholder="Nome do Animal" ng-model="form.nome" required>
							</div>
						</div>
					</div>

					<!-- Número -->
					<div class="form-group">
						<div class="row">
							<div class="col-md-12">
								<label>Número</label>
								<input class="form-control"
type="text" placeholder="Número" ng-model="form.numero" required>
							</div>
						</div>
					</div>

					<!-- Tipo do Bovino -->
					<div class="form-group">
						<div class="row">
							<div class="col-md-12">
```

```
                                        <label>Sexo</label>
                                        <select class="form-control" ng-
model="form.sexo" required>
                                            <option
value="M">Macho</option>
                                            <option
value="F">Fêmea</option>
                                        </select>
                                    </div>
                                </div>
                            </div>


                            <!-- Data Nasc -->
                            <div class="form-group">
                                <div class="row">
                                    <div class="col-md-12">
                                        <label>Data de
Nascimento</label>
                                        <input class="form-control"
type="date" ng-model="form.dataNasc" required>
                                    </div>
                                </div>
                            </div>

                            <!-- Tipo do Bovino -->
                            <div class="form-group">
                                <div class="row">
                                    <div class="col-md-12">
                                        <label>Tipo</label>
                                        <select class="form-control" ng-
model="form.tipoBovino" required>
                                            <option ng-repeat="t in
animalTypes" value="{{t}}">{{t}}</option>
                                        </select>
                                    </div>
                                </div>
                            </div>

                            <!-- Imagem  NOT REQUIRED-->
                            <div class="form-group">
                                <div class="row">
                                    <div class="col-md-12">
                                        <label>Imagem</label>
                                        <input type="file"
accept="image/*" fileread="inputImagem">
                                    </div>
                                </div>
                            </div>

                            <!-- PARENTS info  NOT REQUIRED-->
```

```
                        <div class="well">
                            <div class="form-group">
                                <div class="row">
                                    <div class="col-md-12">
                                        <label>Número da
Reprodutora</label>
                                        <select class="form-
control" ng-model="form.parentNumero">
                                            <option value="">-
-- Sem Informação ---</option>
                                            <option ng-
repeat="a in animals" value="{{a.numero}}" ng-selected="form.parentNumero ==
a.numero">{{a.numero}}</option>
                                        </select>
                                    </div>
                                </div>
                            </div>

                            <div class="form-group">
                                <div class="row">
                                    <div class="col-md-12">
                                        <label>Código do
Reprodutor Externo</label>
                                        <input class="form-
control" type="text" placeholder="Reprodutor Externo" ng-
model="form.parentReprodutorExterno">
                                    </div>
                                </div>
                            </div>
                        </div>

                        <!-- Observação -->
                        <div class="form-group">
                            <div class="row">
                                <div class="col-md-12">
                                    <label>Observação</label>
                                    <textarea class="form-control"
ng-model="form.observacao"></textarea>
                                </div>
                            </div>
                        </div>

                        <!-- ALERTS -->
                        <div class="row">
                            <div class="col-md-12">
                                <div class="alert alert-success" ng-
show="alertSuccess">
                                    <button type="button"
class="close" ng-click="alertSuccess=false"><span>&times;</span></button>

        <strong>{{alertResMessage}}</strong>
```

```
                                        </div>
                                    </div>
                                </div>

                                <div class="row">
                                    <div class="col-md-12">
                                        <div class="alert alert-danger" ng-
show="alertDanger">
                                            <button type="button"
class="close" ng-click="alertDanger=false"><span>&times;</span></button>
                                            {{alertResMessage}}
                                        </div>
                                    </div>
                                </div>
                                <!-- ALERTS END -->

                                <!-- Submit Button -->
                                <div class="form-group">
                                    <div class="row">
                                        <div class="col-md-3">
                                            <button class="btn btn-primary
form-control" ng-click="cadastraAnimal()">Cadastrar</button>
                                        </div>
                                    </div>
                                </div>

                            </form>

                        </div>
                    </div>
                </div>

            </div>
```

rebanho/edit.html

```
<div class="main-container">

        <div class="container container-title">
            <div class="row">
                <div class="col-sm-12">
                    <h1>Animal {{numero}}</h1>
                </div>
            </div>
        </div>

        <div class="container tabs">
            <div class="row">
                <ul class="nav nav-pills">
                    <li class="col-md-2" ng-class="tab == 'animal' ? 'active' :
""><a data-toggle="tab" ng-click="tab = 'animal'">Animal</a></li>
```

```
                        <li class="col-md-2" ng-class="tab == 'cruzas' ? 'active' : 2
""><a data-toggle="tab" ng-click="tab = 'cruzas'">Cruzas</a></li>
                        <li class="col-md-2" ng-class="tab == 'tratamento' ?
'active' : ""><a data-toggle="tab" ng-click="tab = 'tratamento'">Tratamentos</a></li>
                    </ul>
                </div>

        </div>

        <div class="container crud main-info" ng-show="tab == 'animal'">
            <div class="row">
                <div class="col-md-5">
                    <img class="media-object" src="{{form.imagem}}" alt="No
Image" height="340" width="100%">
                </div>

                <div class="col-md-6 col-md-offset-1">
                    <form name="formAnimal">

                        <!-- Nome -->
                        <div class="form-group">
                            <div class="row">
                                <div class="col-md-12">
                                    <label>Nome do Animal</label>
                                    <input class="form-control"
type="text" placeholder="Nome do Animal" ng-model="form.nome" readonly>
                                </div>
                            </div>
                        </div>

                        <!-- Data Nasc -->
                        <div class="form-group">
                            <div class="row">
                                <div class="col-md-12">
                                    <label>Data de
Nascimento</label>
                                    <input class="form-control"
type="date" ng-model="form.dataNasc" readonly>
                                </div>
                            </div>
                        </div>

                        <!-- Sexo -->
                        <div class="form-group">
                            <div class="row">
                                <div class="col-md-12">
                                    <label>Sexo</label>
                                    <input class="form-control"
type="text" ng-model="form.sexo" readonly>
                                </div>
                            </div>
```

```
                    </div>

                    <!-- Tipo do Bovino -->
                    <div class="form-group">
                        <div class="row">
                            <div class="col-md-12">
                                <label>Tipo</label>
                                <select class="form-control" ng-
model="form.tipoBovino" required>
                                    <option ng-repeat="t in
animalTypes" value="{{t}}">{{t}}</option>
                                </select>
                            </div>
                        </div>
                    </div>

                    <!-- Imagem  NOT REQUIRED-->
                    <div class="form-group">
                        <div class="row">
                            <div class="col-md-12">
                                <label>Imagem</label>
                                <input type="file"
accept="image/*" fileread="inputImagem">
                            </div>
                        </div>
                    </div>

                    <div class="well">
                        <div class="form-group">
                            <div class="row">
                                <div class="col-md-12">
                                    <label>Número do
Reprodutor</label>
                                    <input class="form-
control" type="text" placeholder="Sem Registro" ng-model="form.parentNumero"
readonly>
                                </div>
                            </div>
                        </div>

                        <div class="form-group">
                            <div class="row">
                                <div class="col-md-12">
                                    <label>Código do
Reprodutor Externo</label>
                                    <input class="form-
control" type="text" placeholder="Sem Registro" ng-
model="form.parentReprodutorExterno" readonly>
                                </div>
                            </div>
                        </div>
```

```
                </div>

                <!-- Observação -->
                <div class="form-group">
                        <div class="row">
                                <div class="col-md-12">
                                        <label>Observação</label>
                                        <textarea class="form-control"
ng-model="form.observacao"></textarea>
                                </div>
                        </div>
                </div>

                <!-- ALERTS -->
                <div class="row">
                        <div class="col-md-12">
                                <div class="alert alert-success" ng-
show="alertSuccess">
                                        <button type="button"
class="close" ng-click="alertSuccess=false"><span>&times;</span></button>

        <strong>{{alertResMessage}}</strong>
                                </div>
                        </div>
                </div>

                <div class="row">
                        <div class="col-md-12">
                                <div class="alert alert-danger" ng-
show="alertDanger">
                                        <button type="button"
class="close" ng-click="alertDanger=false"><span>&times;</span></button>
                                        {{alertResMessage}}
                                </div>
                        </div>
                </div>
                <!-- ALERTS END -->

                <!-- Submit Button -->
                <div class="form-group">
                        <div class="row">
                                <div class="col-md-3">
                                        <button class="btn btn-primary
form-control" ng-click="editAnimal()">Editar</button>
                                </div>
                                <div class="col-md-3">
                                        <button class="btn btn-danger
form-control" ng-click="deleteAnimal()">Deletar</button>
                                </div>
                        </div>
                </div>
```

```
                        </form>

                    </div>
                </div>
            </div>

            <div class="container main-info" ng-show="tab == 'cruzas'">
                <div class="row cruza-totais" style="margin-bottom: 10px;">
                    <div class="col-md-4 open">
                        <span class="badge">{{cruzaOpenTotal}}</span> EM
GESTAÇÃO
                    </div>
                    <div class="col-md-4 parto">
                        <span class="badge">{{cruzaPartoTotal}}</span>
PARTOS REALIZADOS
                    </div>
                    <div class="col-md-4 abort">
                        <span class="badge">{{cruzaAbortTotal}}</span>
GESTAÇÕES ABORTADAS
                    </div>
                </div>
                <div class="row">
                    <div class="col-md-12">
                        <table class="table table-hover">
                            <tr>
                                <th>Status</th>
                                <th>Data Cruza</th>
                                <th>Data Parto</th>
                                <th>Cod Reprodutor</th>
                                <th>Nome Reprodutor</th>
                                <th>Observação</th>
                            </tr>
                            <tr ng-repeat="cruza in cruzas" ng-class="{success:
cruza.status=='parto', danger: cruza.status=='abort'}">
                                <td
class="status">{{statusLabel[cruza.status]}}</td>
                                <td class="date">{{cruza.dataCruza | date:
"dd/MM/y"}}</td>
                                <td class="date">{{cruza.dataParto | date:
"dd/MM/y"}}</td>
                                <td
class="reprodutor">{{cruza.codReprodutor}}</td>
                                <td
class="reprodutor">{{cruza.nomeReprodutor}}</td>
                                <td
class="reprodutor">{{cruza.observacao}}</td>
                            </tr>
                        </table>
                    </div>
                </div>
```

```
        </div>

        <div class="container main-info" ng-show="tab == 'tratamento'">

                <!--  Pontuais -->
                <div class="row" ng-show="tratamentosPontuais.length > 0">
                        <div class="col-md-12">
                                <h2>Tratamentos Pontuais</h2>
                        </div>
                        <div class="col-md-12">
                                <table class="table">
                                        <tr>
                                                <th>Tratamento</th>
                                                <th>Número</th>
                                                <th>Inicio do Tratamento</th>
                                                <th>Fim do Tratamento</th>
                                                <th>Frequência</th>
                                        </tr>
                                        <tr ng-repeat="tratamento in tratamentosPontuais">
                                                <td class="tratamento" ng-
click="$state.go('saude-animal-tratamento', {'id':
tratamento._id})">{{tratamento.nomeTratamento}}</td>
                                                <td
class="numero">{{tratamento.numero}}</td>
                                                <td>{{tratamento.timeStampInicio | date:
"dd/MM/y"}}</td>
                                                <td>{{tratamento.timeStampFim | date:
"dd/MM/y"}}</td>
                                                <td>A cada {{tratamento.frequencia}}
dia(s)</td>
                                        </tr>
                                </table>
                        </div>
                </div>

                <!-- Contínuos -->
                <div class="row" ng-show="tratamentosContinuos.length > 0">
                        <div class="col-md-12">
                                <h2>Tratamentos Contínuos</h2>
                        </div>
                        <div class="col-md-12">
                                <table class="table">
                                        <tr>
                                                <th>Tratamento</th>
                                                <th>Número</th>
                                                <th>Frequência</th>
                                        </tr>
                                        <tr ng-repeat="tratamento in
tratamentosContinuos">
```

```
                                        <td class="tratamento" ng-
click="$state.go('saude-animal-tratamento', {'id':
tratamento._id})">{{tratamento.nomeTratamento}}</td>
                                        <td
class="numero">{{tratamento.numero}}</td>
                                        <td>A cada {{tratamento.frequencia}}
dias</td>
                                    </tr>
                                </table>
                            </div>
                        </div>
                    </div>

                </div>


rebanho/rebanho.html

<div class="main-container">

    <div class="container container-title">
        <div class="row">
            <div class="col-sm-12">
                <h1>Rebanho</h1>
            </div>
        </div>
    </div>

    <div class="container container-menu">
        <div class="row">

            <div class="col-sm-3">
                <div class="form-group">
                    <div class="input-group">
                        <div class="input-group-addon"><span
class="glyphicon glyphicon-search"></span></div>
                        <input type="text" ng-model="numero"
class="form-control" placeholder="Número">
                    </div>
                </div>
            </div>

            <div class="col-sm-3">
                <div class="form-group">
                    <select class="form-control" ng-
model="tipoBovino">
                        <option value="all">Todos os Tipos</option>
                        <option ng-repeat="t in animalTypes"
value="{{t}}">{{t}}</option>
                    </select>
                </div>
```

```
                    </div>

                    <div class="col-sm-1">
                        <div class="form-group">
                            <button class="btn btn-primary form-control"
title="Cadastrar Animal" ng-click="cadastrarAnimal()"><span class="glyphicon
glyphicon-plus"></span></button>

                        </div>
                    </div>


                </div>
            </div>

            <div class="container container-animals">

                <div class="col-sm-4 animal-card" ng-repeat="animal in animais" ng-
show="(numero === animal.numero || !numero) && (tipoBovino ===
animal.tipoBovino || tipoBovino === 'all')">
                    <div class="panel panel-primary">
                        <div class="panel-heading">{{animal.nome}} <span
class="glyphicon glyphicon-cog" title="Edita Animal" ng-
click="edit(animal.numero)"></span> </div>
                        <div class="panel-body">
                            <div class="row">
                                <div class="col-sm-12">
                                    <img class="media-object"
src="{{animal.imagem}}" alt="No Image" height="240">
                                </div>
                            </div>
                            <div class="row">
                                <div class="col-sm-12">
                                    <div class="ficha"><span class="ficha-
name">Número</span> {{animal.numero}}</div>
                                </div>
                            </div>
                            <div class="row">
                                <div class="col-sm-12">
                                    <div class="ficha"><span class="ficha-
name">Sexo</span> {{animal.sexo}}</div>
                                </div>
                            </div>
                            <div class="row">
                                <div class="col-sm-12">
                                    <div class="ficha ficha-tipo"><span
class="ficha-name">Tipo</span> {{animal.tipoBovino}}</div>
                                </div>
                            </div>
                            <div class="row">
                                <div class="col-sm-12">
```

```
                                        <div class="ficha ficha-
dataNasc"><span class="ficha-name">Data de Nascimento</span>
{{animal.dataNasc}}</div>
                                    </div>
                                </div>
                            </div>

                        </div>
                    </div>

                </div>

</div>
```

register/RegisterController.js

```
(function (){
    "use strict";

    function RegisterController($scope, $state, AuthService){

        var tokenId = AuthService.getTokenId();

        initialize();
        function initialize(){
            if(tokenId) console.log(tokenId);
            $scope.passwdAlert = false;
        };

        $scope.registrar = function(){
            var user = $scope.form;

            if($scope.formRegister.$valid){

                // Check passwd
                if(user.passwd === user.repasswd)
                    AuthService.createAccount(user)
                                .then(
                                    function(res){

    $scope.alertResMessage = res.data.message;

    $scope.alertResMessageComplemento = "Volte para a tela de login.";

    $scope.alertResColor = "alert-success";
                                        $scope.alertRes =
true;
                                    },
                                    function(res){
```

```
        $scope.alertResMessage = res.data.message;

        $scope.alertResMessageComplemento = "Por favor, entre em contato
conosco.";

        $scope.alertResColor = "alert-danger";
                                                                $scope.alertRes =
true;
                                                                }
                                                        );
                                else
                                        $scope.passwdAlert = true;

                        }
                }

        };

        angular.module('ordenhaDigital').controller('RegisterController',
RegisterController);

})();
```

register/register.html

```
<div class="container">
        <div class="col-md-6 col-md-offset-3">
                <div class="panel">
                        <div class="panel-heading">
                                <h2>Registro</h2>
                        </div>
                        <div class="panel-body">
                                <form name="formRegister">

                                        <div class="form-group">
                                                <div class="row">
                                                        <div class="col-md-12">
                                                                <label>Nome Completo</label>
                                                                <input class="form-control"
type="text" placeholder="Nome Completo" ng-model="form.name" required>
                                                        </div>
                                                </div>
                                        </div>

                                        <div class="form-group">
                                                <div class="row">
                                                        <div class="col-md-12">
                                                                <label>CPF</label>
```

```
                                        <input class="form-control"
type="text" placeholder="CPF" ng-model="form.cpf" required>
                                    </div>
                                </div>
                            </div>

                            <div class="form-group">
                                <div class="row">
                                    <div class="col-md-12">
                                        <label>Email</label>
                                        <input class="form-control"
type="email" placeholder="Email" ng-model="form.email" ng-
pattern="/^[^\s@]+@[^\s@]+\.[^\s@]{2,}$/" required>
                                    </div>
                                </div>
                            </div>

                            <div class="form-group">
                                <div class="row">
                                    <div class="col-md-12">
                                        <label>Senha</label>
                                        <input class="form-control"
type="password" placeholder="Senha" ng-model="form.passwd" required>
                                    </div>
                                </div>
                            </div>

                            <div class="form-group">
                                <div class="row">
                                    <div class="col-md-12">
                                        <label>Repetir Senha</label>
                                        <input class="form-control"
type="password" placeholder="Repetir Senha" ng-model="form.repasswd" required>
                                    </div>
                                </div>
                            </div>

                            <!-- ALERTS -->

                            <div class="row">
                                <div class="col-sm-12">
                                    <div class="alert alert-danger" ng-
show="passwdAlert">
                                        <button type="button"
class="close" ng-click="passwdAlert=false"><span>&times;</span></button>
                                        <strong>Senhas
Incorretas!</strong> Por favor repita a senha corretamente.
                                    </div>
                                </div>
                            </div>
```

```
                                        <div class="row">
                                                <div class="col-sm-12">
                                                        <div class="alert {{alertResColor}}" ng-
show="alertRes">
                                                                <button type="button"
class="close" ng-click="alertRes=false"><span&times;</span></button>

        <strong>{{alertResMessage}}</strong> {{alertResMessageComplemento}}
                                                        </div>
                                                </div>
                                        </div>

                                        <!-- BUTTON -->

                                        <div class="form-group">
                                                <div class="row">
                                                        <div class="col-sm-12">
                                                                <button class="form-control btn
btn-primary" ng-click="registrar()">Registrar-se</input>
                                                        </div>
                                                </div>
                                        </div>

                                        <div class="row namerights">
                                                <div class="col-sm-12">
                                                        <small>2017 - Ordenha
Digital</small>
                                                </div>
                                        </div>

                                </form>
                        </div>
                </div>
        </div>
</div>
```

reproducao/ReproducaoCadastroController.js

```
(function (){
        "use strict";

        function Controller($scope, $state, AuthService, ReproducaoService,
RebanhoService){

                var tokenId = AuthService.getTokenId();

                initialize();
                function initialize(){
                        if(!tokenId) $state.go('login');
```

```
                $scope.alertSuccess = false;
                $scope.alertDanger = false;
                $scope.alertResMessage = "#";

                // Init form
                $scope.form = {
                        'dataCruza': new Date()
                }

                RebanhoService.callAnimals().then(function(res){
                        $scope.animals = res.data;
                });
        };

        $scope.cadastraCruza = function(){
                if($scope.formCruza.$valid){
                        var cruza = $scope.form;
                        ReproducaoService.createCruza(cruza)
                                .then(
                                        // Success
                                        function(res){
                                                $scope.alertSuccess = true;
                                                $scope.alertResMessage =
res.data.message;
                                        },
                                        // Error
                                        function(res){
                                                $scope.alertDanger = true;
                                                $scope.alertResMessage =
res.data.message;
                                        }
                                );
                }
                else{
                        $scope.alertDanger = true;
                        $scope.alertResMessage = "Preencha os campos
corretamente.";
                }

        };

};

angular.module('ordenhaDigital').controller('ReproducaoCadastroController',
Controller);

})();
```

reproducao/ReproducaoController.js

```
(function (){
```

```
"use strict";

function ReproducaoController($scope, $state, AuthService,
ReproducaoService){

	var tokenId = AuthService.getTokenId();

	initialize();
	function initialize(){
		if(!tokenId) $state.go('login');

		$scope.tab = "cruza";

		ReproducaoService.callCruzas().then(handleGetCruzas);

		$scope.cruzaStatus = ReproducaoService.getStatus();

		ReproducaoService.cruzaByMonth().then(handleCruzaByMonth);
		ReproducaoService.partoByMonth().then(handlePartoByMonth);
	};

	function handleGetCruzas(res){
		$scope.openStatusCounter = 0;
		$scope.partoStatusCounter = 0;
		$scope.abortStatusCounter = 0;

		var i;
		for(i in res.data){
			res.data[i].dataParto = new Date(res.data[i].dataParto);

			switch(res.data[i].status){
				case 'open':
					$scope.openStatusCounter++;
					break;
				case 'parto':
					$scope.partoStatusCounter++;
					break;
				case 'abort':
					$scope.abortStatusCounter++;
			};
		}

		$scope.cruzas = res.data;
	};

	$scope.editCruza = function(cruza){
		ReproducaoService.editCruza(cruza)
			.then(
				function(res){
					if(cruza.status === 'parto'){
```

```
                                        alert("Parto realizado! Cadastre o novo animal.");
                                        $state.go('rebanho-cadastro', {
                                                'parentNumero': cruza.numero,
                                                'parentReprodutorExterno': cruza.observacao+": "+cruza.codReprodutor+": "+cruza.nomeReprodutor,
                                                'dataNasc': cruza.dataParto
                                        });
                                }
                                else{
                                        alert(res.data.message);
                                        location.reload();
                                }
                        },
                        function(res){
                                alert(res.data.message);
                        }
                );
        };

        $scope.removeCruza = function(cruza, index){
                if(confirm("Tem certeza que quer remover a cruza "+index+"?"))
                        ReproducaoService.deleteCruza(cruza)
                                .then(
                                        function(res){
                                                alert(res.data.message);
                                                location.reload();
                                        },
                                        function(res){
                                                alert(res.data.message);
                                        }
                                );
        };

        $scope.novaCruza = function(){
                $state.go('reproducao-cadastro');
        };


        function generateDataProvider(data){
                var dataProvider = [];

                if(data.length > 0){
                        var firstNode = data[0]['_id'];
                        var lastNode  = data[data.length-1]['_id'];

                        var inicialDate = new Date(firstNode.year+"-"+firstNode.month+"-00:00:00");
                        var finalDate = new Date(lastNode.year+"-"+lastNode.month+"-00:00:00");
                        var today = new Date();
```

```
                        if(inicialDate > today)
                                inicialDate = new Date(today.getFullYear()+"-
"+(today.getMonth()+1)+"-00:00:00");

                        if(finalDate < today)
                                finalDate = new Date(today.getFullYear()+"-
"+(today.getMonth()+1)+"-00:00:00");


                        // Create auxDict
                        var auxDict = {};
                        while(inicialDate <= finalDate){
                                var key =  inicialDate.getFullYear() + "-" + ("0" +
((inicialDate).getMonth() + 1)).slice(-2);
                                auxDict[key] = {
                                        'timeStamp':
moment(inicialDate).format("MMM YYYY"),
                                        'qtd': 0
                                };

                                inicialDate.setMonth(inicialDate.getMonth() + 1);
                        }

                        // put qtds into auxDict
                        for(var i=0, d; d=data[i++];){
                                var key = d['_id'].year+"-"+("0" +
d['_id'].month).slice(-2);
                                auxDict[key].qtd = d.qtd;
                        }

                        var keys = Object.keys(auxDict);
                        keys.sort()

                        for(var i=0,k; k = keys[i++];)
                                dataProvider.push(auxDict[k]);
                }

                return dataProvider;
        }


        function handleCruzaByMonth(res){
                var data = res.data;
                $scope.cruzaByMonthFlag = false;

                if(data.length){
                        $scope.cruzaByMonthFlag = true;
                        var dataProvider = generateDataProvider(data);
                        generateChart('cruzaByMonth', 'Cruzas por Mês',
dataProvider);
```

```
		}
	};

	function handlePartoByMonth(res){
		var data = res.data;
		$scope.partoByMonthFlag = false;

		if(data.length){
			$scope.partoByMonthFlag = true;
			var dataProvider = generateDataProvider(data);
			generateChart('partoByMonth', 'Partos por Mês',
dataProvider);
		}

	};

	function generateChart(id, title, dataProvider){
		AmCharts.makeChart(id,
			{
				"type": "serial",
				"categoryField": "timeStamp",
				"startDuration": 1,
				"theme": "dark",
				"categoryAxis": {
					"gridPosition": "start"
				},
				"chartCursor": {
			    "fullWidth":true,
			    "valueLineEabled":true,
			    "valueLineBalloonEnabled":true,
			    "valueLineAlpha":0.5,
			    "cursorAlpha":0
			  },
				"trendLines": [],
				"graphs": [
					{
						"balloonText": "[[qtd]]",
						"bullet": "round",
						"id": "AmGraph-1",
						"title": title,
						"valueField": "qtd"
					}
				],
				"guides": [],
				"valueAxes": [
					{
						"id": "ValueAxis-1",
						"title": "Quantidade",
						"integersOnly": true
					}
				],
```

```
                                        "allLabels": [],
                                        "balloon": {},
                                        "legend": {
                                                "enabled": true,
                                                "useGraphSettings": true
                                        },
                                        "titles": [
                                                {
                                                        "id": "Title-1",
                                                        "size": 15,
                                                        "text": title
                                                }
                                        ],
                                        "dataProvider": dataProvider

                                }
                        );
                };
        };

        angular.module('ordenhaDigital').controller('ReproducaoController',
ReproducaoController);

})();
```

reproducao/cadastro.html

```
<div class="main-container">

        <div class="container container-title">
                <div class="row">
                        <div class="col-sm-12">
                                <h1>Nova Cruza</h1>
                        </div>
                </div>
        </div>

        <div class="container">

                <form name="formCruza">
                        <div class="row">
                                <!-- Lateral ESQUERDA -->
                                <div class="col-md-6">

                                        <div class="well">

                                                <div class="form-group">
                                                        <div class="row">
                                                                <div class="col-md-12">
                                                                        <label>Código do
Reprodutor Externo</label>
```

```
                                                    <input class="form-
control" type="text" ng-model="form.codReprodutor" required></input>
                                                </div>
                                            </div>
                                        </div>

                                        <div class="form-group">
                                            <div class="row">
                                                <div class="col-md-12">
                                                    <label>Nome do
Reprodutor Externo</label>
                                                    <input class="form-
control" type="text" ng-model="form.nomeReprodutor" required></input>
                                                </div>
                                            </div>
                                        </div>

                                        <div class="form-group">
                                            <div class="row">
                                                <div class="col-md-12">

        <label>Observação</label>
                                                    <input class="form-
control" type="text" ng-model="form.observacao"></input>
                                                </div>
                                            </div>
                                        </div>

                                    </div>


                                </div>

                                <!-- Lateral DIREITA -->
                                <div class="col-md-6">

                                    <div class="form-group">
                                        <div class="row">
                                            <div class="col-md-12">
                                                <label>Número da
Reprodutora</label>
                                                <select class="form-control" ng-
model="form.numero" required>
                                                    <option ng-
repeat="animal in animals" value="{{animal.numero}}">{{animal.numero}}</option>
                                                </select>
                                            </div>
                                        </div>
                                    </div>

                                    <div class="form-group">
```

```
                                        <div class="row">
                                                <div class="col-md-12">
                                                        <label>Data da Cruza</label>
                                                        <input class="form-control"
type="date" ng-model="form.dataCruza" required></input>
                                                </div>
                                        </div>
                                </div>

                                <!-- ALERTS -->
                                <div class="row">
                                        <div class="col-md-12">
                                                <div class="alert alert-success" ng-
show="alertSuccess">
                                                        <button type="button"
class="close" ng-click="alertSuccess=false"><span>&times;</span></button>

        <strong>{{alertResMessage}}</strong>
                                                </div>
                                        </div>
                                </div>

                                <div class="row">
                                        <div class="col-md-12">
                                                <div class="alert alert-danger" ng-
show="alertDanger">
                                                        <button type="button"
class="close" ng-click="alertDanger=false"><span>&times;</span></button>
                                                        {{alertResMessage}}
                                                </div>
                                        </div>
                                </div>
                                <!-- ALERTS END -->


                                <!-- SUBMIT -->
                                <div class="form-group">
                                        <div class="row">
                                                <div class="col-md-12">
                                                        <button class="btn btn-primary
form-control" ng-click="cadastraCruza()">Cadastrar</button>
                                                </div>
                                        </div>
                                </div>

                        </div>
                </div>
        </form>

        </div>
</div>
```

reproducao/reproducao.html

```
<div class="main-container">

	<div class="container container-title">
		<div class="row">
			<div class="col-sm-12">
				<h1>Reprodução</h1>
			</div>
		</div>
	</div>

	<!-- Nav Pills -->
	<div class="container" style="margin-bottom: 40px;">
		<div class="row">
			<!-- NAV -->

				<ul class="nav nav-pills">
					<li class="col-md-2" ng-class="tab == 'cruza' ?
'active' : ""><a data-toggle="tab" ng-click="tab = 'cruza'">Cruzas</a></li>
					<li class="col-md-2" ng-class="tab ==
'cruzaByMonth' ? 'active' : ""><a data-toggle="tab" ng-click="tab =
'cruzaByMonth'">Cruzas por Mês</a></li>
					<li class="col-md-2" ng-class="tab ==
'partoByMonth' ? 'active' : ""><a data-toggle="tab" ng-click="tab =
'partoByMonth'">Partos por Mês</a></li>
				</ul>

		</div>
	</div>


	<div class="container" ng-show="tab == 'cruza' ">
		<div class="row" style="margin-bottom: 5px;">
			<!-- Cruzas counts info -->
			<!-- DESKTOP -->
			<div class="col-md-2 statusCount onlyDesktop">
				<div class="row">
					<span class="label label-default col-md-10">
						Em Gestação <span
class="counter">{{openStatusCounter}}</span>
					</span>
				</div>

				<div class="row">
					<span class="label label-success col-md-10">
						Partos <span
class="counter">{{partoStatusCounter}}</span>
					</span>
				</div>
```

```
                        <div class="row">
                            <span class="label label-danger col-md-10">
                                Abortos <span
class="counter">{{abortStatusCounter}}</span>
                            </span>
                        </div>
                    </div>

                    <!-- MOBILE -->
                    <div class="col-md-2 onlyMobile counters" style="margin-bottom:
5px;">
                        <span class="badge">{{openStatusCounter}}</span> Em
Gestação<br/>
                        <span class="parto"><span
class="badge">{{partoStatusCounter}}</span> Partos</span><br/>
                        <span class="abort"><span
class="badge">{{abortStatusCounter}}</span> Abortos</span>
                    </div>

                    <!-- Nova Cruza button -->
                    <div class="col-md-offset-8 col-md-2">
                        <button class="btn btn-primary" style="width: 100%;"
title="Nova Cruza" ng-click="novaCruza()">
                            <span class="glyphicon glyphicon-plus"></span>
                        </button>
                    </div>
                </div>

                <div class="row">
                    <div class="col-md-12">

                        <table class="table">
                            <tr>
                                <th>Cruza</th>
                                <th>Número</th>
                                <th>Cod Reprodutor</th>
                                <th>Nome Reprodutor</th>
                                <th>Observação</th>
                                <th>Data Cruza</th>
                                <th>Data Parto</th>
                                <th>Status</th>
                                <th></th>
                                <th></th>
                            </tr>
                            <tr ng-repeat="cruza in cruzas" ng-class="{success:
cruza.status=='parto', danger: cruza.status=='abort'}">
                                <td class="index">{{$index + 1}}</td>
                                <td class="numero" ng-
click="$state.go('rebanho-edit', {'numero': cruza.numero, 'tab':
'cruzas'})">{{cruza.numero}}</td>
```

```
                                        <td
class="reprodutor">{{cruza.codReprodutor}}</td>
                                        <td
class="reprodutor">{{cruza.nomeReprodutor}}</td>
                                        <td
class="reprodutor">{{cruza.observacao}}</td>
                                        <td class="date">{{cruza.dataCruza | date:
"dd/MM/y"}}</td>
                                        <td class="date">
                                                <input class="form-control"
type="date" ng-model="cruza.dataParto"></input>
                                        </td>
                                        <td class="status">
                                                <select class="form-control" ng-
model="cruza.status">
                                                        <option ng-repeat="s in
cruzaStatus" ng-selected="cruza.status == s.status"
value="{{s.status}}">{{s.statusName}}</option>
                                                </select>
                                        </td>
                                        <td class="edit">
                                                <button class="btn btn-default" ng-
click="editCruza(cruza)" title="Edit Cruza">
                                                        <span class="glyphicon
glyphicon-floppy-disk"></span>
                                                </button>
                                        </td>

                                        <td class="remove">
                                                <button class="btn btn-default" ng-
click="removeCruza(cruza, $index+1)" title="Remova Cruza">
                                                        <span class="glyphicon
glyphicon-trash"></span>
                                                </button>
                                        </td>

                                </tr>
                        </table>

                </div>
        </div>

    </div>

    <div class="container" ng-show="tab == 'cruzaByMonth'">
            <div class="row"  ng-show="cruzaByMonthFlag">
                    <div class="col-md-12">
                            <div id="cruzaByMonth" style="width: 100%; height:
400px; background-color: #282828;" ></div>
                    </div>
            </div>
```

```
                <div class="row"  ng-show="!cruzaByMonthFlag">
                        <div class="col-md-12">
                                <div class="alert alert-warning"><strong>Ainda não
existem cruzas!</strong></div>
                        </div>
                </div>
        </div>

        <div class="container" ng-show="tab == 'partoByMonth'">
                <div class="row" ng-show="partoByMonthFlag">
                        <div class="col-md-12">
                                <div id="partoByMonth" style="width: 100%; height: 400px;
background-color: #282828;" ></div>
                        </div>
                </div>
                <div class="row" ng-show="!partoByMonthFlag">
                        <div class="col-md-12">
                                <div class="alert alert-warning"><strong>Ainda não
existem partos realizados!</strong></div>
                        </div>
                </div>
        </div>


</div>
```

saudeAnimal/SaudeAnimalCadastroController.js

```
(function (){
        "use strict";

        function Controller($scope, $state, AuthService, SaudeAnimalService,
RebanhoService){

                var tokenId = AuthService.getTokenId();

                initialize();
                function initialize(){
                        if(!tokenId) $state.go('login');

                        // Alert default values
                        $scope.alertSuccess = false;
                        $scope.alertError = false;
                        $scope.alertResMessage = "";

                        // Form default values
                        formDefault();

                        RebanhoService.callAnimals().then(handleCallAnimals);

                };
```

```
		function formDefault(){
			$scope.form = {};
			$scope.form.timeStampInicio = new Date();
			$scope.form.timeStampInicio.setHours(0,0,0);
			$scope.form.frequencia = 1;
			$scope.numeroEventos = 1;
		};


		function handleCallAnimals(res){
			var animals = res.data;

			$scope.animalsByNumber = {};
			$scope.numeros = [];

			for(var i=0, animal; animal=animals[i++];){
				$scope.numeros.push(animal.numero);
				$scope.animalsByNumber[animal.numero] = animal;
			}
		}

		$scope.handleChangeNumber = function(){
			var numero = $scope.form.numero;
			$scope.imagem_blob =
$scope.animalsByNumber[numero].imagem;
		};

		$scope.cadastraTratamento = function(){
			var tratamento = $scope.form;

			// Confere campos
			if(!('numero' in tratamento)){
				$scope.alertError = true;
				$scope.alertResMessage = "Selecione um animal!";
			}
			else if(!('nomeTratamento' in tratamento)){
				$scope.alertError = true;
				$scope.alertResMessage = "Dê um nome ao tratamento!";
			}
			else if(!('tipoTratamento' in tratamento)){
				$scope.alertError = true;
				$scope.alertResMessage = "Escolha o tipo do
tratamento!";
			}
			else if(!('timeStampInicio' in tratamento)){
				$scope.alertError = true;
				$scope.alertResMessage = "Escolha uma data de início!";
			}
			else if(tratamento.tipoTratamento == "pontual" &&
$scope.numeroEventos < 1){
```

```
                $scope.alertError = true;
                $scope.alertResMessage = "Tratamentos pontuais devem
possuir um número de eventos válido!";
            }
            else if(!('frequencia' in tratamento) || tratamento.frequencia < 1){
                $scope.alertError = true;
                $scope.alertResMessage = "A frequencia deve ser de no
mínimo 1 dia!";
            }
            // Tudo ok!
            else{

                // Calc timeStampFim
                if(tratamento.tipoTratamento == "pontual"){
                    var gap    =
parseInt(tratamento.frequencia)*(parseInt($scope.numeroEventos)-1);
                    var fim = new Date(tratamento.timeStampInicio);
                    fim.setDate(fim.getDate()+gap);
                    tratamento.timeStampFim = fim;
                }

                tratamento.eventos = criaEventos(tratamento);
                console.log(tratamento);

                SaudeAnimalService.createTratamento(tratamento)
                    .then(
                        function(res){
                            $scope.alertSuccess = true;
                            $scope.alertResMessage =
res.data.message;
                            formDefault();
                        },
                        function(res){
                            $scope.alertError = true;
                            $scope.alertResMessage =
res.data.message;
                        }
                    );
            }
        };

        function criaEventos(tratamento){
            var eventos = [];

            var ponteiroData = new Date(tratamento.timeStampInicio);

            //  Tratamento PONTUAL
            if(tratamento.tipoTratamento == "pontual"){
                for(var i=0; i<$scope.numeroEventos; i++){
                    eventos.push(generateEvento(ponteiroData));
```

```
                                        ponteiroData.setDate(ponteiroData.getDate() +
parseInt(tratamento.frequencia));
                                }
                                // var dateFim = tratamento.timeStampFim;
                                // while(ponteiroData < dateFim){
                                //      eventos.push(generateEvento(ponteiroData));
                                //      ponteiroData.setDate(ponteiroData.getDate() +
parseInt(tratamento.frequencia));
                                // } // While
                                // eventos.push(generateEvento(dateFim));
                        }

                        // Tratamento CONTÍNUO
                        else {
                                eventos.push(generateEvento(ponteiroData));
                        }

                        return eventos;
                };

                function generateEvento(timeStamp){
                        return {
                                        'timeStamp': new Date(timeStamp),
                                        'status': 'open',
                                        'observacao': ''
                                };
                };

        };

        angular.module('ordenhaDigital').controller('SaudeAnimalCadastroController',
Controller);

})();
```

saudeAnimal/saudeAnimalController.js

```
(function (){
        "use strict";

        function Controller($scope, $state, AuthService, SaudeAnimalService){

                var tokenId = AuthService.getTokenId();

                initialize();
                function initialize(){
                        if(!tokenId) $state.go('login');

                        $scope.tab = "continuo";
```

```
        SaudeAnimalService.callTratamentos().then(handleCallTratamentos)
                };


                function handleCallTratamentos(res){
                        var tratamentos = res.data;

                        $scope.tratamentosContinuos = [];
                        $scope.tratamentosPontuais  = [];

                        var t;
                        for(var i=0; t=tratamentos[i++];){
                                if(t.tipoTratamento === 'continuo')
                                        $scope.tratamentosContinuos.push(t)
                                else
                                        $scope.tratamentosPontuais.push(t);
                        }
                };

                $scope.specificTratamento = function(id){
                        // console.log(id);
                        $state.go('saude-animal-tratamento', {'id': id});
                }
        };

        angular.module('ordenhaDigital').controller('SaudeAnimalController',
Controller);

})();
```

saudeAnimal/saudeAnimalTratamentoController.js

```
(function (){
        "use strict";

        function Controller($scope, $state, AuthService, SaudeAnimalService,
RebanhoService){

                var tokenId = AuthService.getTokenId();

                initialize();
                function initialize(){
                        if(!tokenId) $state.go('login');

                        var id = $state.params.id;
                        SaudeAnimalService.callTratamento(id).then(handleTratamento);
                };

                function handleTratamento(res){
                        $scope.tratamento = res.data[0];
```

```
                // console.log($scope.tratamento);
                var evento;
                for(var i=0; evento=$scope.tratamento.eventos[i++];)
                        evento.timeStamp = new Date(evento.timeStamp);


        RebanhoService.callAnimal($scope.tratamento.numero).then(handleAnimal);
        };

        function handleAnimal(res){
                $scope.animal = res.data;
                // console.log($scope.animal);
        };

        $scope.editEvento = function(){
                var tratamentoId = $state.params.id;
                SaudeAnimalService.editTratamento($scope.tratamento).then(
                        function(res){
                                alert(res.data.message);
                                location.reload();
                        },
                        function(res){
                                alert(res.data.message);
                                location.reload();
                        }
                );
        };
};

angular.module('ordenhaDigital').controller('SaudeAnimalTratamentoController
', Controller);

})();
```

saudeAnimal/cadastro.html

```
<div class="main-container">

        <div class="container container-title">
                <div class="row">
                        <div class="col-sm-12">
                                <h1>Novo Tratamento</h1>
                        </div>
                </div>
        </div>


        <div class="container">

                        <div class="row">
```

```
                    <!-- Roberto Carlos -->
                    <div class="col-md-6 onlyDesktop">
                        <div class="col-md-12">
                            <img class="media-object"
src="{{imagem_blob}}" width="100%" height="340">
                        </div>
                    </div>

                    <!-- Cafú -->
                    <div class="col-md-6">

                        <form name="formTratamento">

                            <div class="col-md-12">
                                <div class="form-group">
                                    <label>Número</label>
                                    <select class="form-control" ng-
model="form.numero" ng-change="handleChangeNumber()" required>
                                        <option ng-
repeat="numero in numeros" value="{{numero}}">{{numero}}</option>
                                    </select>
                                </div>
                            </div>

                            <div class="col-md-12">
                                <div class="form-group">
                                    <label>Nome do
Tratamento</label>
                                    <input class="form-control"
type="text" ng-model="form.nomeTratamento" required></input>
                                </div>
                            </div>

                            <div class="col-md-12">
                                <div class="form-group">
                                    <label>Tipo do
Tratamento</label>
                                    <select class="form-control" ng-
model="form.tipoTratamento" required>
                                        <option
value="continuo">Contínuo</option>
                                        <option
value="pontual">Pontual</option>
                                    </select>
                                </div>
                            </div>

                            <div class="col-md-12" ng-
show="form.tipoTratamento == 'pontual'">
                                <div class="form-group">
```

```
                                        <label>Números de Eventos</label>
                                        <div class="input-group">
                                            <input class="form-control" type="number" ng-model="numeroEventos"></input>
                                            <div class="input-group-addon">evento(s)</div>
                                        </div>
                                    </div>
                                </div>

                                <div class="col-md-12">
                                    <div class="form-group">
                                        <label>Data de Início</label>
                                        <input class="form-control" type="date" ng-model="form.timeStampInicio" required></input>
                                    </div>
                                </div>



                                <div class="col-md-12">
                                    <div class="form-group">
                                        <label>Frequência</label>
                                        <div class="input-group">
                                            <input class="form-control" type="number" ng-model="form.frequencia" required></input>
                                            <div class="input-group-addon">dia(s)</div>
                                        </div>
                                    </div>
                                </div>

                                <div class="col-md-12">
                                    <div class="form-group">
                                        <label>Observação</label>
                                        <textarea class="form-control" ng-model="form.observacao"></textarea>
                                    </div>
                                </div>
                            </form>

                            <!-- Alerts -->

                            <div class="col-md-12">
                                <div class="alert alert-success" ng-show="alertSuccess">
                                    <button type="button" class="close" ng-click="alertSuccess=false"><span>&times;</span></button>

                                    <strong>{{alertResMessage}}</strong>
```

```
                                        </div>
                                    </div>

                                    <div class="col-md-12">
                                        <div class="alert alert-danger" ng-
show="alertError">
                                            <button type="button" class="close"
ng-click="alertError=false"><span&times;</span></button>

        <strong>{{alertResMessage}}</strong>
                                        </div>
                                    </div>

                                    <!-- Submit Button -->
                                    <div class="form-group">
                                        <div class="col-md-3">
                                            <button class="btn btn-primary form-
control" ng-click="alertSuccess=false; alertError=false;
cadastraTratamento()">Cadastrar</button>
                                        </div>

                                    </div>

                                <!-- fim Cafú -->
                                </div>

                        </div>

        </div>

</div>
```

saudeAnimal/saudeAnimal.html

```
<div class="main-container">

        <div class="container container-title">
                <div class="row">
                        <div class="col-sm-12">
                                <h1>Tratamentos</h1>
                        </div>
                </div>
        </div>

        <!-- Nav Pills -->
        <div class="container" style="margin-bottom: 40px;">
                <div class="row">
                        <!-- NAV -->

                                <ul class="nav nav-pills">
```

```
                                    <li class="col-md-3" ng-class="tab == 'continuo' ? 3
'active' : ""><a data-toggle="tab" ng-click="tab = 'continuo'">Tratamentos
Contínuos</a></li>
                                    <li class="col-md-3" ng-class="tab == 'pontual' ?
'active' : ""><a data-toggle="tab" ng-click="tab = 'pontual'">Tratamentos
Pontuais</a></li>
                                </ul>
                        </div>
                </div>

                <div class="container">
                        <div class="row">
                                <div class="col-md-2 col-md-offset-10">
                                        <button class="btn btn-primary" style="width:100%;" ng-
click="$state.go('saude-animal-cadastro')" title="Novo Tratamento">
                                                <span class="glyphicon glyphicon-plus"></span>
                                        </button>
                                </div>
                        </div>
                </div>

                <div class="container" ng-show="tab == 'continuo' ">
                        <div class="row">
                                <div class="col-md-12" ng-show="tratamentosContinuos.length <
1">
                                        <div class="alert alert-warning"><strong>Aviso!</strong>
Ainda não existem tratamentos contínuos.</div>
                                </div>

                                <div class="col-md-12" ng-show="tratamentosContinuos.length >
0">
                                        <table class="table">
                                                <tr>
                                                        <th>Tratamento</th>
                                                        <th>Número</th>
                                                        <th>Frequência</th>
                                                </tr>
                                                <tr ng-repeat="tratamento in
tratamentosContinuos">
                                                        <td class="tratamento" ng-
click="specificTratamento(tratamento._id)">{{tratamento.nomeTratamento}}</td>
                                                        <td class="numero" ng-
click="$state.go('rebanho-edit', {'numero': tratamento.numero, 'tab':
'tratamento'})">{{tratamento.numero}}</td>
                                                        <td>A cada {{tratamento.frequencia}}
dias</td>
                                                </tr>
                                        </table>
                                </div>
                        </div>
                </div>
```

```
        <div class="container" ng-show="tab == 'pontual' ">
                <div class="row">
                <div class="col-md-12" ng-show="tratamentosPontuais.length < 1">
                        <div class="alert alert-warning"><strong>Aviso!</strong>
Ainda não existem tratamentos Pontuais.</div>
                </div>

                <div class="col-md-12" ng-show="tratamentosPontuais.length >
0">
                        <table class="table">
                                <tr>
                                        <th>Tratamento</th>
                                        <th>Número</th>
                                        <th>Inicio do Tratamento</th>
                                        <th>Fim do Tratamento</th>
                                        <th>Frequência</th>
                                </tr>
                                <tr ng-repeat="tratamento in tratamentosPontuais">
                                        <td class="tratamento" ng-
click="specificTratamento(tratamento._id)">{{tratamento.nomeTratamento}}</td>
                                        <td class="numero" ng-
click="$state.go('rebanho-edit', {'numero': tratamento.numero, 'tab':
'tratamento'})">{{tratamento.numero}}</td>
                                        <td>{{tratamento.timeStampInicio | date:
"dd/MM/y"}}</td>
                                        <td>{{tratamento.timeStampFim | date:
"dd/MM/y"}}</td>
                                        <td>A cada {{tratamento.frequencia}}
dia(s)</td>
                                </tr>
                        </table>
                </div>
        </div>
        </div>

</div>


saudeAnimal/saudeAnimalTratamento.html

<div class="main-container">

        <div class="container container-title">
                <div class="row">
                        <div class="col-sm-12">
                                <h1>{{tratamento.nomeTratamento}}</h1>
                        </div>
                </div>
        </div>
```

```
        <div class="container">
                <div class="row">
                        <div class="col-md-5">
                                <img class="media-object onlyDesktop"
src="{{animal.imagem}}" alt="" height="340" width="100%">
                        </div>

                        <div class="col-md-6 col-md-offset-1">
                                <div class="form-group">
                                        <label>Animal</label>
                                        <div class="form-info">{{animal.numero}} -
{{animal.nome}}</div>
                                </div>
                                <div class="form-group">
                                        <label>Início do Tratamento</label>
                                        <div class="form-info">{{tratamento.timeStampInicio
| date: "dd/MM/y"}}</div>
                                </div>
                                <div class="form-group">
                                        <label>Término do Tratamento</label>
                                        <div class="form-info" ng-
show="tratamento.tipoTratamento == 'pontual'">{{tratamento.timeStampFim | date:
"dd/MM/y"}}</div>
                                        <div class="form-info" ng-
show="tratamento.tipoTratamento == 'continuo'">O tratamento é Contínuo</div>
                                </div>
                                <div class="form-group">
                                        <label>Frequência dos eventos</label>
                                        <div class="form-info">A cada
{{tratamento.frequencia}} dia(s)</div>
                                </div>

                                <div class="form-group">
                                        <label>Observação</label>
                                        <div class="form-
info">{{tratamento.observacao}}</div>
                                </div>

                        </div>

                </div>

        </div>
        <div class="container">
                <div class="row">
                        <div class="col-sm-12">
                                <h1>Eventos</h1>
                        </div>
                </div>

                <div class="row">
```

```
                    <div class="col-md-12">
                        <table class="table">
                            <tr ng-repeat="evento in tratamento.eventos" ng-
class="{success: evento.status=='ok', danger: evento.status=='abort'}">
                                <!-- <td>{{evento.timeStamp | date:
"dd/MM/y"}}</td> -->
                                <td>
                                    <input class="form-control"
type="date" ng-model="evento.timeStamp"></input>
                                </td>

                                <td style="width: 50%;">
                                    <input class="form-control" type="text"
placeholder="Observação" ng-model="evento.observacao"></input>
                                </td>

                                <td>
                                    <select class="form-control" ng-
model="evento.status">
                                        <option
value="open">Aguardando</option>
                                        <option
value="ok">Realizado</option>
                                        <option
value="abort">Abortado</option>
                                    </select>
                                </td>
                            </tr>
                        </table>
                    </div>
                </div>
                <div class="row">
                    <div class="col-md-2 col-md-offset-10">
                        <button class="btn btn-primary" style="width:100%;"
title="Atualizar" ng-click="editEvento()">
                            <span class="glyphicon glyphicon-floppy-
disk"></span>
                        </button>
                    </div>
                </div>
            </div>

</div>
```

public/directives/

fileRead.js

```
angular.module('ordenhaDigital').directive("fileread", [function () {
  return {
    scope: {
```

```
            fileread: "="
        },
        link: function (scope, element, attributes) {
            element.bind("change", function (changeEvent) {
                var reader = new FileReader();
                reader.onload = function (loadEvent) {
                    scope.$apply(function () {
                        scope.fileread.result = loadEvent.target.result;
                        scope.fileread.size = changeEvent.target.files[0].size;
                    });
                }
                reader.readAsDataURL(changeEvent.target.files[0]);
            });
        }
    }
}]);
```

public/services/

AuthService.js

```
"use strict";
(function(){

        function AuthService($http){

                var authservice = {
                        login: login,
                        logout: logout,
                        createAccount: createAccount,
                        getTokenId: getTokenId,
                        setTokenId: setTokenId
                };

                return authservice;

                function login(form){
                        return $http.post('/login', { "data": form })
                };

                function logout(){
                        return $http.get('/logout', {
                                headers: {
                                        'x-access-token': getTokenId()
                                }
                        })
                };

                function createAccount(user){
                        return $http.post('/user', { "data": user } )
                };
```

```
        function getTokenId(){
            var token = localStorage.getItem('profile');
            return token;
        };

        function setTokenId(token){
            localStorage.setItem('profile', token);
            return;
        }

    };

    angular.module('ordenhaDigital').factory('AuthService', AuthService);

})();
```

OrdenhaService.js

```
"use strict";
(function(){

    function OrdenhaService($http, AuthService){

        var ordenhaservice = {
            callOrdenhas: callOrdenhas,
            callOrdenhasByMonth: callOrdenhasByMonth,
            storeOrdenha: storeOrdenha,
            removeOrdenha: removeOrdenha,
            findOrdenhasWeek: findOrdenhasWeek,
            upsertMultiOrdenhas: upsertMultiOrdenhas,
            ordenhaByTimeStamp: ordenhaByTimeStamp
        };

        return ordenhaservice;

        function callOrdenhas(){
            return $http.get('/ordenha', {
                headers: {
                    'x-access-token': AuthService.getTokenId()
                }
            });
        };


        function callOrdenhasByMonth(){
            return $http.get('/ordenhas-by-month', {
                headers: {
                    'x-access-token': AuthService.getTokenId()
                }
```

```
                });
        };

        function storeOrdenha(ordenha){
                return $http.post('/ordenha', { "ordenha": ordenha }, {
                        headers: {
                                'x-access-token': AuthService.getTokenId()
                        }
                });
        };

        function removeOrdenha(id){
                return $http.delete('/ordenha?id='+id, {
                        headers: {
                                'x-access-token': AuthService.getTokenId()
                        }
                });
        };

        function ordenhaByTimeStamp(timeStamp){
                return $http.get('/ordenha-by-
timestamp?timeStamp='+timeStamp, {
                        headers: {
                                'x-access-token': AuthService.getTokenId()
                        }
                });
        };

        function findOrdenhasWeek(){
                return $http.get('/ordenha-multi', {
                        headers: {
                                'x-access-token': AuthService.getTokenId()
                        }
                });
        };

        function upsertMultiOrdenhas(ordenhas){
                return $http.post('/ordenha-multi', {'ordenhas': ordenhas}, {
                        headers: {
                                'x-access-token': AuthService.getTokenId()
                        }
                });
        };

    };

    angular.module('ordenhaDigital').factory('OrdenhaService', OrdenhaService);

})();
```

RebanhoService.js

```
"use strict";
(function(){

	function RebanhoService($http, AuthService){

		var rebanhoservice = {
			callAnimals: callAnimals,
			animalTypes: animalTypes,
			cadastraAnimal: cadastraAnimal,
			callAnimal: callAnimal,
			editAnimal: editAnimal,
			deleteAnimal: deleteAnimal,
			callVacaDeLeite: callVacaDeLeite
		};

		return rebanhoservice;

		function animalTypes(){
			return [
				"Vaca de Leite",
				"Bezerro",
				"Reprodutor",
				"Novilho"
			];
		};

		function callAnimals(){
			return $http.get('/animal', {
				headers: {
					'x-access-token': AuthService.getTokenId()
				}
			})
		};

		function cadastraAnimal(animal){
			return $http.post('/animal', { "animal": animal },{
				headers: {
					'x-access-token': AuthService.getTokenId()
				}
			});
		}

		function callAnimal(numero){
			return $http.get('/animal/'+numero,{
				headers: {
					'x-access-token': AuthService.getTokenId()
				}
			});
		}
```

```
		function editAnimal(numero, animal){
			return $http.put('/animal/'+numero, {'animal': animal}, {
				headers: {
					'x-access-token': AuthService.getTokenId()
				}
			});
		}

		function deleteAnimal(numero){
			return $http.delete('/animal/'+numero, {
				headers: {
					'x-access-token': AuthService.getTokenId()
				}
			});
		}

		function callVacaDeLeite(){
			return $http.get('/animal-de-leite',{
				headers: {
					'x-access-token': AuthService.getTokenId()
				}
			});
		}

	};

	angular.module('ordenhaDigital').factory('RebanhoService', RebanhoService);

})();
```

ReproducaoService.js

```
"use strict";
(function(){

	function ReproducaoService($http, AuthService){

		var reproducaoservice = {
			getStatus: getStatus,
			callCruzas: callCruzas,
			editCruza: editCruza,
			deleteCruza: deleteCruza,
			createCruza: createCruza,
			cruzaByMonth: cruzaByMonth,
			partoByMonth: partoByMonth,
			callCruzasForSpecificAnimal: callCruzasForSpecificAnimal
		};

		return reproducaoservice;
```

```
function getStatus(){
	return [
		{'status': 'open', 'statusName': 'Em gestação'},
		{'status': 'parto', 'statusName': 'Parto realizado'},
		{'status': 'abort', 'statusName': 'Gestação abortada'},
	];
};

function callCruzas() {
	return $http.get('/cruza', {
		headers: {
			'x-access-token': AuthService.getTokenId()
		}
	});
};

function callCruzasForSpecificAnimal(numero){
	return $http.get('/cruza/'+numero , {
		headers: {
			'x-access-token': AuthService.getTokenId()
		}
	});
};

function editCruza(cruza) {
	return $http.put('/cruza?id='+cruza['_id'], {'cruza': cruza}, {
		headers: {
			'x-access-token': AuthService.getTokenId()
		}
	});
};

function deleteCruza(cruza) {
	return $http.delete('/cruza?id='+cruza['_id'], {
		headers: {
			'x-access-token': AuthService.getTokenId()
		}
	});
};

function createCruza(cruza) {
	return $http.post('/cruza', {'cruza': cruza}, {
		headers: {
			'x-access-token': AuthService.getTokenId()
		}
	});
};

function cruzaByMonth() {
	return $http.get('/cruza/cruza/bymonth', {
```

```
                headers: {
                    'x-access-token': AuthService.getTokenId()
                }
            });
        };

        function partoByMonth() {
            return $http.get('/cruza/parto/bymonth', {
                headers: {
                    'x-access-token': AuthService.getTokenId()
                }
            });
        };

    };

    angular.module('ordenhaDigital').factory('ReproducaoService',
ReproducaoService);

})();
```

SaudeAnimalService.js

```
"use strict";
(function(){

    function Service($http, AuthService){

        var service = {
            callTratamentos: callTratamentos,
            callTratamento: callTratamento,
            editTratamento: editTratamento,
            createTratamento: createTratamento,
            getTratamentosByAnimal: getTratamentosByAnimal,
            proximosTratamentosAbertos: proximosTratamentosAbertos
        };

        return service;


        function callTratamentos() {
            return $http.get('/tratamento', {
                headers: {
                    'x-access-token': AuthService.getTokenId()
                }
            });
        };

        function callTratamento(id) {
            return $http.get('/tratamento/'+id, {
```

```
                headers: {
                    'x-access-token': AuthService.getTokenId()
                }
            });
        };

        function editTratamento(tratamento) {
            return $http.put('/tratamento-eventos', {'tratamento': tratamento},
{
                headers: {
                    'x-access-token': AuthService.getTokenId()
                }
            });
        };

        function createTratamento(tratamento){
            return $http.post('/tratamento', {'tratamento': tratamento}, {
                headers: {
                    'x-access-token': AuthService.getTokenId()
                }
            });
        };

        function getTratamentosByAnimal(numero){
            return $http.get('/tratamento-animal/'+numero, {
                headers: {
                    'x-access-token': AuthService.getTokenId()
                }
            });
        };

        function  proximosTratamentosAbertos(days){
            return $http.get('/tratamento-eventos/'+days, {
                headers: {
                    'x-access-token': AuthService.getTokenId()
                }
            });
        }

    };

    angular.module('ordenhaDigital').factory('SaudeAnimalService', Service);

})();
```

UserService.js

```
"use strict";
(function(){
```

```
	function Service($http, AuthService){

		var service = {
			callUser: callUser,
		};

		return service;


		function callUser(){
			return $http.get('/user', {
				headers: {
					'x-access-token': AuthService.getTokenId()
				}
			});
		};
	};

	angular.module('ordenhaDigital').factory('UserService', Service);

})();
```

public/

Index.html

```
<!DOCTYPE html>
<html lang="pt-BR" ng-app="ordenhaDigital">

	<head>
		<meta charset="utf-8">
		<meta name="viewport" content="width=device-width, initial-scale=1">
		<title>Ordenha Digital</title>
		<link rel="stylesheet" type="text/css"
href="vendor/bootstrap/dist/css/bootstrap.min.css">
		<link rel="stylesheet" type="text/css"
href="vendor/bootstrap/dist/css/bootstrap-theme.min.css">
		<link rel="stylesheet" type="text/css" href="css/main.css">
	</head>

	<body ui-router-styles>

		<!-- Navbar -->
		<nav class="navbar navbar-inverse" ng-controller="NavController">
			<div class="container">
				<div class="navbar-header">
					<button type="button" class="navbar-toggle
collapsed" data-toggle="collapse" data-target=".navbar-collapse" ng-show="logado">
						<span class="sr-only">Toggle
navigation</span>
```

```
                                        <span class="icon-bar"></span>
                                        <span class="icon-bar"></span>
                                        <span class="icon-bar"></span>
                                    </button>
                                    <a href="#" class="navbar-brand" ng-
click="goTo('home')">Ordenha Digital</a>
                                </div>

                                <div class="collapse navbar-collapse">
                                    <ul class="nav navbar-nav navbar-right" ng-
show="logado">
                                        <li><a href="#" ng-
click="goTo('rebanho')">Rebanho</a></li>
                                        <li><a href="#" ng-
click="goTo('ordenha')">Ordenhas</a></li>
                                        <li><a href="#" ng-
click="goTo('reproducao')">Reprodução</a></li>
                                        <li><a href="#" ng-click="goTo('saude-
animal')">Saúde Animal</a></li>
                                        <li><a href="#" title="Logout" ng-
click="doLogout()"><span class="glyphicon glyphicon-off"></span></a></li>
                                    </ul>
                                </div>
                            </div>
                        </nav>

                        <!-- MAIN CONTAINER -->
                        <div ui-view=""></div>

                        <!-- AngularJS Components -->
                        <script src="vendor/jquery/dist/jquery.min.js"></script>
                        <script src="vendor/bootstrap/dist/js/bootstrap.js"></script>
                        <script src="vendor/angular/angular.js"></script>
                        <script src="vendor/angular-ui-router/release/angular-ui-
router.js"></script>
                        <script src="vendor/angular-ui-router-styles/ui-router-styles.js"></script>
                        <script src="vendor/angular-locale-pt-br/angular-locale_pt-
br.js"></script>
                        <script src="vendor/moment/moment.js"></script>
                        <script src="vendor/amcharts/dist/amcharts/amcharts.js"></script>
                        <script src="vendor/amcharts/dist/amcharts/serial.js"></script>
                        <script src="vendor/amcharts/dist/amcharts/themes/dark.js"></script>

                        <!-- App Modules -->
                        <script type="text/javascript" src="module.js"></script>
                        <script type="text/javascript" src="routes.js"></script>

                        <!-- App Controllers -->
                        <script type="text/javascript"
src="app/login/LoginController.js"></script>
```

```
                <script type="text/javascript"
src="app/register/RegisterController.js"></script>
                <script type="text/javascript" src="app/nav/NavController.js"></script>
                <script type="text/javascript"
src="app/home/HomeController.js"></script>
                <script type="text/javascript"
src="app/rebanho/RebanhoController.js"></script>
                <script type="text/javascript"
src="app/rebanho/RebanhoCadastroController.js"></script>
                <script type="text/javascript"
src="app/rebanho/RebanhoEditController.js"></script>
                <script type="text/javascript"
src="app/animal/AnimalController.js"></script>
                <script type="text/javascript"
src="app/ordenha/OrdenhaController.js"></script>
                <script type="text/javascript"
src="app/ordenha/OrdenhaCadastroController.js"></script>
                <script type="text/javascript"
src="app/reproducao/ReproducaoController.js"></script>
                <script type="text/javascript"
src="app/reproducao/ReproducaoCadastroController.js"></script>
                <script type="text/javascript"
src="app/saudeAnimal/SaudeAnimalController.js"></script>
                <script type="text/javascript"
src="app/saudeAnimal/SaudeAnimalTratamentoController.js"></script>
                <script type="text/javascript"
src="app/saudeAnimal/SaudeAnimalCadastroController.js"></script>

                <!-- App Services -->
                <script type="text/javascript" src="services/AuthService.js"></script>
                <script type="text/javascript"
src="services/RebanhoService.js"></script>
                <script type="text/javascript"
src="services/OrdenhaService.js"></script>
                <script type="text/javascript"
src="services/ReproducaoService.js"></script>
                <script type="text/javascript"
src="services/SaudeAnimalService.js"></script>
                <script type="text/javascript" src="services/UserService.js"></script>

                <!-- App Directives -->
                <script type="text/javascript" src="directives/fileRead.js"></script>
        </body>

</html>
```

module.js

```
angular.module('ordenhaDigital', [
        "ui.router",
        "uiRouterStyles"
```

```
]);

// Bind $state and $statePrams in $rootScope to use $state.go inside ng-click
angular.module('ordenhaDigital').run(function ($rootScope, $state, $stateParams) {
        $rootScope.$state = $state;
  $rootScope.$stateParams = $stateParams;
});
```

Routes.js

```
angular.module('ordenhaDigital')
        .config([
                "$stateProvider",
                "$urlRouterProvider",
                function($stateProvider,$urlRouterProvider){

                        $urlRouterProvider.otherwise('/home');

                        // Declare states
                        var states = {

                                'login': {
                                        url: '/login',
                                        templateUrl: 'app/login/login.html',
                                        controller: 'LoginController',
                                        data: {
                                                css: 'css/login.css'
                                        }
                                },

                                'register': {
                                        url: '/register',
                                        templateUrl: 'app/register/register.html',
                                        controller: 'RegisterController',
                                        data: {
                                                css: 'css/login.css'
                                        }
                                },

                                'home': {
                                        url: '/home',
                                        templateUrl: 'app/home/home.html',
                                        controller: 'HomeController',
                                        data: {
                                                css: 'css/home.css'
                                        }
                                },

                                'rebanho': {
                                        url: '/rebanho',
                                        templateUrl: 'app/rebanho/rebanho.html',
```

```
        controller: 'RebanhoController',
        data: {
            css: 'css/rebanho.css'
        }
    },

    'rebanho-cadastro': {
        url: '/rebanho/cadastro',
        templateUrl: 'app/rebanho/cadastro.html',
        controller: 'RebanhoCadastroController',
        params: {
            'parentNumero': '',
            'parentReprodutorExterno': '',
            'dataNasc': ''
        }
    },

    'rebanho-edit': {
        url: '/rebanho/edit/:numero',
        templateUrl: 'app/rebanho/edit.html',
        controller: 'RebanhoEditController',
        data: {
            css: 'css/rebanho-edit.css'
        },
        params: {
            'numero': '',
            'tab': 'animal'
        }
    },

    'animal' : {
        url: '/animal/:numero',
        templateUrl: 'app/animal/animal.html',
        controller: 'AnimalController',
        params: {
            'numero': 'numero'
        }
    },

    'ordenha': {
        url: '/ordenha',
        templateUrl: 'app/ordenha/ordenha.html',
        controller: 'OrdenhaController',
        data: {
            css: 'css/ordenha.css'
        }
    },

    'ordenha-cadastra': {
        url: '/ordenha/cadastro',
        templateUrl: 'app/ordenha/cadastro.html',
```

```
                controller: 'OrdenhaCadastroController',
                data: {
                    css: 'css/ordenha.css'
                }
            },

            'reproducao': {
                url: '/reproducao',
                templateUrl: 'app/reproducao/reproducao.html',
                controller: 'ReproducaoController',
                data: {
                    css: 'css/reproducao.css'
                }
            },

            'reproducao-cadastro': {
                url: '/reproducao/cadastro',
                templateUrl: 'app/reproducao/cadastro.html',
                controller: 'ReproducaoCadastroController'
            },

            'saude-animal': {
                url: '/saude-animal',
                templateUrl: 'app/saudeAnimal/saudeAnimal.html',
                controller: 'SaudeAnimalController'
            },

            'saude-animal-tratamento': {
                url: '/saude-animal/tratamento/:id',
                templateUrl:
'app/saudeAnimal/saudeAnimalTratamento.html',
                controller: 'SaudeAnimalTratamentoController'
            },

            'saude-animal-cadastro': {
                url: '/saude-animal/cadastro',
                templateUrl: 'app/saudeAnimal/cadastro.html',
                controller: 'SaudeAnimalCadastroController'
            }

        };

        // Input states in stateProvider
        angular.forEach(states, function(config, name) {
    $stateProvider.state(name, config);
  });
    }
]);
```