análise e desenvolvimento de sistemas

Breno de Oliveira Souza

Pablo Henrique Paulino da Costa

Wedson de souza

**“FazenTECH”**

Sete Lagoas

2020

Breno de Oliveira Souza

Pablo Henrique Paulino da Costa

Wedson de souza

**“FazenTECH”**

Trabalho acadêmico elaborado para criação de um sistema gerencial para uma fazenda sustentável, apresentado como atividade interdisciplinar do semestre.

Orientador (a) presencial: Prof. (a) Bruno

Orientador (a) a distância: Prof.(a) Vanice Dalto

Sete Lagoas

2020

Sumário

[1. INTRODUÇÃO 4](#_Toc54714380)

[1.1. “FazenTECH” 4](#_Toc54714381)

[2. ANÁLISE ORIENTADA A OBJETOS II 5](#_Toc54714382)

[2.1. Diagrama de Use Cases: 6](#_Toc54714383)

[2.1.1. Documentação dos Use Cases: 7](#_Toc54714384)

[2.1.2. Use Case: Manter Sistema 8](#_Toc54714385)

[2.1.3. Cenário Principal: 8](#_Toc54714386)

[2.1.4. Cenário Alternativo 4: 10](#_Toc54714387)

[2.2. Diagrama de Classes: 11](#_Toc54714388)

[2.3. Diagrama de Máquina de Estados para a classe “Plantio” 14](#_Toc54714389)

[2.4. Diagrama de Atividades 15](#_Toc54714390)

[2.5. Diagrama de Sequência 16](#_Toc54714391)

[3. BANCO DE DADOS II 17](#_Toc54714392)

[3.1. Modelo entidade – relacionamento: 17](#_Toc54714393)

[3.2. Script em SQL: 17](#_Toc54714394)

[4. PROGRAMAÇÃO WEB I 20](#_Toc54714395)

[5. CONCLUSÃO: 26](#_Toc54714396)

[6. REFERÊNCIAS BIBLIOGRÁFICAS: 27](#_Toc54714397)

# INTRODUÇÃO

Foi proposto o desenvolvimento de um trabalho com base na situação geradora de aprendizagem, de se criar um sistema para gerenciar uma fazenda tecnológica sustentável, com intuito de auxiliar o controle de plantações e produção leiteira.

## “FazenTECH”

Dentro da situação geradora de aprendizagem, criamos o seguinte escopo para o sistema denominado de FazenTECH:

O sistema inicialmente terá o planejamento de plantio armazenando informações importantes para um bom gerenciamento das terras e controle da pecuária da fazenda. A navegação do sistema será feita de maneira intuitiva através de um menu, na parte superior do sistema.

O sistema deve manter sobre a fazenda informações como tamanho total da fazenda, tamanho da reserva (área para a proteção de recursos hídricos, nascentes e prevenção de erosão do solo), onde se localiza a fazenda e um campo para observações.

Para o planejamento e controle do plantio o sistema permitirá armazenar informações sobre o solo, tais como: análise completa do solo, níveis de Carbono e Nitrogênio, tamanho de cada área manejada, tipo do solo, riscos de erosões e correções de solo usados.

O sistema deve manter informações sobre cada plantio, tais como: formas de polinização da muda plantada, estado da rotação da cultura, controle de pragas usado, detalhes da adubagem, data do plantio, data do último controle de pragas usado, data da adubação, data da colheita e quantidade produzida. Para cada cultura o sistema também deve gerenciar, a fabricação de mudas, guardar informações como, nome da muda, número de mudas disponíveis, espécie, data de semeada e data de desbrota, porta enxerto (raiz), gema (Copa ou Caule, que dará frutos) e formas de irrigação.

Deve fazer o controle de máquinas, ferramentas e equipamentos que a fazenda possui. Para o controle de máquinas deve se armazenar: nome, setor usado, funcionário responsável, data de aquisição, valor gasto na compra, durabilidade, se teve ou não manutenção, data da última revisão e um campo para observações. Para o controle de ferramentas deve se armazenar: nome, funcionário que usa, descrição para que é usado, valor gasto na compra, data da compra, durabilidade e campo para observações. E para os equipamentos de proteção deve se armazenar informações como: nome, funcionário que usa, descrição para que é usado, valor gasto na compra, data da compra, durabilidade e campo para observações.

Manter informações sobre produtos e suas produções, tais como: tipo de produto, nome do produto, origem do produto (animal, vegetal), quantidade em estoque, análise da qualidade, preço, custo, data da produção, data de validade.

Manter informações sobre o manejo da criação animal. Para vacas leiteiras guardar as seguintes informações: identificação do animal, nome, data de nascimento, origem, produção total de leite, espécie(raça), data da última ordenha, temperatura do leite, estado da inseminação(se está inseminada ou não), estimativa de parto, secagem esperada, tempo de ruminação por dia, quantidade de partos com sucesso e descrição de problemas da saúde do animal. Para o registro de ordenha deve armazenar: vaca, funcionário, horário iniciado, quantidade produzida, equipamento usado, horário final e data da ordenha. E para controle dos animais doentes armazenar: identificação do animal, nome, espécie do animal, data de nascimento, detalhes do tratamento usado, dia que iniciou o tratamento e dia da finalização do tratamento.

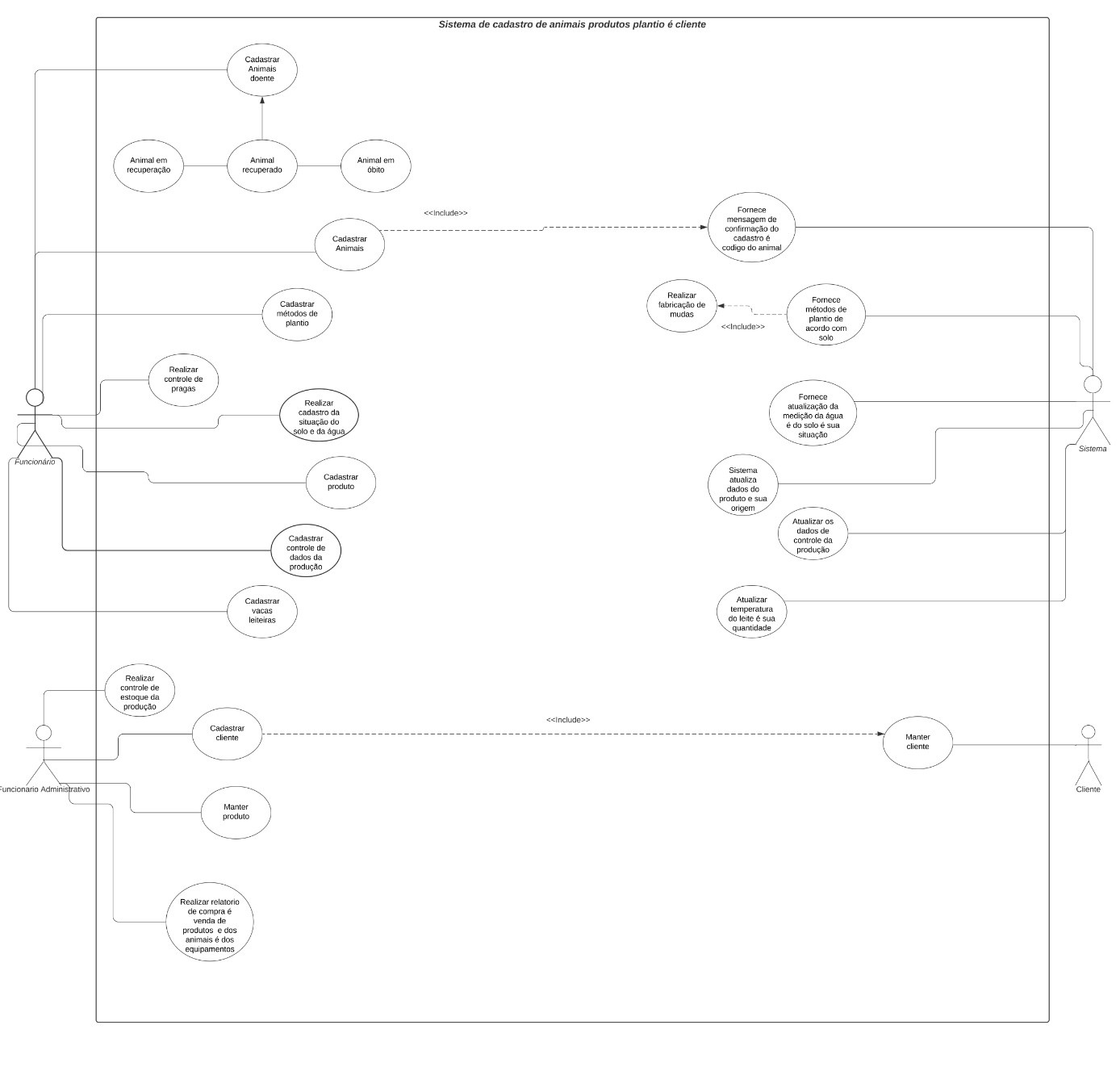
Além disso o sistema deve possibilitar ao usuário consultar os animais que estão ou ficaram doentes, apresentando informações do tratamento usado, espécie do animal, início e fim do tratamento, além do responsável pelo tratamento. O usuário também poderá consultar um relatório com informações dos plantios, trazendo os controles de pragas usados e sua data de uso. O usuário também poderá consultar um gráfico onde mostra a produção total de cada vaca leiteira e seu número de partos.

# ANÁLISE ORIENTADA A OBJETOS II

Para a documentação do sistema, dentro do que foi solicitado, segue abaixo os diagramas, que fazem parte da entrega direcionada a disciplina de análise orientada a objetos 2:

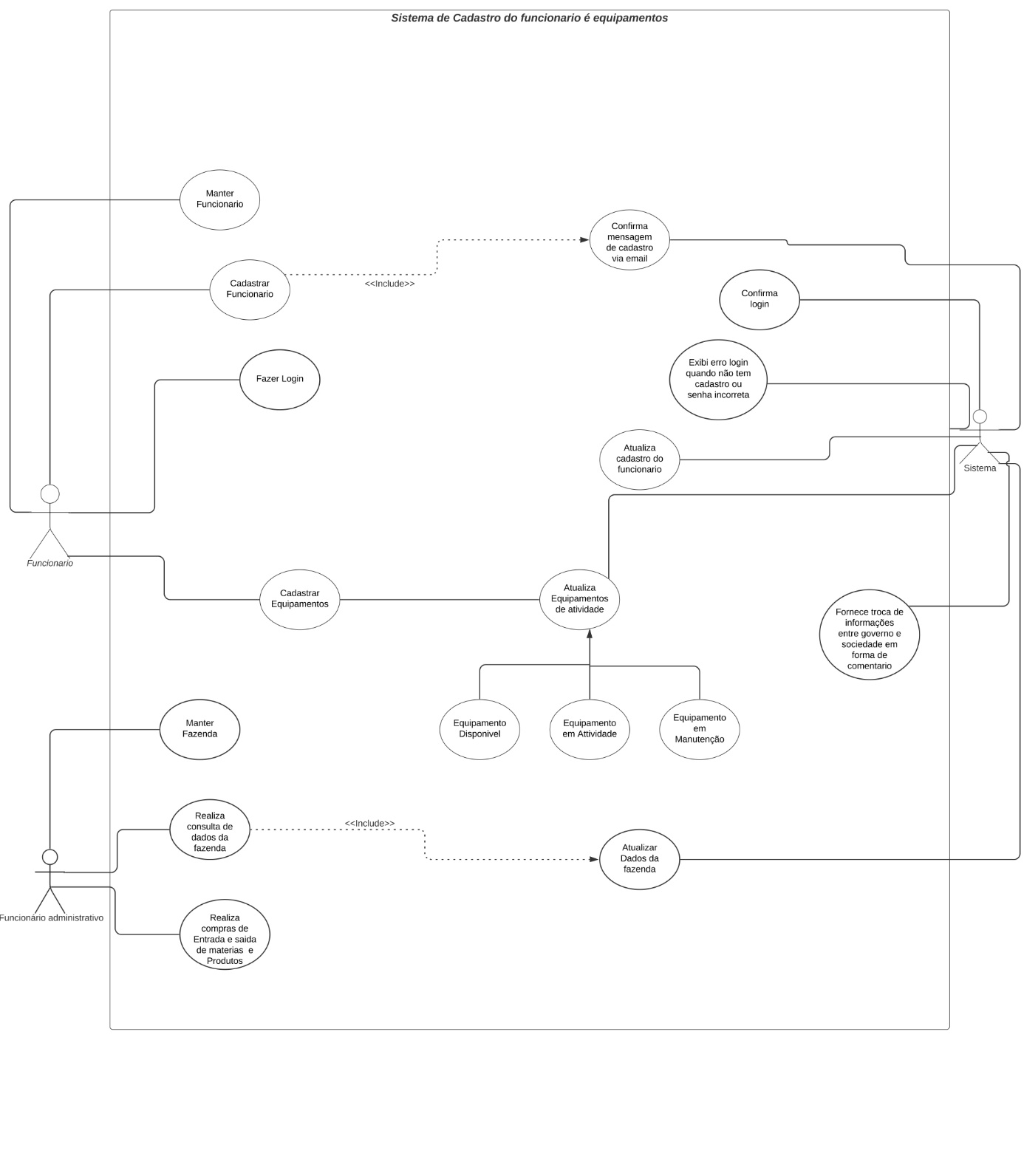
## Diagrama de Use Cases:

Como explicado por Jacques “O Diagrama de Use Cases tem o objetivo de auxiliar a comunicação entre os analistas e o cliente, descrevendo um cenário que mostra as funcionalidades do sistema do ponto de vista do usuário”, para o projeto, usamos o seguinte diagrama, figura 1:



Figura

Na figura 2, mostramos o diagrama de casos de uso resumido com os principais casos de uso.



Figura

### Documentação dos Use Cases:

A descrição do caso de uso pode ser informal, típica ou detalhada. Cada tipo de descrição empregada requer níveis diferentes de entendimento e, assim, maior será a precisão do entendimento do caso de uso, a narrativa estruturada de tal forma que todos os membros envolvidos no desenvolvimento possam entender as aplicabilidades das funções que o sistema deverá atender, pode ela ser:

• Informal: possui o nome do caso de uso e uma descrição de sua funcionalidade.

• Típica: identifica o ator que inicia o caso de uso, apresenta a descrição do fluxo normal e alternativo.

• Detalhada: identifica o ator que inicia o caso de uso, informa o objetivo, as condições de disparo (triggers), apresenta a descrição do fluxo normal e alternativo.

### Use Case: Manter Sistema

Como solicitado segue a descrição completa de um dos casos de uso, em dois cenários possíveis. Para o caso de uso Manter Sistema, para o ator da ação o Sistema.

Descrição: Sistema informa como é feito o cadastro de pessoas (funcionários), animais, produtos, equipamentos e outras funções que o sistema oferece.

### Cenário Principal:

1. Funcionário acessa o sistema.
2. Sistema válida dados para acesso.
3. Funcionário seleciona opção cadastrar.
4. Sistema fornece opões de cadastro.
5. Sistema exibe id dos funcionários, nome, usuário e senha, data de nascimento, e cpf
6. Sistema exibe id da máquina, se teve manutenção, última revisão, funcionário responsável, fornece nome da máquina, ultimo setor ao usá-la, data de aquisição, valor da compra, e durabilidade.
7. Sistema exibe id da ferramenta, funcionário responsável, fornece nome da ferramenta, ultimo setor ao usá-lo, data de aquisição, valor da compra, e durabilidade.
8. Sistema exibe id da fazenda, área total, localização, área de reserva legal e observações.
9. Sistema exibe id do equipamento de proteção, nome do equipamento, funcionário responsável, ultimo setor a usá-lo, data de aquisição, valor da compra, e durabilidade.
10. Sistema exibe id do animal doente, espécie, tipo de tratamento que deve ser usado, data de início do tratamento, data final do tratamento, funcionário responsável, nome do animal, e data do nascimento do animal.
11. Sistema exibe id da muda, espécie, quantidade de mudas, data da semeadura, data da última desbrota e forma de irrigação usada.
12. Sistema exibe id da ordenha, data da ordenha, horário de início, horário do fim da ordenha e quantidade de litros.
13. Sistema exibe id do produto, nome, origem, quantidade em estoque, analise de qualidade, custo, data de fabricação e data de validade.
14. Sistema exibe id do solo, analise do solo, níveis de carbono, níveis de nitrogênio, tipo de solo, área manejada, risco de erosões e correções de solo usadas.
15. Sistema exibe id do varejista, nome, endereço, data da última compra, detalhes da compra e quais produtos foram comprados.
16. Sistema exibe id do plantio, característica de polinização da muda, data de plantio, data da colheita, estado de rotação, data do último controle de pragas usado, data de adubação, detalhes da adubação, quantidade produzida e funcionário responsável.
17. Sistema exibe id da vaca, origem, produção total de leite, raça, data da última ordenha, temperatura do leite, se esta inseminada, se estive de parto a secagem esperada, tempo de ruminação por dia, número de partos, descrição de saúde do animal, nome da vaca e data do seu nascimento.
18. Sistema fornece opção de acessar e salvar os relatórios em um arquivo csv. O relatório que apresenta os animais que estão ou ficaram doentes, consultando informações do tratamento usado, espécie do animal, início e fim do tratamento, além do responsável pelo tratamento e o relatório que apresenta informações dos plantios, trazendo os controles de pragas usados e sua data de uso.
19. Sistema encerra o caso de uso.

### Cenário Alternativo 4:

1. Sistema valida que existe código de cadastro.
2. Sistema recupera dados associado a página.
3. Sistema permite ao funcionário: excluir, alterar, consultar dados dos funcionários cadastrados e fazer novo cadastro.
4. Sistema permite alterar, excluir, realizar consulta dos dados dos varejistas e fazer novo cadastro.
5. Sistema permite alterar, excluir, realizar consulta dos dados das vacas leiteiras e fazer novo cadastro.
6. Sistema permite alterar, excluir, realizar consulta dos dados dos solos e fazer um novo cadastro.
7. Sistema permite alterar, excluir, realizar consulta dos dados dos produtos e fazer novo cadastro.
8. Sistema permite alterar, excluir, realizar consulta dos dados das ordenhas e também fazer novo cadastro.
9. Sistema permite alterar, excluir, realizar consulta dos dados das mudas e fazer um novo cadastro.
10. Sistema permite alterar, excluir, realizar consulta dos dados das máquinas e fazer um novo cadastro.
11. Sistema permite alterar, excluir, realizar consulta dos dados das ferramentas e fazer um novo cadastro.
12. Sistema permite alterar, excluir, realizar consulta dos dados das fazendas e fazer novo cadastro.
13. Sistema permite alterar, excluir, realizar consulta dos equipamentos de proteção e realizar um novo cadastro.
14. Sistema permite alterar, excluir, realizar consulta dos dados animais doentes e fazer um novo cadastro.
15. Sistema permite alterar, excluir, realizar consulta dos dados dos plantios e fazer um novo cadastro.
16. Sistema encerra o caso de uso.

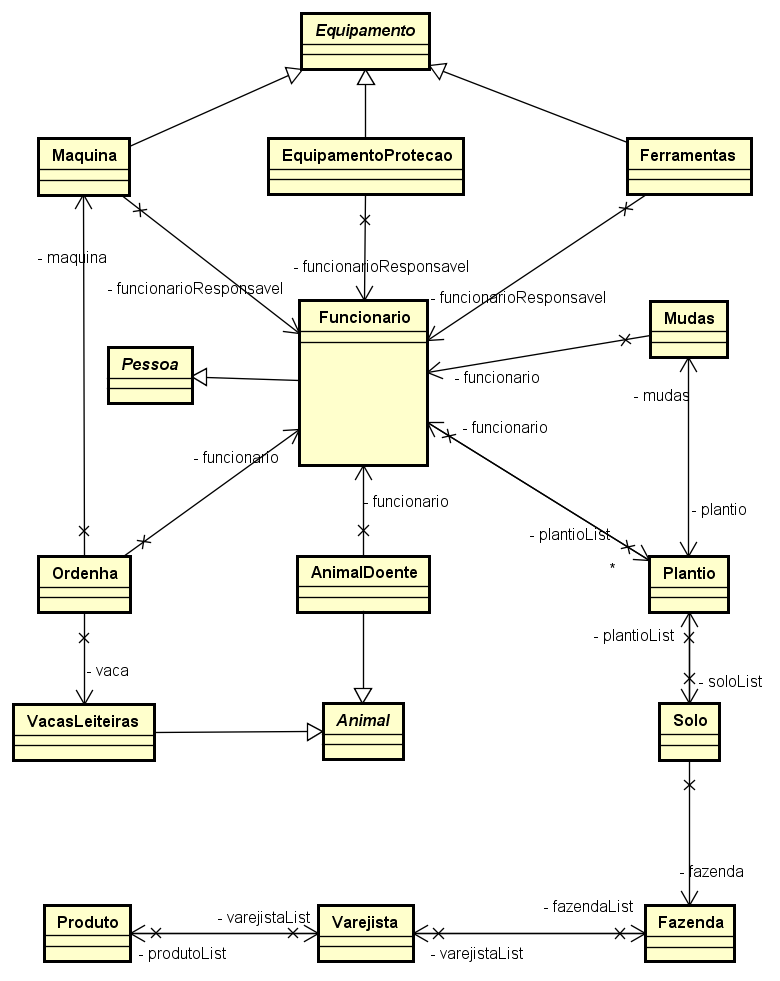
## Diagrama de Classes:

O diagrama de classe da figura 3, apresenta:



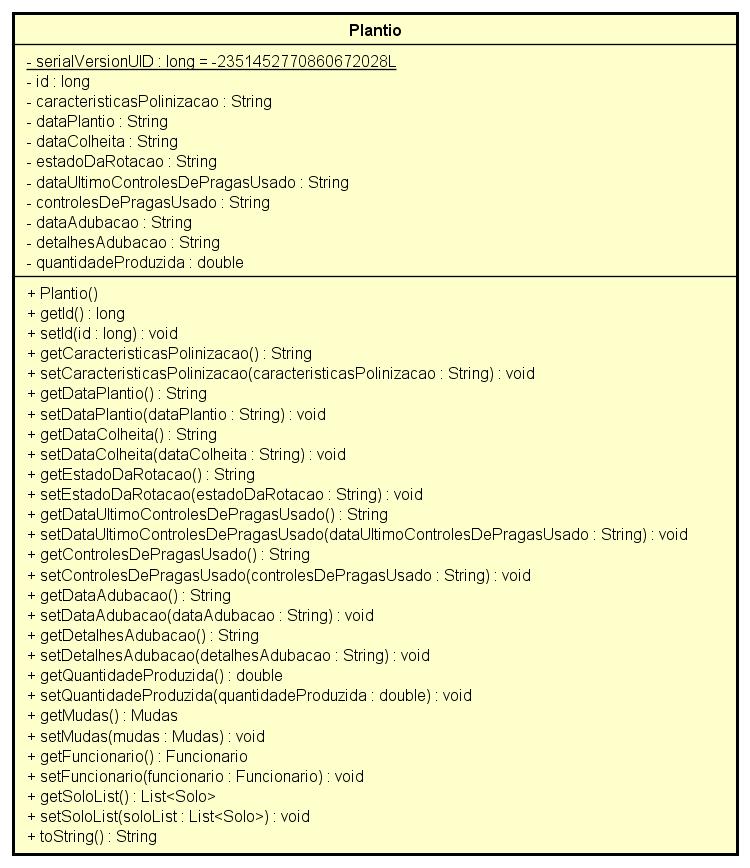
Figura

Na figura 4, tem se o Diagrama de classes resumido, sem atributos e metodos, para ficar mais legivel.



Figura

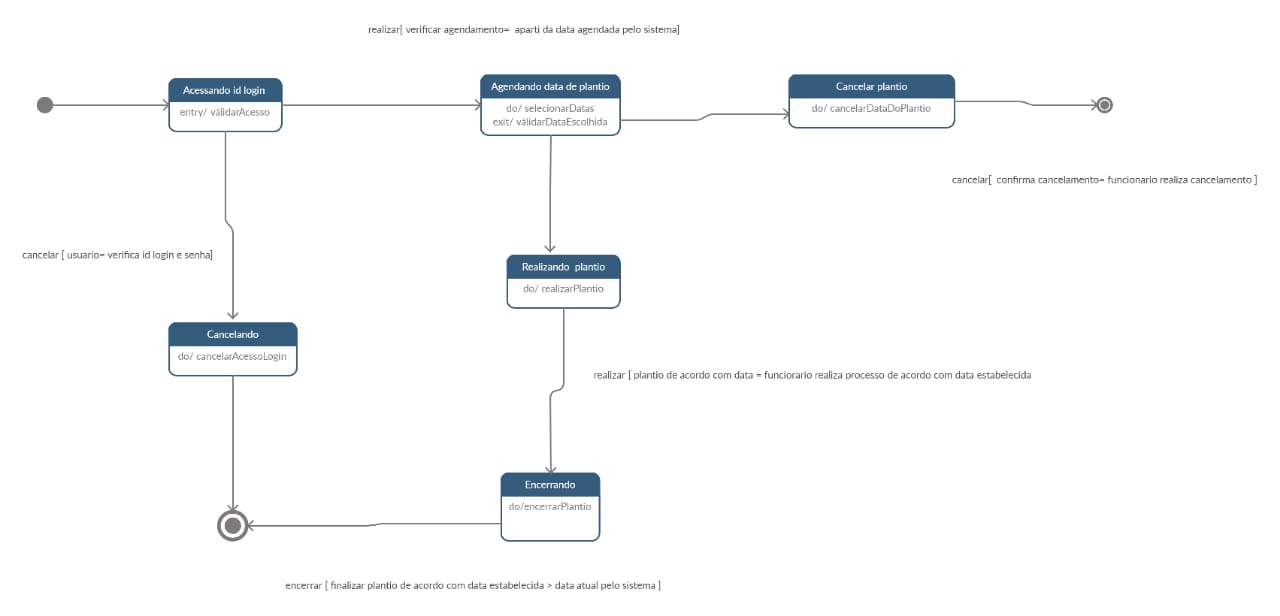
E na figura 5, tem-se em detalhe a classe plantio, com seus metodos e atributos.



Figura

## Diagrama de Máquina de Estados para a classe “Plantio”

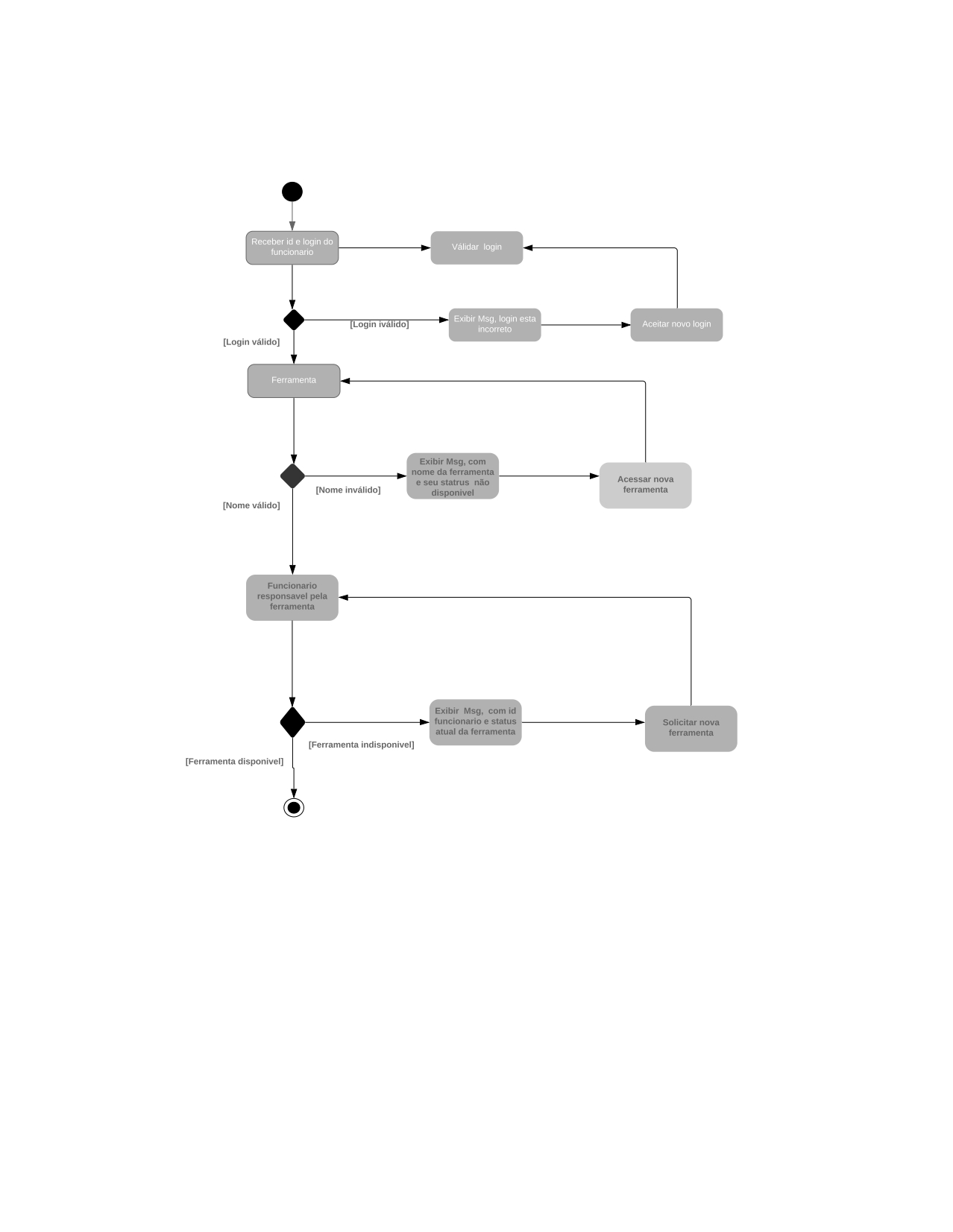
A seguir na figura 6 está o diagrama de máquina de estados detalhando os estados possíveis da classe plantio.



Figura

## Diagrama de Atividades

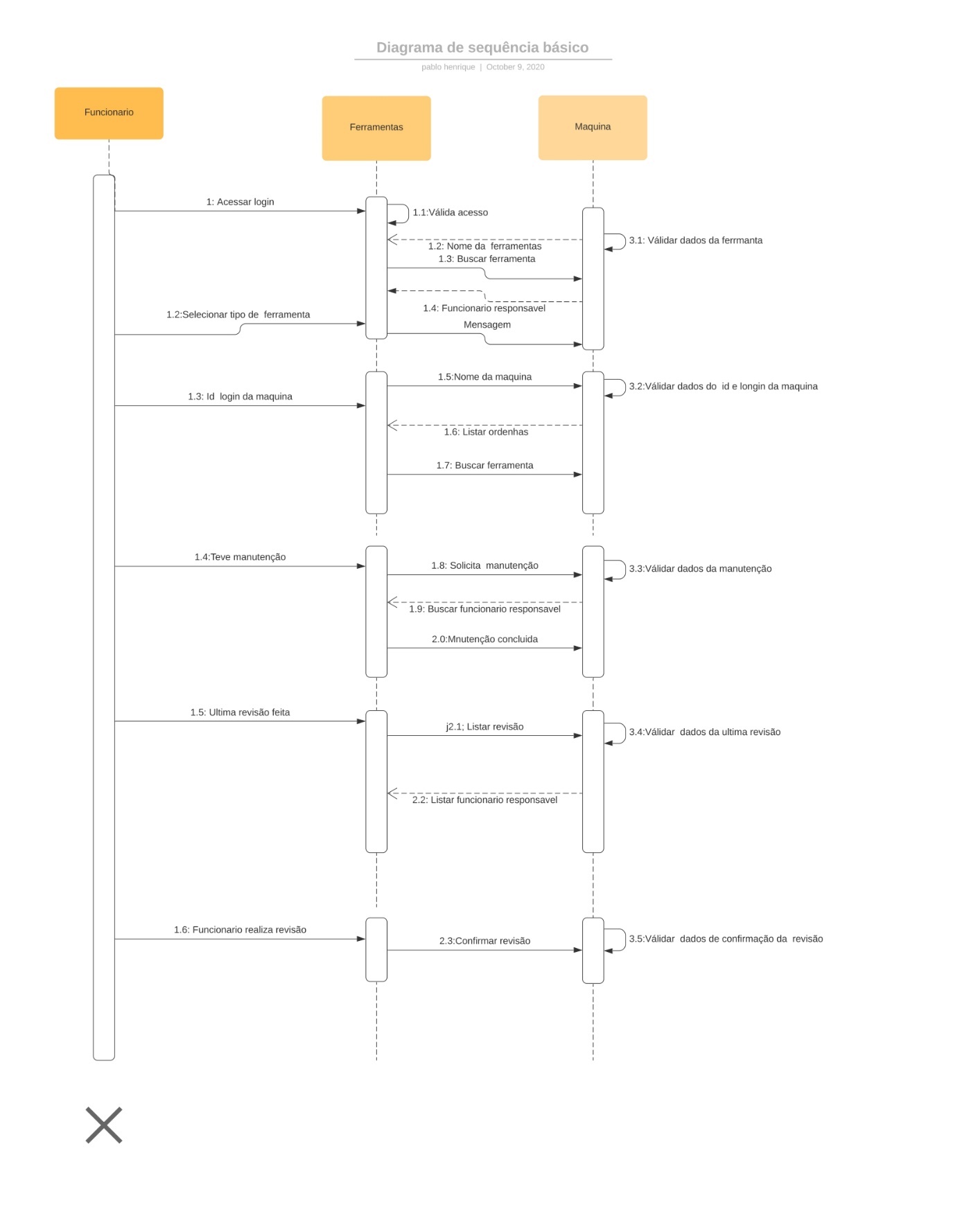
Na figura 7 está o diagrama de atividades referente ao caso de uso: do funcionário acessar o sistema para consultar informações sobre uma ferramenta, detalhando as atividades do sistema nessa ação.



Figura

## Diagrama de Sequência

A figura 8 é o Diagrama de Sequência correspondente ao caso de uso do funcionário acessar o sistema para verificar informações sobre as maquinas da fazenda.



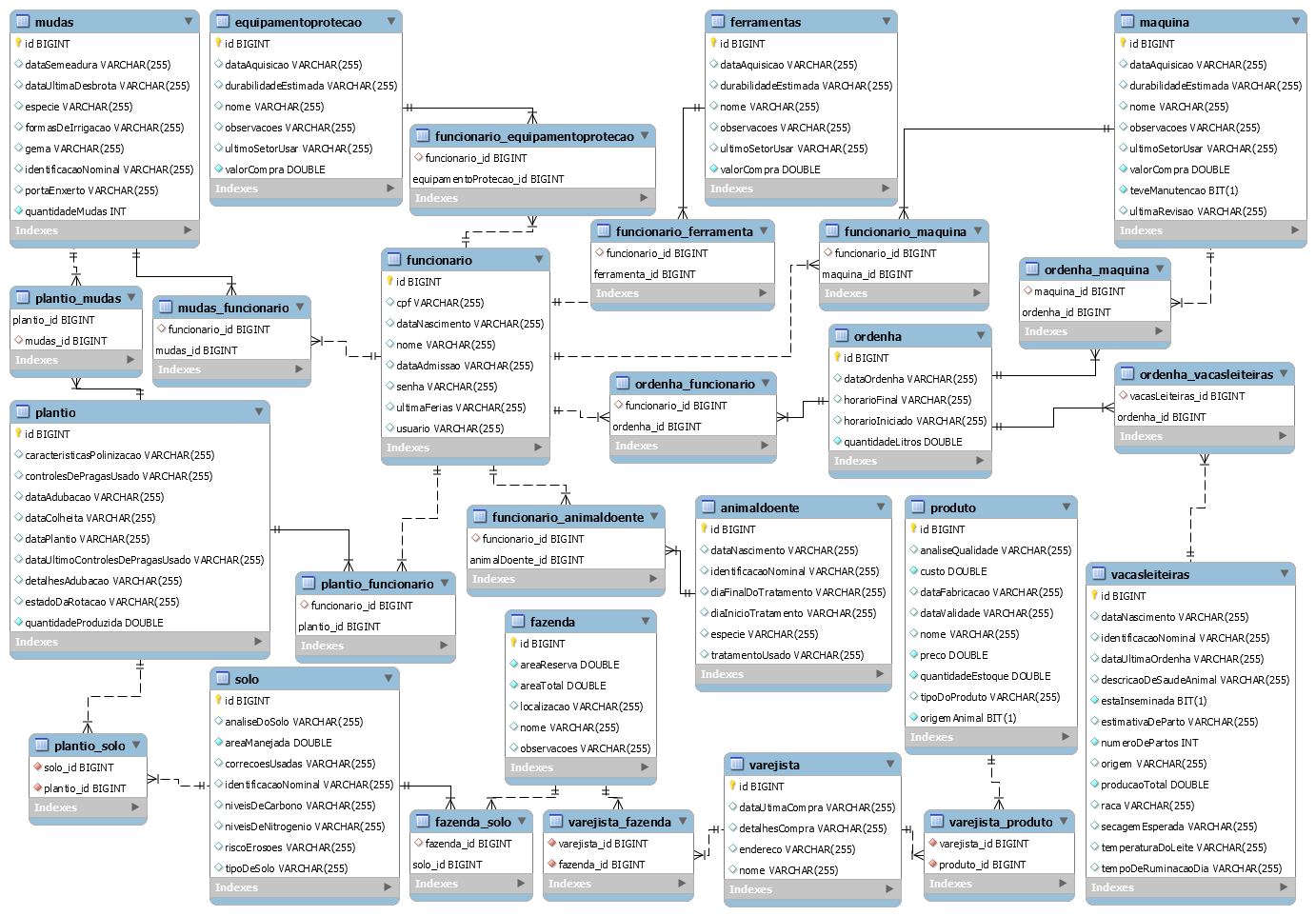
Figura

# BANCO DE DADOS II

O sistema precisa armazenar várias informações para processo de controle da fazenda, tais informações usadas para criação das classes e relacionamentos das mesmas.

## Modelo entidade – relacionamento:

Como descrito no Wikipédia “o modelo entidade relacionamento(modelo ER) é um [modelo de dados](https://pt.wikipedia.org/wiki/Modelo_de_dados) para descrever os dados ou aspectos de informação de um domínio de negócio ou seus requisitos de processo, de uma maneira abstrata que em última análise se presta a ser implementada em um [banco de dados](https://pt.wikipedia.org/wiki/Banco_de_dados), como um [banco de dados relacional](https://pt.wikipedia.org/wiki/Banco_de_dados_relacional)”. Para a situação, elaboramos o seguinte modelo:



Figura

## Script em SQL:

O Bancos de dados Relacional é um conjunto de arquivos relacionados entre si com registros sobre pessoas, lugares ou coisas. São coleções organizadas de dados que se relacionam de forma a criar algum sentido (Informação) e dar mais eficiência durante uma pesquisa ou estudo cientifico.

Dentro dessa definição, foi criado o script abaixo contendo alguns dos comandos DDL usados para criação do banco de dados e suas tabelas. Não foram adicionados os comandos de todas as tabelas para não ficar algo cansativo e extenso.

CREATE DATABASE `fazendabd`;

CREATE TABLE `fazendabd`.`fazenda` (

`id` bigint NOT NULL AUTO\_INCREMENT,

`areaReserva` double NOT NULL,

`areaTotal` double NOT NULL,

`localizacao` varchar(255) DEFAULT NULL,

`nome` varchar(255) DEFAULT NULL,

`observacoes` varchar(255) DEFAULT NULL,

PRIMARY KEY (`id`)

);

CREATE TABLE `fazendabd`.`fazenda\_solo` (

`fazenda\_id` bigint DEFAULT NULL,

`solo\_id` bigint NOT NULL,

PRIMARY KEY (`solo\_id`),

KEY `FK\_t2cya7ubad62u79oy1yrd7bm5` (`fazenda\_id`),

CONSTRAINT `FK\_abvjdrv65b35sgmheelylkpjl` FOREIGN KEY (`solo\_id`) REFERENCES `solo` (`id`),

CONSTRAINT `FK\_t2cya7ubad62u79oy1yrd7bm5` FOREIGN KEY (`fazenda\_id`) REFERENCES `fazenda` (`id`)

);

CREATE TABLE `fazendabd`.`plantio` (

`id` bigint NOT NULL AUTO\_INCREMENT,

`caracteristicasPolinizacao` varchar(255) DEFAULT NULL,

`controlesDePragasUsado` varchar(255) DEFAULT NULL,

`dataAdubacao` varchar(255) DEFAULT NULL,

`dataColheita` varchar(255) DEFAULT NULL,

`dataPlantio` varchar(255) DEFAULT NULL,

`dataUltimoControlesDePragasUsado` varchar(255) DEFAULT NULL,

`detalhesAdubacao` varchar(255) DEFAULT NULL,

`estadoDaRotacao` varchar(255) DEFAULT NULL,

`quantidadeProduzida` double NOT NULL,

PRIMARY KEY (`id`)

);

CREATE TABLE `fazendabd`.`plantio\_solo` (

`solo\_id` bigint NOT NULL,

`plantio\_id` bigint NOT NULL,

UNIQUE KEY `UK\_6jnc3m46685mpchgjc12j5foj` (`plantio\_id`),

UNIQUE KEY `UK\_huuqv136ph59oamcod21uwnli` (`solo\_id`),

CONSTRAINT `FK\_6jnc3m46685mpchgjc12j5foj` FOREIGN KEY (`plantio\_id`) REFERENCES `plantio` (`id`),

CONSTRAINT `FK\_huuqv136ph59oamcod21uwnli` FOREIGN KEY (`solo\_id`) REFERENCES `solo` (`id`)

);

CREATE TABLE `fazendabd`.`solo` (

`id` bigint NOT NULL AUTO\_INCREMENT,

`analiseDoSolo` varchar(255) DEFAULT NULL,

`areaManejada` double NOT NULL,

`correcoesUsadas` varchar(255) DEFAULT NULL,

`identificacaoNominal` varchar(255) DEFAULT NULL,

`niveisDeCarbono` varchar(255) DEFAULT NULL,

`niveisDeNitrogenio` varchar(255) DEFAULT NULL,

`riscoErosoes` varchar(255) DEFAULT NULL,

`tipoDeSolo` varchar(255) DEFAULT NULL,

PRIMARY KEY (`id`)

);

Foi criado o script abaixo contendo alguns dos comandos DML usados para inserir dados nas tabelas do banco de dados criado. Não foram adicionados os comandos de todas as tabelas para não ficar algo cansativo e extenso.

Insert Into fazendabd.fazenda (id, areaReserva, areaTotal, localizacao,

nome, observacoes)

Values (01, 13.0, 78.3, "Proximo a nascente do Rio Cipó", "Fazenda Cipó Amarelo", "Filial do Seu José"),

(02, 24.0, 189.3, "A 500m da ponte de Jequitibá", "Fazenda Jequitibá amarelo", "Matriz do Seu José");

Insert Into fazendabd.plantio (id, caracteristicasPolinizacao, controlesDePragasUsado, dataAdubacao,

dataColheita, dataPlantio, dataUltimoControlesDePragasUsado, detalhesAdubacao, estadoDaRotacao,

quantidadeProduzida)

Values (01, "Natural com abelhas", "Apenas controle organico", "14-10-2017", "Todos os anos no mês de Setembro", "08-08-2017", "05-10-2019", "Uso de esterco e adubo quimico npk", "Inicial", 20000), -- goiaba roxa

(02, "Natural com abelhas", "Apenas controle organico", "21-10-2017", "Todos os anos no mês de Setembro", "15-08-2017", "08-10-2019", "Uso de esterco e adubo quimico npk", "Inicial", 15000), -- goiaba branca

(03, "Polinização quimica", "Eliminar fontes de inóculo como maxixe", "10-04-2020", "05-06-2020", "05-03-2020", "08-05-2020", "Uso de esterco e adubo quimico npk", "A terceira rotação, necessário girar cultura", 2500);

Insert Into fazendabd.solo (id, analiseDoSolo, areaManejada, correcoesUsadas,

identificacaoNominal, niveisDeCarbono, niveisDeNitrogenio, riscoErosoes, tipoDeSolo)

Values (01, "Solo em bom estado", 5.05,"Método da solução tampão SMP ", "Terrreno ao norte da sede", "Média radicular em 30 %", "Média radicular em 20 %", "Sem riscos de erosões", "Solo em arido"),

(02, "Solo em correção", 3.04, "Método da saturação por bases", "Terreno do lago", "Média radicular em 5%", "Média radicular em 2%", "Com riscos de erosões", "Solo seco"),

(03, "Solo estavel", 6.05, "Nenhuma", "Terreno abaixo do corrego", "Média radicular em 30 %", "Média radicular em 40 %", "Sem riscos de erosões", "Solo umido");

Insert Into fazendabd.fazenda\_solo (fazenda\_id, solo\_id)

Values (02, 02), (01, 03), (02, 01);

Insert Into fazendabd.plantio\_solo (solo\_id, plantio\_id)

Values (01, 02), (02, 03);

Foi criado os scripts abaixo contendo os comandos DQL usados para consultar informações no banco de dados tais como:

Dados de todos os animais doentes, que foram tratados, buscando assim informações do tratamento usado, espécie do animal, início e fim do tratamento, além do responsável pelo tratamento.

SELECT adt.identificacaoNominal AS "Nome", adt.especie AS "Espécie", adt.diaInicioTratamento AS "Início do tratamento", adt.diaFinalDoTratamento AS "Final do tratamento", adt.tratamentoUsado AS "Tratamento usado", fun.nome AS "Responsável"

FROM fazendabd.animaldoente AS adt

INNER JOIN fazendabd.funcionario\_animaldoente AS fadt ON adt.id = fadt.animalDoente\_id

INNER JOIN fazendabd.funcionario AS fun ON fun.id = fadt.funcionario\_id;

Informações do produto como nome, custo e preço de venda do mesmo, nome dos varejistas que compraram esses produtos.

SELECT pro.nome AS "Produto", pro.custo AS "Custo unitário", pro.preco AS "Preço unitário", var.nome AS "Varejista", var.dataUtimaCompra AS "Data da última compra", var.detalhesCompra "Detalhes da compra"

FROM fazendabd.produto AS pro

INNER JOIN fazendabd.varejista\_produto AS vp ON vp.produto\_id = pro.id

INNER JOIN fazendabd.varejista AS var ON var.id = vp.varejista\_id;

Informações das ordenhas e das vacas ordenhadas, além da máquina usada e funcionário que executou a ação.

SELECT val.identificacaoNominal AS "Nome da Vaca", val.producaoTotal AS "Produção total de leite", val.secagemEsperada AS "Secagem esperada", ord.dataOrdenha AS "Data", ord.horarioIniciado AS "Início da ordenha", ord.horarioFinal AS "Término da ordenha", ord.quantidadeLitros AS "Quantidade de leite tirado", fun.nome AS "Funcionário ", maq.nome AS "Máquina Usada"

FROM fazendabd.vacaleiteira AS val

INNER JOIN fazendabd.ordenha\_vacaleiteira AS ov ON ov.vacaLeiteira\_id = val.id

INNER JOIN fazendabd.ordenha AS ord ON ord.id = ov.ordenha\_id

INNER JOIN fazendabd.ordenha\_maquina AS om ON om.ordenha\_id = ord.id

INNER JOIN fazendabd.maquina AS maq ON maq.id = om.maquina\_id

INNER JOIN fazendabd.ordenha\_funcionario AS orf ON orf.ordenha\_id = ord.id

INNER JOIN fazendabd.funcionario AS fun ON fun.id = orf.funcionario\_id;

Detalhes e datas do que foi feito e quando foi feito nas plantações

SELECT faz.nome AS "Identificação da Fazenda", mud.identificacaoNominal AS "Plantação", mud.quantidadeMudas AS "Quantidade de mudas plantadas", mud.formasDeIrrigacao AS "Formas de irrigação", mud.dataSemeadura AS "Data da semeadura", mud.dataUltimaDesbrota AS "Data da última desbrota", pla.dataPlantio AS "Data que plantou as mudas", pla.dataAdubacao AS "Data da adubação", pla.detalhesAdubacao AS "Detalhes da adubação", pla.dataColheita AS "Data da colheita", pla.quantidadeProduzida AS "Quantidade da produção total"

FROM fazendabd.fazenda AS faz

INNER JOIN fazendabd.fazenda\_solo AS fs ON fs.fazenda\_id = faz.id

INNER JOIN fazendabd.solo AS sol ON sol.id = fs.solo\_id

INNER JOIN fazendabd.plantio\_solo AS ps ON ps.solo\_id = sol.id

INNER JOIN fazendabd.plantio AS pla ON pla.id = ps.plantio\_id

INNER JOIN fazendabd.plantio\_muda AS pm ON pm.plantio\_id = pla.id

INNER JOIN fazendabd.muda AS mud ON mud.id = pm.muda\_id;

Plantações e controles de pragas e sua data de uso

SELECT mud.identificacaoNominal AS "Plantação", pla.controlesDePragasUsado AS "Controles de pragas usado", pla.dataUltimoControlesDePragasUsado AS "Data do último controle de praga"

FROM fazendabd.muda AS mud

INNER JOIN fazendabd.plantio\_muda AS pl ON pl.muda\_id = mud.id

INNER JOIN fazendabd.plantio AS pla ON pla.id = pl.plantio\_id;

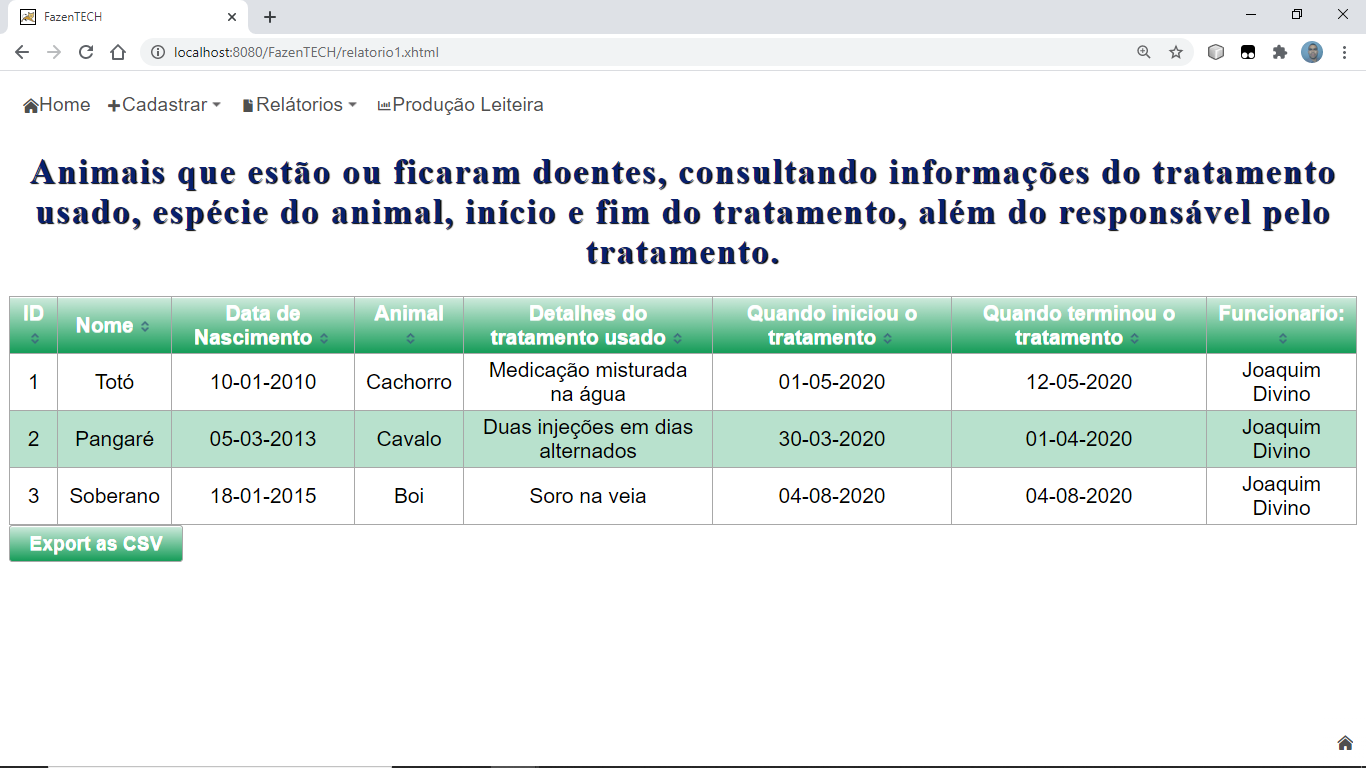
# PROGRAMAÇÃO WEB I

Além do solicitado que seria uma tela para controle de vacas leiteiras todo o trabalho foi desenvolvido com interface web, tendo então todos os cadastros e relatórios sendo acessados na interface web. Alguns dados momentaneamente, estão sendo cadastrados apenas via banco de dados, no futuro isso será implementado.

Atendendo ao solicitado o sistema permite cadastrar e alterar o cadastro de animais doentes (veja figura 10), podendo consultar também um relatório de todos os animais doentes, tratamentos usados e o funcionário responsável pelo tratamento como apresentado na figura 11.

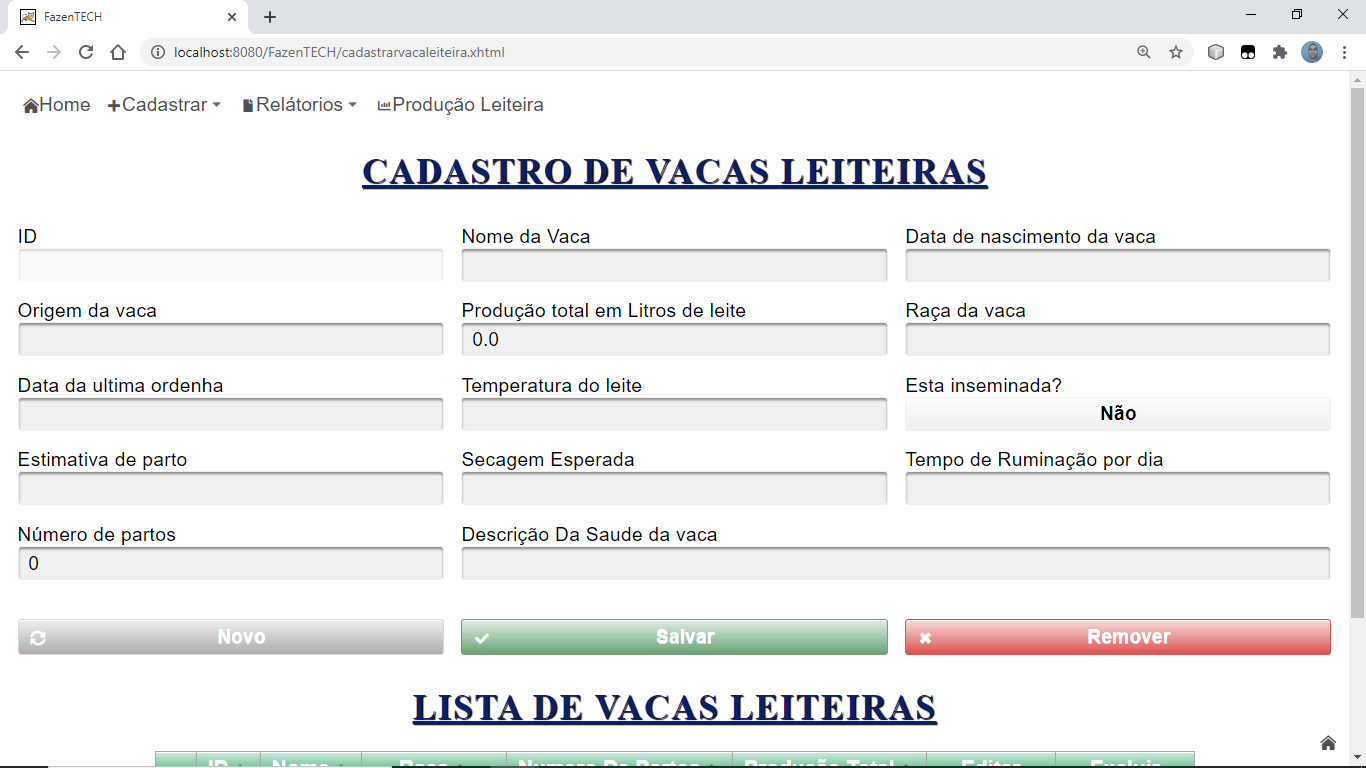


Figura



Figura

O sistema possibilita também como solicitado o controle de vacas leiteiras (veja figura 12), assim como o controle da produção leiteira, horário e data que ocorreu cada ordenha (veja figura 13).

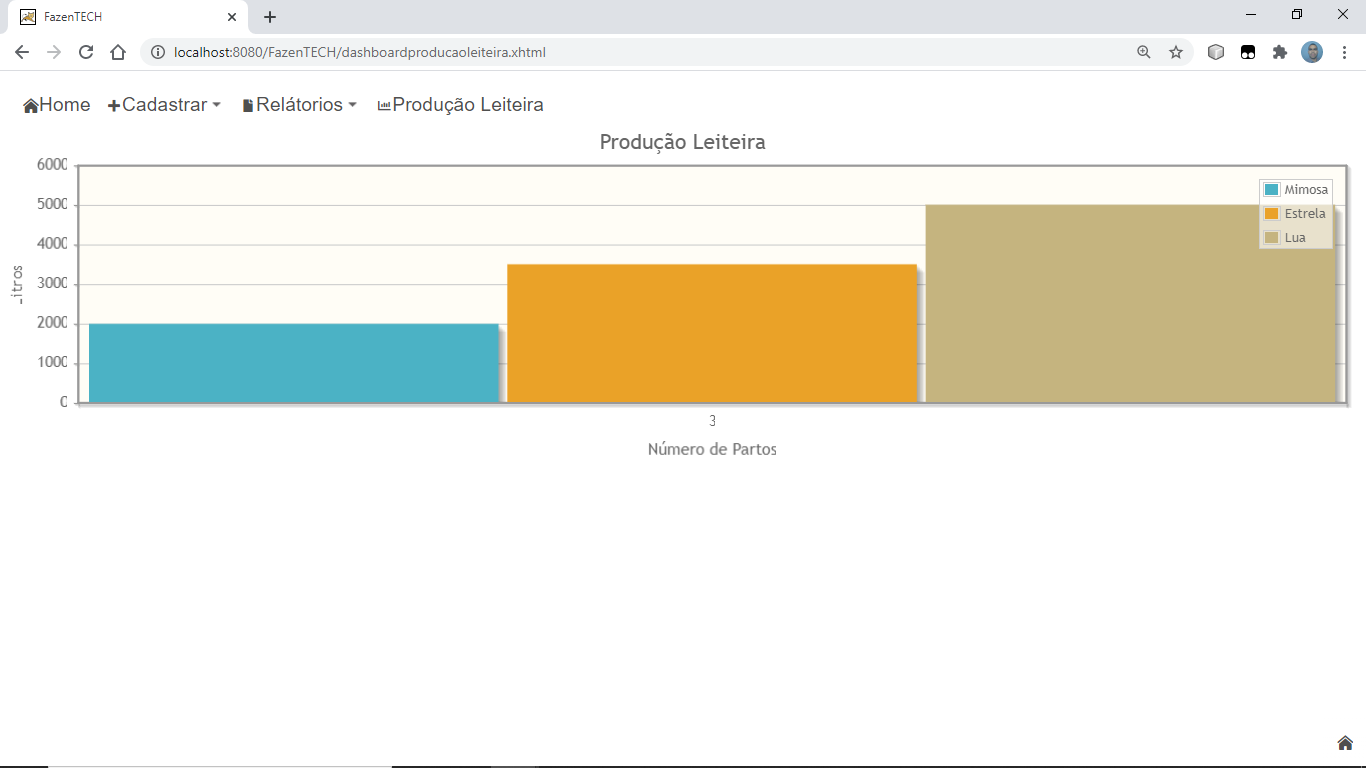


Figura



Figura

E para uma melhor apresentação de resultados, também é gerado um gráfico com a produção do total de leite de cada vaca, separadas por número de partos (bezerros), como exemplo veja a figura 14.



Figura

Para exemplificar o que foi produzido, abaixo segue a codificação de uma das páginas, que faz uso das tecnologias JSF (JavaServer Faces), PrimeFaces (xhtml), CSS, XML e HTML5.

<?xml version="1.0" encoding="ISO-8859-1" ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:h="http://java.sun.com/jsf/html"

xmlns:f="http://java.sun.com/jsf/core"

xmlns:p="http://primefaces.org/ui"

xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>

<h:outputStylesheet name="primeicons/primeicons.css" library="primefaces" />

<h:outputStylesheet library="webjars" name="font-awesome/5.12.0/css/all.min-jsf.css" />

<h:outputStylesheet library="webjars" name="font-awesome/5.12.0/css/v4-shims.min-jsf.css" />

<title>FazenTECH</title>

<style type="text/css">

.white-button.ui-button.ui-state-default {

background-color: #ADADAD;

color: #FFF;

border-color: #d6d6d6;

border-color: var(- -input-border-color, #d6d6d6);

}

.white-button.ui-button.ui-state-default:enabled:hover, .white-button.ui-button.ui-state-default:focus

{

background-color: #f2f2f2;

border-color: #ccc;

color: #373a3c;

border-color: var(- -input-hover-border-color, #ccc);

}

.green-button.ui-button.ui-state-default {

background-color: #69A376;

border-color: #69A376;

color: #fff;

}

.green-button.ui-button.ui-state-default:enabled:hover, .green-button.ui-button.ui-state-default:focus{

background-color: #4cae4c;

border-color: #69A376;

color: #000;

}

.red-button.ui-button.ui-state-default {

background-color: #d9534f;

border-color: #d9534f;

color: #fff;

}

.red-button.ui-button.ui-state-default:enabled:hover, .red-button.ui-button.ui-state-default:focus{

background-color: #d43f3a;

border-color: #d9534f;

}

.ui-state-default {

background-color: #159C58; /\*verde\*/

color: white;

}

.cor\_input {

background-color: #F0F0F0; /\* cinza claro\*/

border-color: #F0F0F0;

color: black;

}

.value {

font-weight: bold;

}

.id\_par {

background-color: rgba(21, 156, 88, 0.3) !important; /\*verde\*/

color: #000000 !important;

}

.id\_impar {

background-color: white !important;

color: #000000 !important;

}

h1 {

text-align: center;

color: #07206E; /\*azul\*/

letter-spacing: 1.5px;

text-shadow: 1px 1px 1px black;

}

.legenda\_input {

color: black;

letter-spacing: 0.2px;

}

.fixa\_rodape {

position: fixed;

bottom: 0px;

right: 0px;

}

</style>

</h:head>

<h:body>

<h:form>

<p:menubar

style="background-color: white !important; border:none!important;">

<p:menuitem value="Home" icon="pi pi-home" url="index.xhtml" />

<p:submenu label="Cadastrar" icon="pi pi-plus">

<p:menuitem value="Animal Doente" url="cadastraranimaldoente.xhtml"

icon="pi pi-plus" />

<p:menuitem value="Equipamento de prot."

url="cadastrarequipamentoprotecao.xhtml" icon="pi pi-plus" />

<p:menuitem value="Fazenda" url="cadastrarfazenda.xhtml"

icon="pi pi-plus" />

<p:menuitem value="Ferramenta" url="cadastrarferramenta.xhtml"

icon="pi pi-plus" />

<p:menuitem value="Funcionário" url="cadastrarfuncionario.xhtml"

icon="pi pi-plus" />

<p:menuitem value="Maquina" url="cadastrarmaquina.xhtml"

icon="pi pi-plus" />

<p:menuitem value="Muda" url="cadastrarmuda.xhtml"

icon="pi pi-plus" />

<p:menuitem value="Ordenha" url="cadastrarordenha.xhtml"

icon="pi pi-plus" />

<p:menuitem value="Plantio" url="cadastrarplantio.xhtml"

icon="pi pi-plus" />

<p:menuitem value="Produto" url="cadastrarproduto.xhtml"

icon="pi pi-plus" />

<p:menuitem value="Solo" url="cadastrarsolo.xhtml" icon="pi pi-plus" />

<p:menuitem value="Vaca leiteira" url="cadastrarvacaleiteira.xhtml"

icon="pi pi-plus" />

<p:menuitem value="Varejista" url="cadastrarvarejista.xhtml"

icon="pi pi-plus" />

</p:submenu>

<p:submenu label="Relátorios" icon="pi pi-file">

<p:menuitem value="Relátorios 1" url="relatorio1.xhtml"

icon="pi pi-file" />

<p:menuitem value="Relátorios 2" url="relatorio2.xhtml"

icon="pi pi-file" />

</p:submenu>

<p:menuitem value="Produção Leiteira"

url="dashboardproducaoleiteira.xhtml" icon="pi pi-chart-bar" />

</p:menubar>

</h:form>

<h:form styleClass="fixa\_rodape">

<p:menubar style="background-color: rgba(255,255,255,0.1) !important; border:none!important;">

<p:menuitem icon="pi pi-home" onclick="window.scroll(0,0)" />

</p:menubar>

</h:form>

<h1>Relatório com informações dos plantios, trazendo os controles

de pragas usados e sua data de uso.</h1>

<h:form>

<p:dataTable id="tableId" value="#{plantioBean.plantios}" var="p"

rowStyleClass="#{p.id % 2 == 0 ? 'id\_par' : 'id\_impar'}"

tableStyle="width:auto; font-size: 19px; text-align: center; margin: auto;">

<p:column headerText="ID" sortBy="#{p.id}">

<p:outputLabel value="#{p.id}" />

</p:column>

<p:column headerText="Nome" sortBy="#{p.muda.identificacaoNominal}">

<p:outputLabel value="#{p.muda.identificacaoNominal}" />

</p:column>

<p:column headerText="Data do Último Controle de Pragas Usado"

sortBy="#{p.dataUltimoControlesDePragasUsado}">

<p:outputLabel value="#{p.dataUltimoControlesDePragasUsado}" />

</p:column>

<p:column headerText="Controle de Pragas Usados"

sortBy="#{p.controlesDePragasUsado}">

<p:outputLabel value="#{p.controlesDePragasUsado}" />

</p:column>

</p:dataTable>

<p:commandButton value="Export as CSV" ajax="false">

<p:dataExporter type="csv" target="tableId"

fileName="controlesDePragasUsadoPorMuda" />

</p:commandButton>

</h:form>

</h:body>

</html>

# CONCLUSÃO:

Esse projeto teve como objetivo a elaboração e criação de um sistema gerencial para uma fazenda sustentável, apresentado como atividade interdisciplinar do semestre. Proporcionou o desenvolvimento de habilidades como: empatia ao colocar na visão de um fazendeiro, solucionadores de problemas do cotidiano de uma equipe de programação, buscando inovação e familiaridade com desenvolvimento.

O projeto desenvolvido com códigos, scripts SQL e diagramas, estão disponíveis na integra no link: <https://github.com/brenooliveira37/FazenTECH.git> . O mesmo também está disponível para melhorias futuras e correções, para que no futuro seja implantado.

# REFERÊNCIAS BIBLIOGRÁFICAS:

JACQUES. UML. Disponível em: <http://www.dsc.ufcg.edu.br/~jacques/cursos /map/html/uml/diagramas/usecases/usecases.htm>. Acesso em: 21 Out. 2020.

Wikipedia. Disponível em: <https://pt.wikipedia.org/wiki/Modelo\_entidade \_relacionamento>. Acesso em: 21 Out. 2020.