

# Desenvolvimento de um software de controle das atividades de propriedades agrícolas

Leandro Rafael Schmitt Taschetto<sup>1</sup>, Mirkos Ortiz Martins<sup>1</sup>, Gustavo Stangherlin Cantarelli<sup>1</sup>

<sup>1</sup>Curso de Sistemas de Informação – Centro Universitário Franciscano  
CEP 97010-032 – Santa Maria – RS – Brasil

taschettoster@gmail.com, mirkos@unifra.br, gus.cant@unifra.br

**Abstract.** *Due the growing necessity of managers and agricultural owners to keep costs control of performed activities and profitability control of performed activities in the property, this study aims to plan and develop a web system focused on the management of rural properties. To the development of this proposal, it was used the methodology of agile development FDD, which was developed with the programming language for web applications PHP, allied to the CakePHP framework, the Database Manager System MySQL. The planning and development of rural activities was mirrored in application actions in order to facilitate the control of rural properties, through data registration, via web interface, monitoring through reports and maintenance of activity history.*

**Resumo.** *Em razão da crescente necessidade dos gestores e proprietários agrícolas manterem um controle de atividades desempenhadas no campo, este estudo tem por objetivo planejar e desenvolver um sistema web focado na gerência de propriedades rurais. Para o desenvolvimento dessa proposta, foi utilizada a metodologia de desenvolvimento ágil FDD e a aplicação foi desenvolvida com a linguagem de programação para aplicações web PHP, aliada ao framework CakePHP, com Sistema Gerenciador de Banco de Dados (SGBD) MySQL. O planejamento e desenvolvimento de atividades rurais foi espelhado em ações da aplicação de forma a facilitar o controle de propriedades rurais, através de registro de dados, via interface web, monitoramento através de relatórios e manutenção de histórico de atividades.*

## 1. Introdução

Atualmente, a tecnologia de informação abrange, de forma geral, a maioria dos modelos de gestão. Da mesma forma, o setor do agronegócio necessita da introdução da tecnologia para facilitar o controle, reduzir custos e facilitar as tomadas de decisões.

No cenário atual, com a predominância dos sistemas web devido à facilidade de uso das aplicações web e pelo fato de ser de fácil desenvolvimento por meio do *framework*, o qual auxilia e agiliza o processo, reduzindo custos e tempo com programação, os sistemas web têm como grande vantagem a facilidade de acesso, obtido por diversos tipos de dispositivos, como desktops, notebooks, smartphones, tablets, entre outros. O uso de aplicações web proporciona ao cliente, por meio do paradigma cliente-servidor, fazer uso do sistema sem a necessidade de instalação da aplicação no dispositivo. Somente é necessário que o cliente faça uso de um navegador web, com a facilidade de criação de páginas responsivas que se adequam a diversos dispositivos e

tamanhos de tela. Por esses motivos, os sistemas web estão sendo cada vez mais difundidos em aplicações comerciais.

Portanto, este trabalho visa o planejamento e a implementação de um software web personalizado para gestores de propriedades agrícolas, visando a adequação total de suas necessidades. Desse modo, ele procura abranger o máximo de itens necessários na rotina de uma empresa rural, possibilitando facilitar a gestão, visualizar informações referentes à rotina da propriedade rural, possibilitar tomadas de decisões mais precisas e prover um meio de apresentar resultados, como diminuição de custos, além de evitar perdas e desperdícios durante todo o processo operacional, aumentando, com isso, a lucratividade na safra.

O software oferece, também, demonstrativos de atividades diárias, maquinário, manutenções realizadas no maquinário, consumo de combustível, além de relatório de gastos e despesas com tudo isso, inclusive com colaboradores.

### **1.1. Justificativa**

Verifica-se a necessidade atual de os gestores de propriedades rurais visualizarem dados referentes a custos, despesas, estoque e investimentos, como quantidade de combustível, insumos, defensivos, sementes e funcionários, utilizados nas atividades agrícolas. Além disso, tem-se em vista os problemas enfrentados na área do agronegócio, como falta de controle do estoque, custos, controle do planejamento diário e dificuldade com manutenções, que podem ser resolvidos por meio da implementação de um software projetado para a propriedade.

### **1.2. Objetivo geral**

Planejar e desenvolver um software web para auxiliar o gestor da propriedade rural no controle e gerenciamento das atividades realizadas na propriedade, bem como monitorar o custo operacional.

### **1.3. Objetivos específicos**

Os objetivos específicos do trabalho estão listados a seguir:

- Realizar levantamento das prioridades, necessidades, obrigações, despesas e custos da atividade;
- Utilizar linguagem PHP para desenvolver a aplicação web;
- Utilizar *framework* CakePHP como auxílio no desenvolvimento da aplicação;
- Fazer uso do Sistema Gerenciador de Banco de Dados (SGBD) MySQL;
- Gerenciar despesas com funcionários;
- Gerenciar maquinário, custos com combustível, manutenção;
- Fazer o controle da produção e rentabilidade (sacas por hectare);
- Gerenciar custos com insumos, defensivos, sementes, aplicações aéreas de defensivos, entre outros;
- Gerar relatórios de maquinário, propriedade, manutenção, custos, combustível, consumo de combustível por máquina, safra.

## **1.4 Estrutura do trabalho**

Na Seção 2, é apresentado o referencial teórico, com as tecnologias que serão utilizadas para o desenvolvimento do software, também aborda os trabalhos correlatos e como se relacionam com a proposta do presente projeto.

Na Seção 3, é apresentada a metodologia utilizada no desenvolvimento deste trabalho, já a Seção 4 mostra o desenvolvimento de atividades que compõe o trabalho e a Seção 5 expõe os resultados do desenvolvimento da aplicação.

A Seção 6, apresenta a conclusão sobre o trabalho e, por fim, a Seção 7 expõe as referências utilizadas como base para o desenvolvimento do trabalho.

## **2. Referencial Teórico**

Esta seção consiste na realização da revisão bibliográfica para desenvolver o trabalho. Assim, apresentam-se trabalhos relacionados, tecnologias estudadas e propostas para a realização do sistema.

O projeto do sistema web foi desenvolvido utilizando várias tecnologias, contribuindo tanto para a usabilidade como para o seu desenvolvimento. Dessa forma, serão apresentadas as tecnologias que farão parte da implementação do sistema.

### **2.1. Agronegócio, agricultura e insumos agrícolas**

Define-se o agronegócio como produção rural em maior escala, desde o plantio à criação de rebanhos, em terras próprias ou arrendadas.

“O agronegócio é um conjunto de empresas que produzem insumos agrícolas, as propriedades rurais, as empresas de processamento e toda a distribuição. No Brasil o termo é usado quando se refere a um tipo especial de produção agrícola, caracterizada pela agricultura em grande escala, baseada no plantio ou na criação de rebanhos e em grandes extensões de terra. Estes negócios, via de regra, se fundamentam na propriedade latifundiária bem como na prática de arrendamentos” [Callado 2006].

Agricultura é um termo utilizado para o cultivo de plantas e realizado pelo produtor rural, o qual, utilizando técnicas de produção, consegue obter diversos produtos agrícolas para consumo próprio e para comércio, gerando lucro para si. Para Diehl (1984), “a agricultura é a arte de obter do solo, mantendo sempre a sua fertilidade, o máximo lucro”.

Insumo agrícola é todo fator de produção utilizado com o objetivo de garantir a nutrição e a proteção das plantas, com o intuito de obter a produtividade necessária na lavoura, com produtos finais de boa qualidade. Também podem ser considerados insumos os maquinários, os equipamentos, as manutenções e o combustível consumido pelas máquinas [Neves 1998].

Os fertilizantes, como o adubo, são usados para suprir as deficiências de nutrientes no solo. Os defensivos podem ser divididos em agrotóxicos, que são produtos químicos utilizados para o controle de pragas, insetos e doenças que podem ocorrer nas plantações, e herbicidas similares aos agrotóxicos, que têm a finalidade específica de controle de ervas daninhas; seu principal benefício é a rapidez de ação [Neves 1998].

## 2.2. Gestão de custos

A gestão de custos utiliza ferramentas da contabilidade, como a Unidade de Esforço de Produção (UEP), para visualizar de forma concreta dados referentes a custos e, dessa forma, utilizá-los nas tomadas de decisão.

“O método da UEP baseia-se na unificação da produção para simplificar o processo de controle de gestão. A mensuração do desempenho da empresa é feita por meio de custos e medidas físicas de eficiência, eficácia e produtividade”. [Bornia 2002].

Portanto, contabilidade de custos é o ramo da contabilidade que tem por objetivo produzir informações para diversos níveis gerenciais de uma organização, auxiliando no controle e no planejamento das operações e das tomadas de decisões [Wernke 2004].

O custo de produção é a soma dos valores de todos os recursos (insumos e serviços) utilizados no processo produtivo de uma atividade agrícola, em certo período de tempo e que podem ser classificados em curto e longo prazos. Também podemos evidenciar que a estimativa dos custos está ligada à gestão da tecnologia, ou seja, à alocação eficiente dos recursos produtivos e o conhecimento dos preços destes recursos. Economicamente, a questão relativa ao curto ou longo prazo refere-se à possibilidade de variação dos fatores de produção. Geralmente é considerado curto prazo se pelo menos um dos fatores de produção não puder variar no período considerado, quando no longo prazo, todos os fatores podem variar [CASTRO et al, 2009].

A gestão de custos rurais, por envolver diversos fatores, como insumos, custos de produção rentabilidade, gestão de maquinários, funcionários, difere da gestão de outros tipos de custos. Também a gestão de custos de atividades agrícolas varia bastante conforme o domínio de negócio, como o tipo de solo, cultivar, tamanho da propriedade.

O cálculo dos custos de produção pode ser realizado calculando o somatório de todos os custos e despesas, necessários para produção na safra. A rentabilidade deve ser calculada, com base em dados estatísticos e comparações entre produção de uma determinada cultivar.

## 2.3. CakePHP

O CakePHP é um *framework* web desenvolvido para dar mais agilidade e flexibilidade ao desenvolvimento de sistemas web. O CakePHP fornece uma base robusta e sólida às aplicações, tratando aspectos desde a requisição inicial do usuário até a renderização de uma página web, visto que segue o princípio *Model-View-Controller* (MVC), que permite customizar e estender facilmente muitos aspectos de sua aplicação [CakePHP 2017].

O objetivo do CakePHP é simplificar o processo de desenvolvimento na construção de aplicações web, dado que possui um núcleo que organiza o banco de dados e demais recursos que ajudam a reduzir a codificação. Isso facilita o programador na hora de gerar formulários e páginas, criando, automaticamente, formulários, seguindo os atributos da tabela do banco de dados. Além disso, possui outros recursos importantes, como validação embutida, listas de controle de acesso (ACLs) que fornecem uma forma eficiente de controlar o tráfego dentro e fora da sua rede, segurança e componentes de manipulação de sessão e *cache* de *view*. Dessa maneira, o desenvolvedor não precisa

desenvolver o código em PHP desde o início, reduzindo, drasticamente, custos e tempo com programação.

### **2.3.1. Bootstrap**

Bootstrap é o mais popular *framework* HTML, CSS e JS para desenvolvimento de projeto responsivo e focado para dispositivos móveis na web. O Bootstrap torna o desenvolvimento web mais rápido e fácil e é feito para pessoas de todos os níveis e para dispositivos de qualquer forma ou tamanho [Bootstrap 2016]. O Bootstrap também possui uma gama de componentes *plugins* que auxiliam o designer na implementação.

### **2.3.2. PHP**

O PHP é uma linguagem interpretada que é executada no servidor web antes de a página ser enviada ao cliente. Portanto, ele fica impossibilitado de ter acesso ao script, recebendo apenas uma página HTML resultante da execução do PHP no servidor, possibilitando maior privacidade e segurança [Converse e Park 2002].

O PHP é uma linguagem de programação dinâmica, isto é, possui alta integração com outras linguagens, como HTML e JavaScript. Ele também possui funções, como integração com o banco de dados.

## **2.4. SGBD MySQL**

O MySQL é um Sistema Gerenciador de Banco de Dados (SGBD), um software livre com base na licença *General Public License* (GPL), mas que também possui licença comercial caso o software não seja GPL. É um dos mais utilizados em aplicações web e possui fácil integração com PHP [Oracle 2017].

## **2.5. Trabalhos correlatos**

Nesta seção, são apresentados os trabalhos relacionados, que serviram de base para o desenvolvimento do trabalho proposto.

### **2.5.1. Gerenciamento de estoque e custos para agricultura através de um sistema informatizado**

O trabalho desenvolvido por Alcântara (2012) traz em pauta as necessidades de informatização no gerenciamento de propriedades rurais, propondo a implementação de um software desenvolvido em linguagem Java, que utiliza SGBD MySQL para controle de estoque, consumo de combustível, insumos e demais custos gerados na propriedade. Tal trabalho pretende auxiliar o agricultor e o agrônomo no gerenciamento de propriedades; proporcionar a realização de cálculos (antes feitos manualmente) por meio de um sistema informatizado; controlar o estoque de defensivos disponíveis para aplicação; gerenciar o consumo de combustível; e facilitar o controle da quantidade de defensivos necessários para aplicação em determinada área.

### **2.5.2. Agrojava: sistema de gestão para pequenas propriedades rurais**

O trabalho desenvolvido por Weber (2013) trata de um software para gestão rural, desenvolvido em linguagem Java, com banco de dados PostgreSQL. Tem por finalidade facilitar a administração de pequenas propriedades rurais, possibilitando uma visão ampla sobre custos de produção que, ocasionalmente, passam despercebidos. Por meio da

geração de relatórios com informações precisas, o software ajuda nas tomadas de decisão, reduzindo, assim, gastos desnecessários e, logo, aumentando a lucratividade.

### **2.5.3. Desenvolvimento de um software para gestão de pecuária**

O trabalho de Boeira e Cantarelli (2016) objetiva desenvolver um software para auxiliar na gestão da pecuária, utilizado como ferramenta para monitorar as atividades na criação de bovinos. Atualmente, os clientes estão cada vez mais exigentes em receber carnes de qualidade e, por esse motivo, os pecuaristas têm estado preocupados em melhorar a criação e a qualidade do gado e, principalmente, tornar possível o aumento da lucratividade.

Para tanto, foram desenvolvidas duas aplicações: a principal, web utilizando PHP com *framework* Bootstrap e SGDB MySQL, e a secundária, desenvolvida em Java para dispositivos móveis Android, servindo para incrementar a aplicação principal e funcionando tanto on-line como off-line. Também foi utilizado o sensor RFID nos brincos dos animais e no curral, o que permite sua identificação.

Sobre o trabalho, é importante destacar a importância da aplicação *mobile*, por facilitar a coleta das informações nas atividades cotidianas de criação de bovinos, como, por exemplo: aplicação de medicamentos, pesagens, apertes, entre outras tarefas realizadas no manejo dos animais.

### **2.5.4. Considerações sobre trabalhos correlatos**

A partir dos trabalhos correlatos pesquisados, nota-se a importância da informatização na gestão de agronegócio, visto que ela agrega facilidade na gestão financeira, melhora a organização das tarefas de manejo agrícola e reduz os custos de produção, aumentando, assim, a lucratividade.

No caso dos trabalhos de Alcântara (2012) e Weber (2013), foram propostos softwares de gestão agrícola desenvolvidos em linguagem Java para Computador Pessoal (PC), não permitindo acesso pela web ou por dispositivos móveis. Boeira e Cantarelli (2016) propuseram um software web e um auxiliar, desenvolvido para dispositivos móveis, para gestão de pecuária e rebanhos.

No caso do presente trabalho, ele possui benefícios em relação aos trabalhos de Alcântara (2012) e Weber (2013), como o desenvolvimento web, que facilita a utilização em múltiplos dispositivos como tablets, smartphones, notebooks, facilitando a utilização do software em diversos lugares da propriedade. Também em relação ao trabalho de Alcântara (2012), este projeto propõe gerar mais relatórios de fundamental importância para os gestores, visto que auxiliam no controle e redução dos custos e despesas, e ajudam na tomada de decisões. O presente projeto propõe somente a abordagem na produção agrícola, diferentemente do trabalho de Boeira e Cantarelli (2016) que é focado na área da pecuária.

## **3. Metodologia**

No que se refere à metodologia de desenvolvimento do trabalho, observa-se que, atualmente, o uso de metodologias ágeis para desenvolvimento de *software* predomina, já que ela pode reduzir o tempo de desenvolvimento do *software*, se comparada às metodologias clássicas [Barbosa *et al.* 2006].

A metodologia utilizada neste trabalho contempla a metodologia de desenvolvimento ágil *Feature Driven Development* (FDD), ou seja, Desenvolvimento Guiado por Funcionalidades, que consiste na execução de cinco processos: desenvolver modelo inicial, criar lista de funcionalidades, planejar por funcionalidade, arquitetar por funcionalidade e construir por funcionalidade. Cada processo possui uma lista de tarefas a serem executadas, que possuem seus requisitos para execução [Palmer e Felsing 2002]. A Figura 1 mostra as duas fases da metodologia, a de Concepção e Planejamento e a fase de Construção.

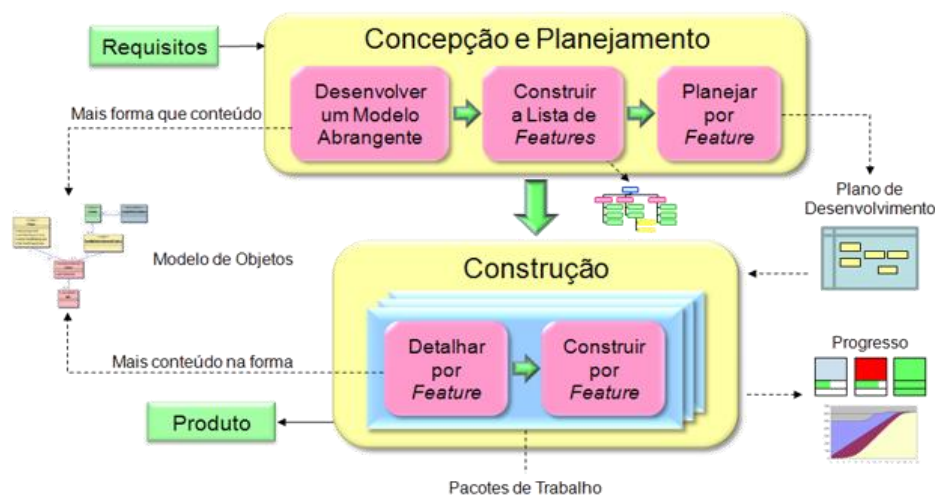


Figura 1. Diagrama de fases do FDD [Moretti Júnior 2016]

### 3.1. Construção da lista de funcionalidades

Nesta etapa, será elaborada uma lista com os Requisitos Funcionais (RF) e Não Funcionais (RNF), que levam em conta as funcionalidades propostas no sistema, com o intuito de demonstrar como o software será desenvolvido e o grau de complexidade e relevância de cada requisito do software.

#### 3.1.1. Requisitos Funcionais

- RF01. Gerenciar Usuário: O sistema deverá possuir um módulo para realizar cadastros, alteração, consulta e exclusão de usuários.
  - ✓ Complexidade: Baixa.
  - ✓ Relevância: Essencial.
- RF02. Gerenciar Propriedade: O sistema deverá possuir um módulo para realizar cadastros, alteração, consulta e exclusão de propriedades.
  - RF02.1. Gerenciar Investimentos: O sistema deverá possuir um módulo para realizar cadastros, alteração, consulta e exclusão de investimentos na propriedade.
    - ✓ Complexidade: Baixa.
    - ✓ Relevância: Essencial.
- RF03. Gerenciar Maquinário: O sistema deverá possuir um módulo para realizar cadastros, alteração, consulta e exclusão de máquinas.
  - ✓ Complexidade: Baixa.
  - ✓ Relevância: Essencial.

- RF04. Gerenciar Manutenção: O sistema deverá possuir um módulo para realizar cadastros, alteração, consulta e exclusão de manutenções do maquinário.
  - RF04.1 Gerenciar Manutenção Preventiva: O sistema deverá possuir um módulo para realizar cadastros, alteração, consulta e exclusão de manutenções, bem como calcular seus custos e quando deve ser feita a próxima manutenção na máquina.
  - RF04.2 Gerenciar Manutenção Corretiva: O sistema deverá possuir um módulo para realizar cadastros, alteração, consulta e exclusão de manutenções corretivas. Também deverá calcular o custo com essa manutenção na máquina.
  - RF04.3 Gerenciar Manutenção Periódica: O sistema deverá possuir um módulo para realizar cadastros, alteração, consulta e exclusão de manutenções, bem como calcular seus custos e quando deve ser feita a próxima manutenção na máquina, também deve atender todas as normas da manutenção periódica.
  - ✓ Complexidade: Alta.
  - ✓ Relevância: Essencial.
- RF05. Gerenciar Combustível: O sistema deverá possuir um módulo para realizar cadastros, alteração, consulta e exclusão de combustíveis no sistema.
  - RF05.1 Calcular consumo de combustível por atividade: O sistema deverá possuir um módulo capaz de calcular o consumo de combustível em uma determinada atividade.
  - RF05.2 Calcular consumo de combustível por máquina: O sistema deverá possuir um módulo capaz de calcular o consumo de combustível por uma determinada máquina.
  - ✓ Complexidade: Média.
  - ✓ Relevância: Essencial.
- RF06. Gerenciar Custos: O sistema deverá possuir um módulo para realizar cadastros, alteração, consulta e exclusão de custos.
  - ✓ Complexidade: Média.
  - ✓ Relevância: Essencial.
- RF07. Gerenciar Safra: O sistema deverá possuir um módulo para realizar cadastros, alteração, consulta e exclusão de safras no sistema.
  - RF07.1. Gerenciar Controle de Atividade diárias: O sistema deverá possuir um módulo para realizar cadastros, alteração, consulta e exclusão de atividades diárias no sistema.
  - RF07.2. Gerenciar Investimentos Safra: O sistema deverá possuir um módulo para realizar cadastros, alteração, consulta e exclusão de investimentos realizados na safra.
    - RF07.2.1 Gerenciar Insumos: O sistema deverá possuir um módulo para realizar cadastros, alteração, consulta e exclusão de investimentos em insumos.
    - RF07.2.2 Gerenciar Sementes: O sistema deverá possuir um módulo para realizar cadastros, alteração, consulta e exclusão de investimentos em sementes.



- RF07.2.3 Gerenciar Defensivos: O sistema deverá possuir um módulo para realizar cadastros, alteração, consulta e exclusão de investimentos em defensivos.
  - RF7.4. Gerenciar Rentabilidade: O sistema deverá possuir um módulo capaz de efetuar cálculos da rentabilidade de produção da safra.
    - ✓ Complexidade: Média.
    - ✓ Relevância: Essencial.
- RF08. Gerenciar Relatórios: O sistema deverá possuir um módulo para geração de relatórios de custos, atividades diárias, investimentos na propriedade, rentabilidade da safra, maquinário, investimentos da safra e consumo de combustível, bem como a realização de cálculos para geração desses relatórios.
  - ✓ Complexidade: Média.
  - ✓ Relevância: Essencial.

### 3.1.2. Requisitos Não Funcionais

- RNF01. SGBD MySQL: Utilizar o MySQL para gerenciamento do banco de dados do sistema.
  - ✓ Complexidade: Média.
  - ✓ Relevância: Essencial.
- RNF02. Linguagem de programação PHP: Utilizar a IDE (Ambiente de Desenvolvimento Integrado) NetBeans versão 8.2 e linguagem de programação PHP e HTML para desenvolvimento da aplicação web.
  - ✓ Complexidade: Alta.
  - ✓ Relevância: Essencial.
- RNF03. *Framework* CakePHP: Utilizar o *Framework* CakePHP como auxílio no desenvolvimento do sistema e na comunicação com banco de dados.
  - ✓ Complexidade: Média.
  - ✓ Relevância: Desejável.

### 3.2. Projeto

O desenvolvimento do projeto segue a metodologia FDD, visando uma melhor compreensão do processo de gerenciamento de propriedades e atividades agrícolas, permitindo o embasamento necessário para desenvolvimento da aplicação web. Nos Apêndices A, B e C apresenta-se a modelagem inicial do sistema, como o diagrama de domínio do sistema, diagrama de caso de uso do sistema e descritivos de casos de uso que apresentam as funcionalidades do sistema.

### 3.3. Construir por funcionalidade

Nesta seção são apresentados os Diagramas de Classe (Figura 2), o Diagrama Entidade-Relacionamento (DER) do banco de dados do sistema (Figura 3) e o Modelo de Desenvolvimento.

O Diagrama de Classes demonstra a interação das classes do sistema, que contém seus atributos, características e métodos de cada classe, demonstrando as funcionalidades projetadas no sistema.

O software foi desenvolvido para plataforma web, utilizando a linguagem PHP, com auxílio do *framework* CakePHP, utilizando a arquitetura MVC (*Model-View-Controller* ou Modelo-Visão-Controller) que separa a aplicação em três partes: A *Model* que representa os dados, a *View* que permite a visualização dos dados e o *Controller* manipula e roteia as requisições dos usuários.

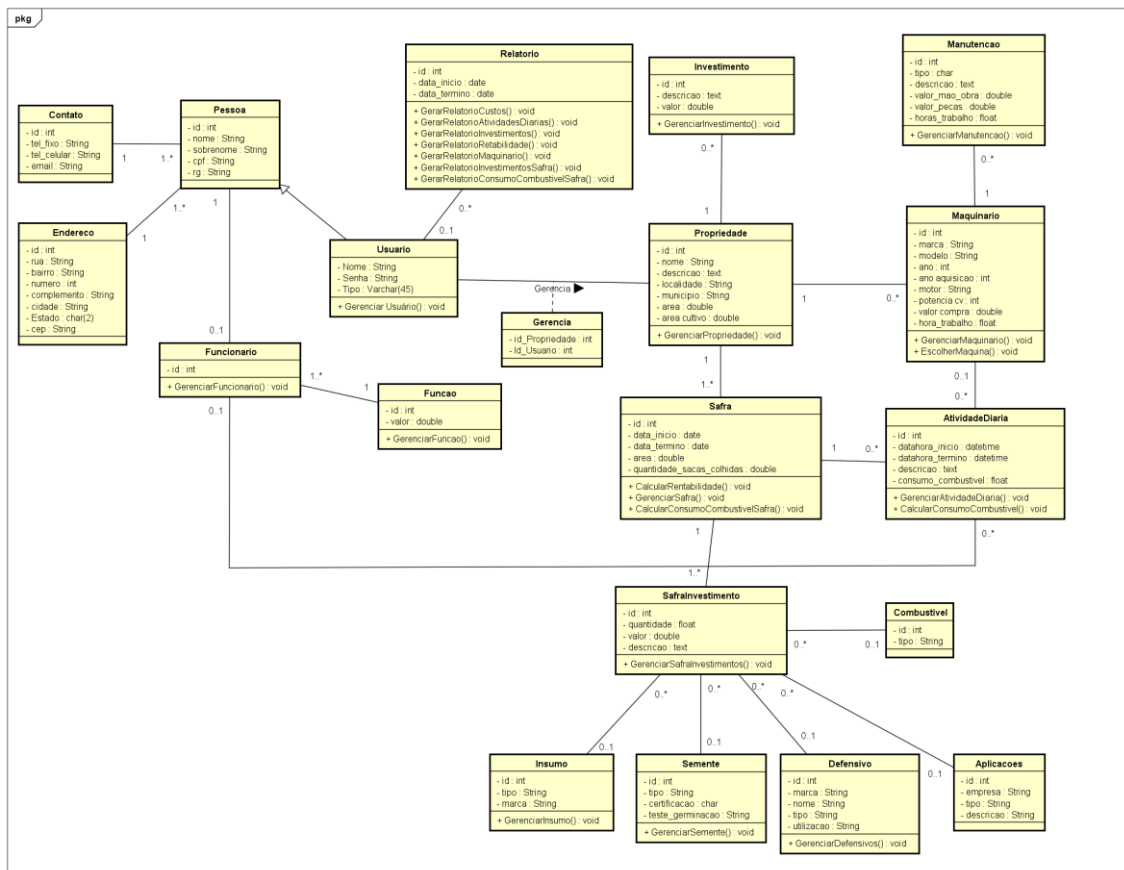


Figura 2. Diagrama de Classes do Sistema

O DER representa a estrutura do banco de dados, demonstrando, por meio de tabelas e seus respectivos relacionamentos e cardinalidades, uma visão ampla do modelo do banco de dados do sistema normalizado, exibido na Figura 3.

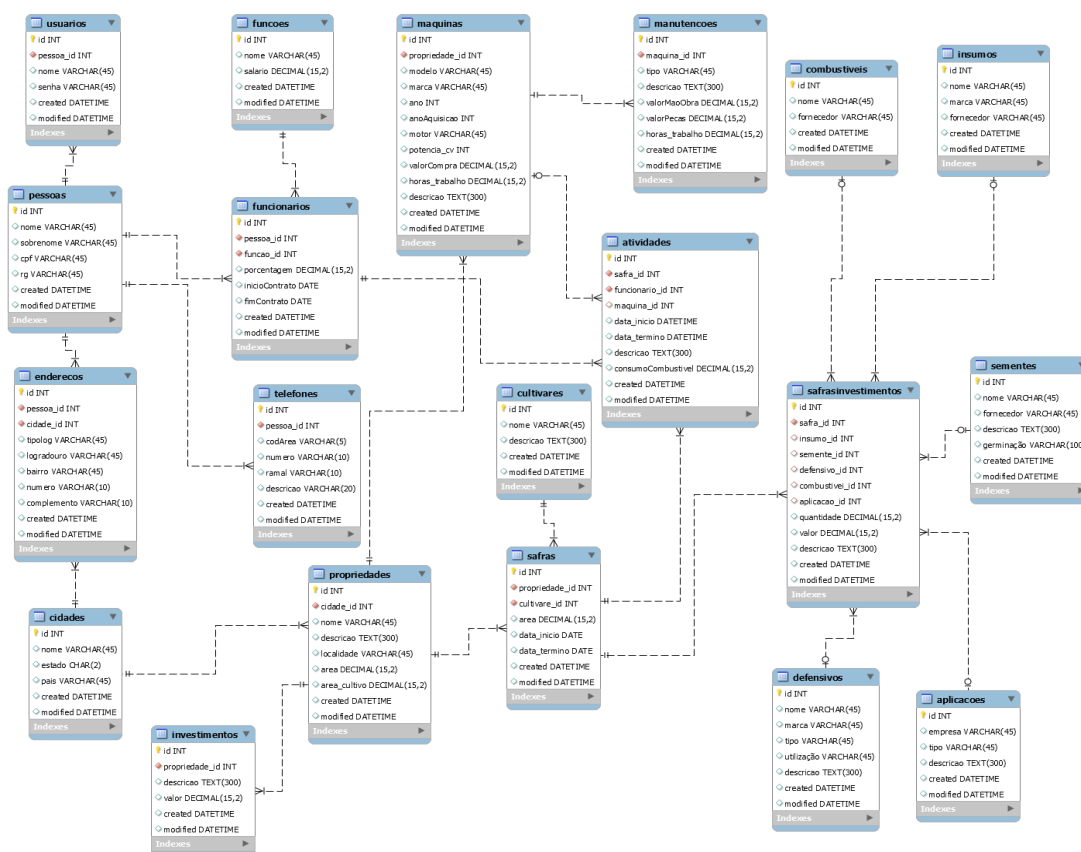


Figura 3. Diagrama de Entidade-Relacionamento (DER) do Sistema

#### 4. Resultados

A interface do sistema web possui um menu de navegação localizado à esquerda, que permite escolher as opções de gerenciamento de: pessoas, endereços e contatos, propriedades, maquinas e manutenções, atividades, safras e cultivares e safra investimentos.

O menu de navegação ainda possui sub-menus, permitindo o gerenciamento das diversas funcionalidades do sistema. A interface de gerenciamento de pessoas, permite ao usuário cadastrar novas pessoas, visualizar, editar e excluir cadastros já existentes (Figura 4).

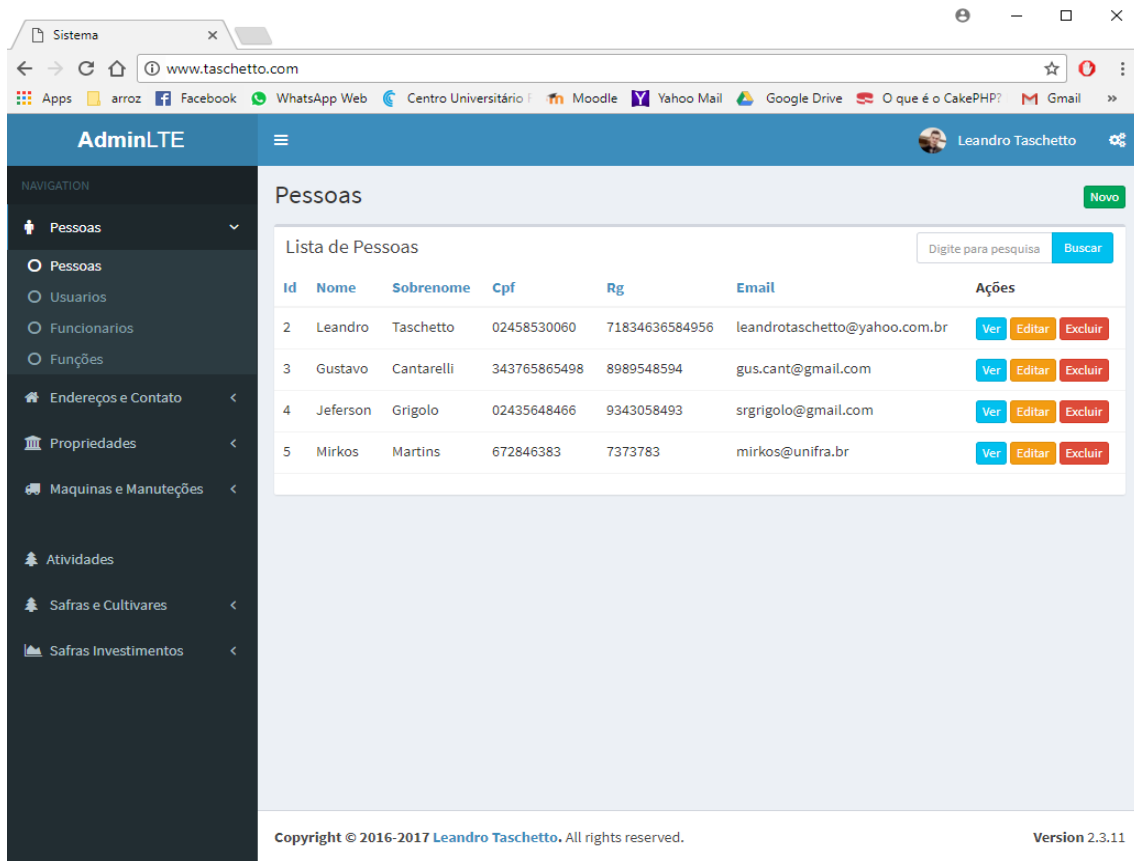


Figura 4. Interface do Sistema – Gerenciamento de Pessoas

A interface de Gerenciamento de Máquinas, permite o cadastro, edição e exclusão de máquinas no sistema. Além de exibir as máquinas cadastradas, é possível visualizar as manutenções realizadas nas mesmas e também as atividades por ela realizada na propriedade (Figura 5).

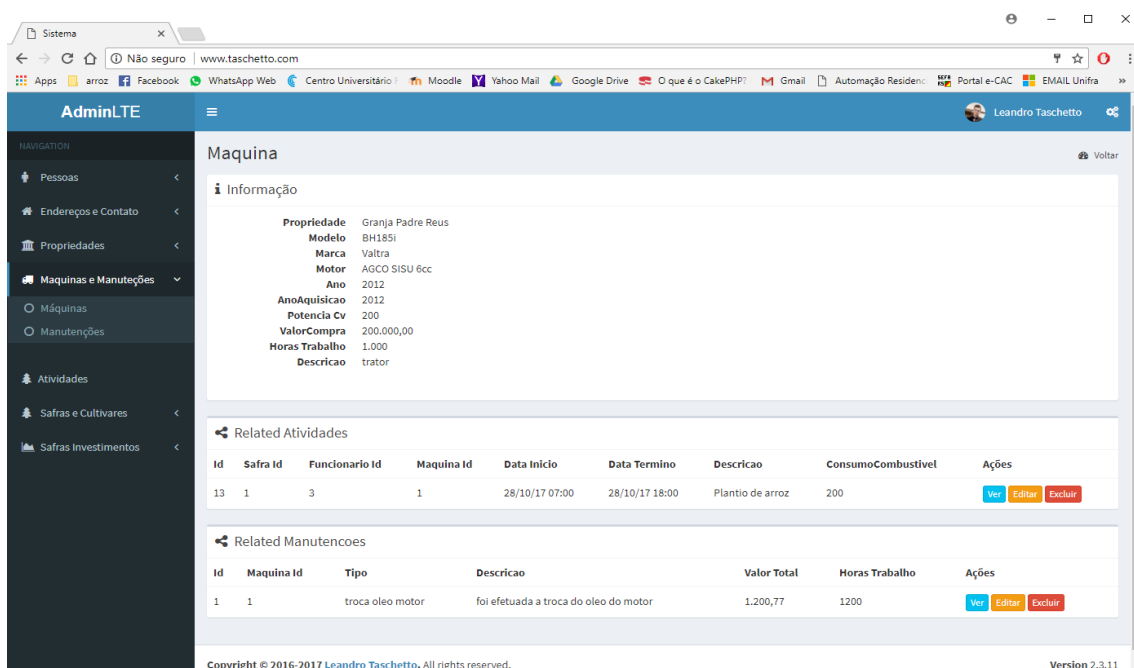


Figura 5. Interface do Sistema – Gerenciamento de Máquinas

A interface de relatório de sementes, mostra detalhes sobre este produto, com informações relevantes como porcentagem de germinação, e uma breve descrição sobre este tipo de semente, podemos destacar o ciclo médio, o vigor inicial, entre outros (Figura 6).

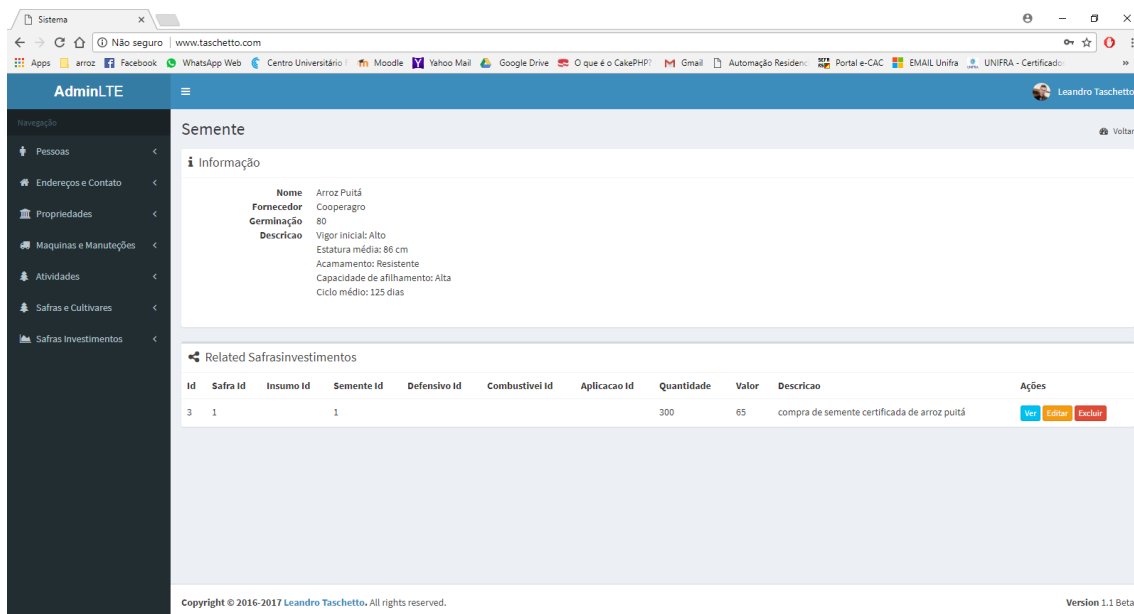


Figura 6. Interface do Sistema – Relatório de Sementes .

Essa interface, mostra detalhes sobre as propriedades cadastradas no sistema, com informações importantes como localidade, cidade área total e área de cultivo, permitindo registro de mais de uma propriedade no sistema, cada uma tendo suas safras, suas máquinas, suas atividades e seus funcionários (Figura 7).

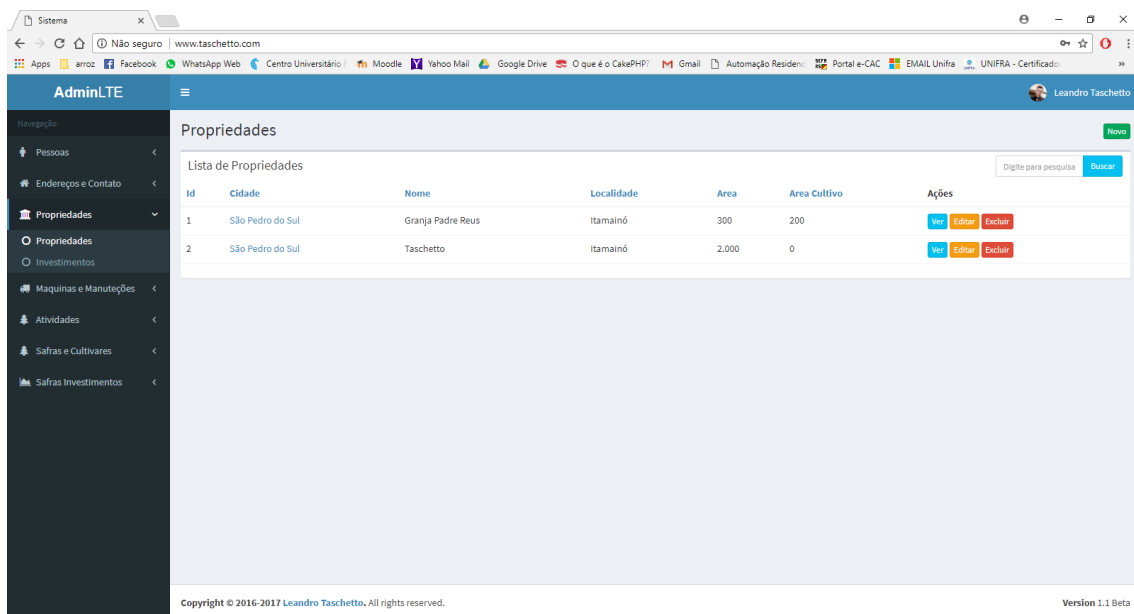


Figura 6. Interface do Sistema – Propriedades .

Essa interface, mostra detalhes sobre os funcionários cadastradas no sistema, contendo informações como função, nome, início de contrato, termino de contrato e porcentagem em sacas relativo a safra, esta apresentada em quantidade de sacas (Figura 7).

Id	Pessoa	Funcao	Porcentagem	InicioContrato	FimContrato	Ações
2	Jeferson	Operador	500	17/01/17	18/06/17	Ver Editar Excluir
3	Gustavo	Operador	200	19/05/17		Ver Editar Excluir
4	Leandro	Operador	2.000	19/05/17		Ver Editar Excluir
6	Mirkos	Operador	200	24/10/17		Ver Editar Excluir

Figura 7. Interface do Sistema – Funcionários .

## 5. Conclusão

Com o desenvolvimento desta proposta tornou-se possível obter relatórios evidenciando informações importantes ao gestor, assim como condições para melhorar as tomadas de decisões, visando a assertividade dos negócios e obtendo, por meio da aplicação, relatórios sobre custos, atividades, máquinas e uso dos insumos utilizados na lavoura. Além disso, a aplicação proporciona o controle de manutenções das máquinas e também o controle das atividades diárias realizadas na propriedade.

O software é de fácil utilização e pode ser acessado facilmente em qualquer local da propriedade, é compatível com diversos tipos de hardwares, como desktops, notebooks, desktops e smartphones, que disponibilizam um navegador web, não necessitando instalação no hardware do cliente.

Como perspectivas para trabalhos futuros, poderá ser elaborado um módulo integrado ao sistema, que trabalhe com a área da pecuária, que também é importante para o setor do agronegócio. Além disso, é possível o desenvolvimento de um módulo que gere relatórios estatísticos da produção na propriedade, mostrando índices de rentabilidade e comparações entre safras.

## Referências

- Alcântara, V. C. (2012) “Gerenciamento de estoque e custos para agricultura através de um sistema informatizado”. Trabalho Final de Graduação, Curso Ciência da Computação, Centro Universitário Franciscano, Santa Maria, RS.
- Barbosa, A. *et al.* (2006) “Metodologia ágil: *Feature Driven Development*”, [http://paginas.fe.up.pt/~aaguilar/es/artigos%20finais/es\\_final\\_22.pdf](http://paginas.fe.up.pt/~aaguilar/es/artigos%20finais/es_final_22.pdf), Abril.
- Boeira, U. S. e Cantarelli, G. S. (2016) “Desenvolvimento de um software para gestão de pecuária”. Trabalho Final de Graduação, Curso Sistemas de Informação, Centro Universitário Franciscano, Santa Maria, RS.
- Bootstrap. (2016) “Bootstrap, Getting started”, <http://getbootstrap.com/getting-started>, Abril.
- Bornia, A. C. (2002) “Análise gerencial de custos”, Porto Alegre, Bookman.
- CakePHP. (2017) “*Cookbook Documentation* Versão 2.x”, [https://book.cakephp.org/2.0/\\_downloads/pt/CakePHPCookbook.pdf](https://book.cakephp.org/2.0/_downloads/pt/CakePHPCookbook.pdf), Março.
- Callado, A. A. C. (2006) “Agronegócio”, São Paulo, 1ª edição.
- Converse, T. e Park, J. (2002) “PHP: a bíblia”, Rio de Janeiro, Elsevier.
- Castro, E. R (2009) “Teoria dos Custos”, SANTOS, Maurinho Luiz dos et al. Microeconomia Aplicada. Visconde do Rio Branco: Suprema, 2009.
- Diehl, R. (1984) “Agricultura geral”. Lisboa, Clássica Editora.
- FDD. (2002) “*Feature Driven Development Processes*”, <http://www.featuredrivendevelopment.com/files/fddprocessesA4.pdf>, Abril.
- Moretti Júnior, J. C. (2016) “Desenvolvimento dirigido por funcionalidades (FDD), Desenvolvimento de software enxuto (LSD) e Processo Unificado Ágil (AUP)”, Universidade Federal do Paraná, Curitiba, PR.
- Neves, M.F (1998) “Sistema Agroindustrial da Cana-de-Açúcar: Caracterização das Transformações entre Empresas de Insumos, Produtores de Cana e Usinas”. In *XXXVI Congresso da Sociedade Brasileira de Economia e Sociologia Rural - SOBER*, Poços de Caldas, MG, 10 a 14 de agosto de 1998, v. 01, p. 559-572.
- Oracle. (2017) “MySQL”, <https://www.oracle.com/br/mysql/index.html>, Março.
- Palmer, S. R. e Felsing, J. M. (2002) “A Practical Guide to Feature-Driven Development”, Prentice Hall.
- Weber, F. A. (2013). “Agrojava: Sistema de gestão para pequenas propriedades rurais”. In *Revista Interdisciplinar de Ensino, Pesquisa e Extensão*, vol. 1, n. 1.
- Wernke, R. (2004) “Gestão de custos: uma abordagem prática”. São Paulo, Atlas, 2ª edição.

## Apêndice A. Planejamento Inicial

Esta seção apresenta, o planejamento e modelagem inicial do sistema.

### 1.1. Desenvolver modelo inicial

Após a análise da proposta, foi possível desenvolver um modelo abrangente, demonstrado por um Diagrama de Domínio apresentado na Figura 8, o qual tem por objetivo mostrar, graficamente, o domínio do sistema, sendo apresentadas suas entidades e dependências, além da maneira como elas se relacionam entre si.

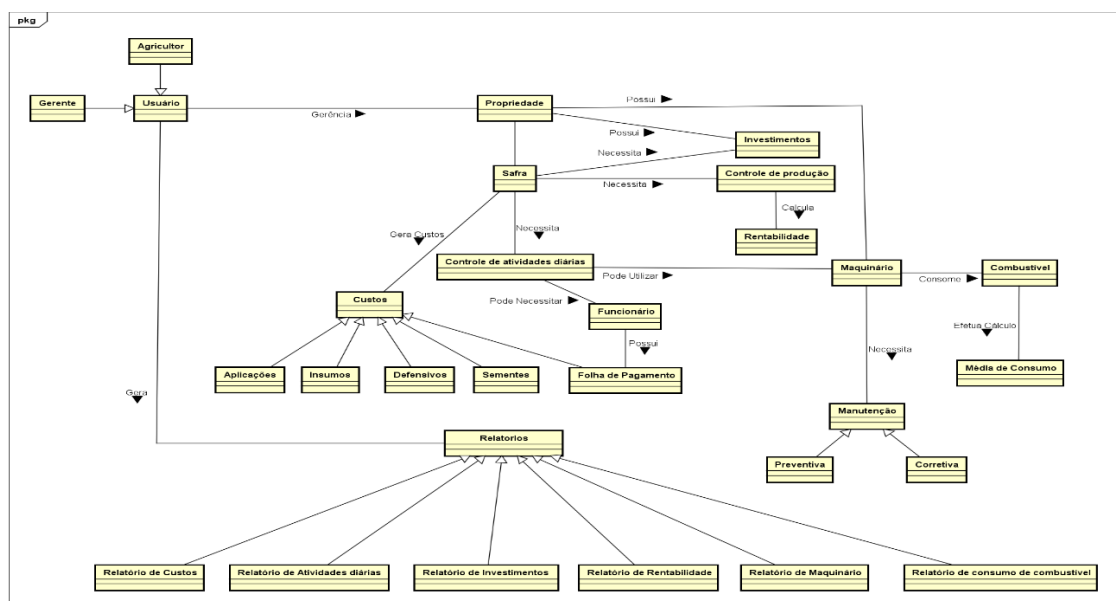


Figura 8. Diagrama de Domínio do sistema

Conforme mostrado na Figura 1, é possível ter uma visão das funcionalidades do sistema proposto, bem como as associações entre as classes conceituais do domínio de aplicação do sistema.

### 1.2. Planejar por funcionalidade

O planejamento por funcionalidade tem por objetivo produzir um plano de negócio, tornando possível determinar a ordem em que as funcionalidades serão desenvolvidas e apresentando sua complexidade para desenvolvimento, assim como a relevância de cada uma. No Quadro 1, também é demonstrado o tempo que levará para ser implementado cada requisito funcional do sistema.

Quadro 1. Planejamento por funcionalidade

Ordem	Funcionalidade	Tempo	Complexidade	Relevância
01	RF02. Gerenciar Propriedade	15 dias	Baixa	Essencial
02	RF07. Gerenciar Safra	15 dias	Média	Essencial
03	RF06. Gerenciar Custos	20 dias	Alta	Essencial

(Continua)



(Conclusão)

Ordem	Funcionalidade	Tempo	Complexidade	Relevância
04	RF03. Gerenciar Maquinário	10 dias	Baixa	Essencial
05	RF04. Gerenciar Manutenção	20 dias	Alta	Essencial
06	RF05. Gerenciar Combustível	10 dias	Média	Essencial
07	RF08. Gerenciar Relatórios	20 dias	Média	Desejável
08	RF01. Gerenciar Usuário	5 dias	Baixa	Essencial

### 1.3. Arquitetar por funcionalidade

Nesta seção, é apresentado o Diagrama de Caso de Uso, exibido na Figura 9, que demonstra uma visão ampla das funcionalidades que serão desenvolvidas na aplicação. Também é demonstrada a interação do autor com as funcionalidades.

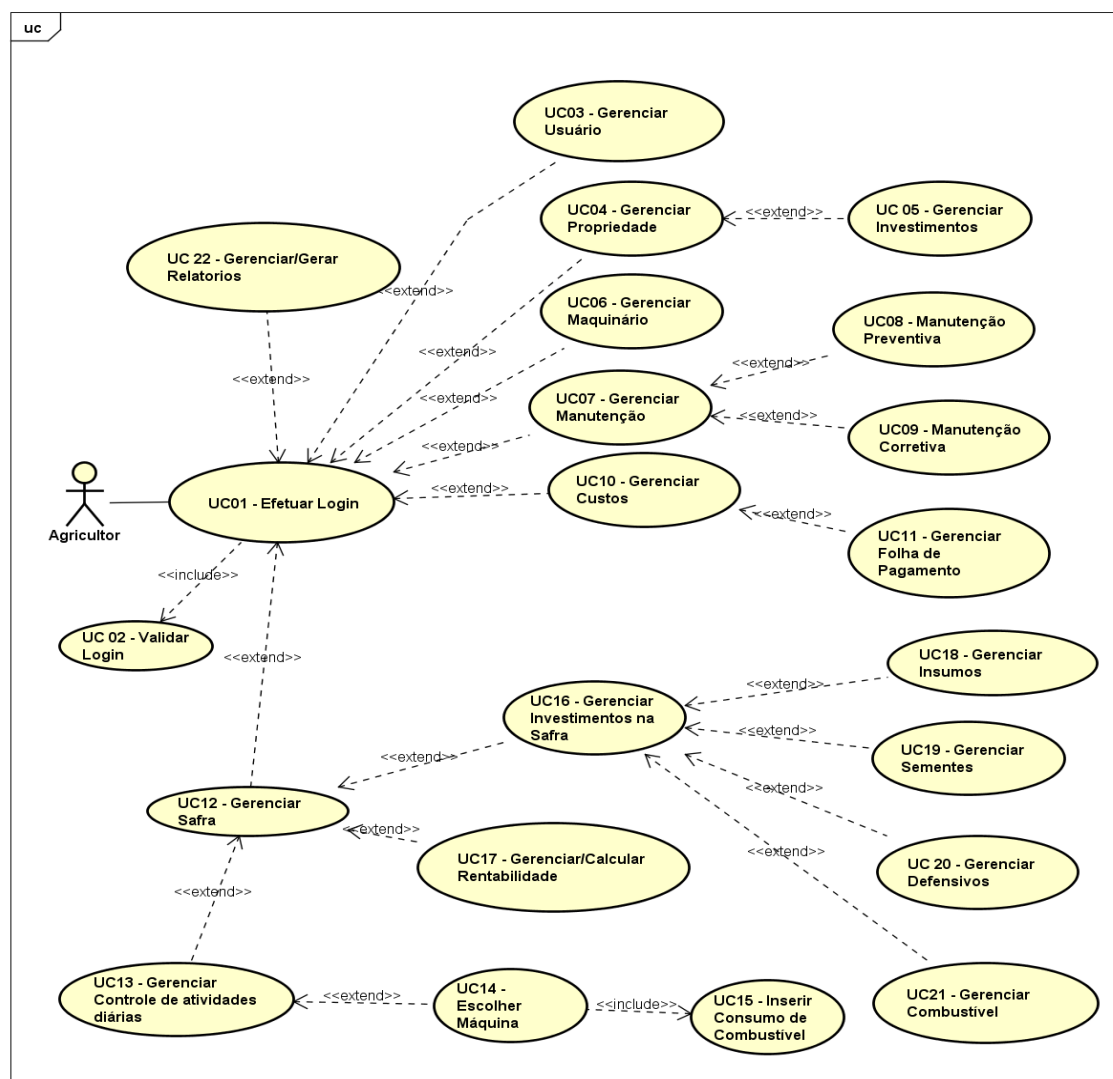


Figura 9. Diagrama de Caso de Uso do Sistema

Após o desenvolvimento do Diagrama de Caso de Uso, são desenvolvidos os descritivos dos casos de uso mais importantes, em que serão demonstrados os processos que podem ocorrer pelo fluxo principal e alternativo, também podendo ocorrer um fluxo de exceção, caso de uso UC16 apresentado no Quadro 2, e demais Diagrama de Caso de Uso importantes para o sistema são mostrados no Apêndice B nos Quadros de 3 a 7.

O Quadro 2 detalha o caso de uso UC16, que descreve o processo gerenciamento de custos, realizado pelo usuário.

Quadro 2. Descritivo de Caso de Uso – UC16 – Gerenciar Investimentos Safra

Identificação	UC16.
Caso de uso	Gerenciar Investimentos na Safra.
Descrição	Usuário poderá cadastrar, alterar ou deletar investimentos na safra.
Ator principal	Agricultor.
Pré-condições	Usuário logado no sistema.
Pós-condições	Custos cadastrado / alterado / excluído no sistema.
Fluxo principal	<p>P01. O usuário deve escolher a opção desejada: Pesquisar [FA01], Novo [FA02], Pesquisar Custos[FA03], Gerenciar Custos [FA04], Gravar [FA05], Limpar [FA06], Apagar [FA07].</p> <p>P02. Sistema grava as informações no banco de dados.</p> <p>P03. Fim do caso de uso.</p>
Fluxo alternativo	<p>FA01. Pesquisar investimentos: o usuário pode pesquisar os investimentos cadastrados no sistema.</p> <p>FA01.1. Após realizar a pesquisa, o sistema preenche todos os campos com os dados do investimento pesquisado.</p> <p>FA01.2. O usuário pode alterar os dados listados.</p> <p>FA02. Novo: Permite que o usuário cadastre um novo custo, deve preencher todos os campos.</p> <p>FA03. Permite pesquisar os custos cadastrados do tipo escolhido no campo Escolher Tipo de Custo.</p> <p>FA04. Gerenciar Custo: conforme o tipo escolhido no campo Escolher Tipo de Custo, o sistema exibe o caso de uso correspondente: UC18 – Gerenciar insumos ou UC19 – Gerenciar Sementes ou UC20 – Gerenciar Defensivos ou UC21 – Gerenciar Combustível.</p> <p>FA05. Gravar: grava no banco de dados os dados que foram preenchidos nos campos.</p> <p>FA05.1. Se não estiverem completos todos os campos de dados FE01.</p>

(Continua)

(Conclusão)

Fluxo alternativo	FA06. Limpar: sistema limpa todos os campos. FA07. Apagar: Sistema exclui investimento pesquisado. FA07.1. Se Investimento não foi pesquisado FE02.
Fluxo de exceção	FE01. Não incluir e exibir a mensagem “Preencha todos os campos”. FE02. Não excluir e exibir a mensagem “Utilize o campo pesquisar investimentos”.
Pontos de extensão	“UC18 – Gerenciar Insumos”. “UC19 – Gerenciar Sementes”. “UC20 – Gerenciar Defensivos”. “UC21 – Gerenciar Combustível”.

A Figura 10 apresenta um esboço de um protótipo de interface, que representa o caso de uso UC16 – Gerenciar Investimentos na Safra.

A Web Page

http://controle.taschetto.com/investimentos\_safra

### Gerenciar Investimentos na Safra

Pesquisar Investimentos:

Safra:

Descrição:

Valor Unitário:

Quantidade:

Escolher tipo de custo:   
Insumos  
Sementes  
Defensivos  
Combustivei

Pesquisar Custo Selecionado:

Figura 10. Protótipo de interface – UC16 – Gerenciar Investimentos na Safra

Nessa fase da metodologia FDD, ainda é apresentado o Diagrama de Atividades do sistema (Figura 11), o qual foi construído usando a ferramenta de modelagem *Astah Community*, que tem por finalidade mostrar os fluxos de execução do *software* proposto. Este diagrama segue os padrões da UML, e possui subatividades, que são utilizados para detalhar as atividades do sistema, estas subatividades são apresentadas nas figuras de 12 a 19 do Apêndice C.

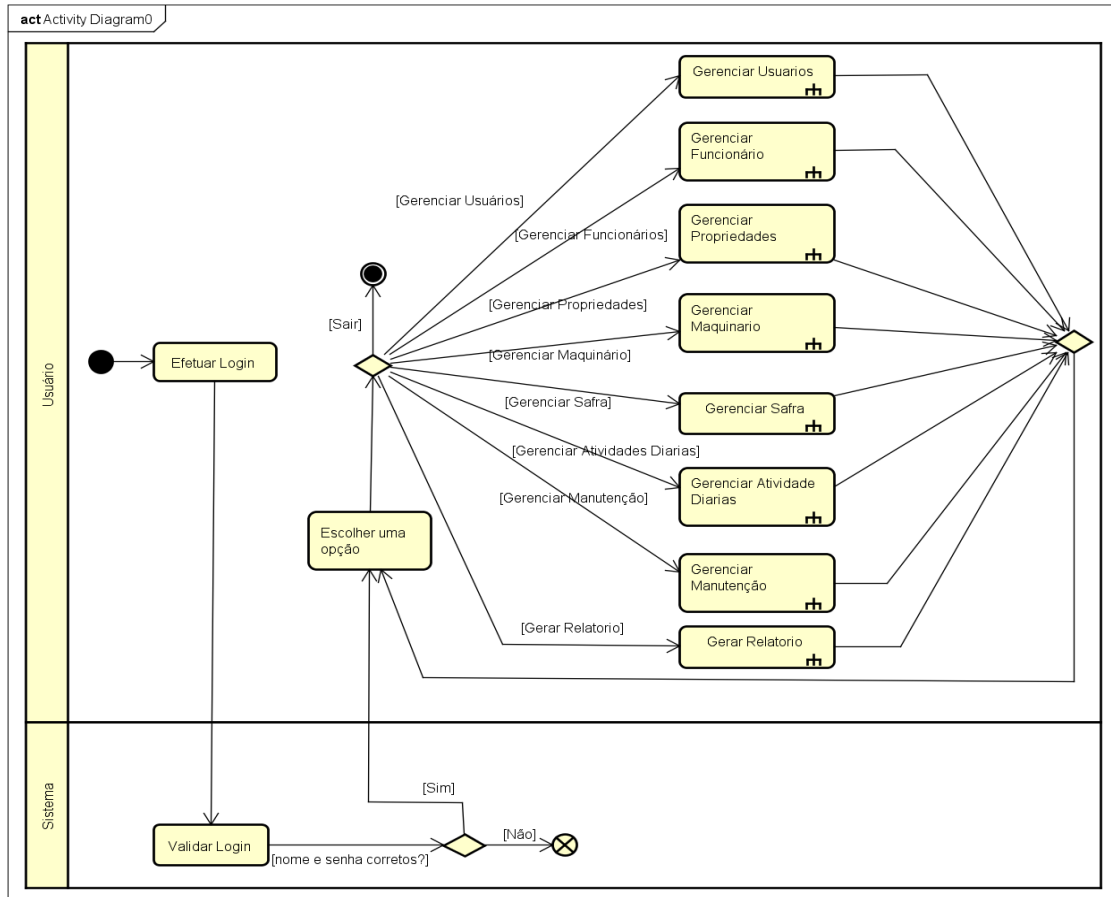


Figura 11. Diagrama de Atividades do sistema

## Apêndice B. Descritivos de Caso de Uso

Nesta seção, os demais descritivos de caso de uso são apresentados nos Quadros 1 ao 5. Estes têm por finalidade detalhar as funcionalidades principais que serão desenvolvidas na aplicação.

Quadro 3. Descritivo de Caso de Uso – UC04 – Gerenciar Propriedades

Identificação	UC 04.
Caso de uso	Gerenciar Propriedades.
Descrição	Usuário poderá cadastrar, alterar, excluir propriedades.
Ator principal	Agricultor.
Pré-condições	Usuário logado no sistema.
Pós-condições	Propriedade cadastrada / alterada / excluída no sistema.
Fluxo principal	P01. O usuário deve escolher a opção desejada: incluir [FA01], alterar [FA02], excluir [FA03]. P02. Sistema grava as informações no banco de dados. P03. Fim do caso de uso.
Fluxo alternativo	FA01. Incluir propriedade. FA01.1. Se a propriedade já existe no banco de dados FE01. FA01.2. Se não estiverem completos todos os campos de dados FE02. FA02. Alterar propriedade. FA02.1 Se não estiverem completos todos os campos de dados FE02. FA03. Excluir propriedade.
Fluxo de Exceção	FE01. Não incluir, exibir mensagem “propriedade já cadastrada no banco de dados”. FE02. Não cadastrar, exibir mensagem “preencha todos os campos”.
Casos de uso estendidos	“UC05 - Gerenciar Investimentos”.

Quadro 4. Descritivo de Caso de Uso – UC05 – Gerenciar Investimentos

Identificação	UC 05.
Caso de uso	Gerenciar Investimentos.
Descrição	Usuário poderá cadastrar, alterar, deletar Investimentos.
Ator principal	Agricultor.
Pré-condições	Usuário logado no sistema.
Pós-condições	Investimentos cadastrado / alterado / excluído no sistema.

(Continua)

(Conclusão)

Fluxo principal	P01. O usuário deve escolher a opção desejada: incluir [FA01], alterar [FA02], excluir [FA03]. P02. Sistema grava as informações no banco de dados. P03. Fim do caso de uso.
Fluxo alternativo	FA01. Incluir Investimento: o usuário pode incluir um novo investimento no sistema. FA01.1. Se Investimento já existe no banco de dados FE01. FA01.2. Se não estiverem completos todos os campos de dados FE02. FA02. Alterar Investimento: o usuário pode alterar um investimento cadastrado no sistema. FA02.1. Se não estiverem completos todos os campos de dados FE02. FA03. Excluir Investimento: o usuário pode excluir um investimento no sistema.
Fluxo de exceção	FE01. Não incluir, exibir mensagem “propriedade já cadastrada no banco de dados”. FE02. Não cadastrar, exibir mensagem “preencha todos os campos”.

Quadro 5. Descritivo de Caso de Uso – UC07 – Gerenciar Manutenção

Identificação	UC 07.
Caso de uso	Gerenciar Manutenção.
Descrição	Usuário poderá escolher qual tipo de manutenção (preventiva ou corretiva) deseja gerenciar.
Ator principal	Agricultor.
Pré-condições	Usuário logado no sistema.
Pós-condições	Manutenção cadastrada / alterada / excluída no sistema.
Fluxo principal	P01. O usuário deve escolher a opção desejada: Gerenciar Manutenção Preventiva [FA01], Gerenciar Manutenção Corretiva [FA02]. P02. Sistema exibe o caso de uso escolhido. P03. Fim do caso de uso.
Fluxo alternativo	FA01. Exibir UC 08 Manutenção Preventiva. FA02. Exibir UC 09 Manutenção Corretiva.
Casos de uso estendidos	“UC08 - Manutenção Preventiva”. “UC09 - Manutenção Corretiva”.

Quadro 6. Descritivo de Caso de Uso – UC08 – Gerenciar Manutenção Preventiva

Identificação	UC 08.
Caso de uso	Manutenção Preventiva.
Descrição	Usuário poderá cadastrar, alterar, deletar Manutenções preventivas.
Ator principal	Agricultor.
Pré-condições	Usuário logado no sistema.
Pós-condições	Manutenção preventiva cadastrada / alterada / excluída no sistema.
Fluxo principal	P01. O usuário deve escolher a opção desejada: incluir [FA01], alterar [FA02], excluir [FA03]. P02. Sistema grava as informações no banco de dados. P03. Fim do caso de uso.
Fluxo alternativo	FA01. Incluir manutenção preventiva: o usuário deve preencher todos os campos para incluir uma nova manutenção preventiva. FA01.1 Se a manutenção preventiva já foi cadastrada no sistema FE01. FA01.2. Se todos os campos não forem preenchidos FE02. FA02. Alterar manutenção preventiva: o usuário deve alterar os dados de uma manutenção preventiva. FA02.1. Se todos os campos não forem preenchidos FE02. FA03. Excluir manutenção preventiva: o usuário deve excluir uma manutenção preventiva do sistema.
Fluxo de exceção	FE01. Não incluir, exibir mensagem “manutenção já cadastrada no banco de dados”. FE02. Não cadastrar, exibir mensagem “preencha todos os campos”.

Quadro 7. Descritivo de Caso de Uso – UC09 – Gerenciar Manutenção Corretiva

Identificação	UC 09.
Caso de uso	Manutenção Corretiva.
Descrição	Usuário poderá cadastrar, alterar, deletar manutenções corretivas.
Ator principal	Agricultor.
Pré-condições	Usuário logado no sistema.
Pós-condições	Manutenção corretiva cadastrada / alterada / excluída no sistema.
Fluxo principal	P01. O usuário deve escolher a opção desejada: incluir [FA01], alterar [FA02], excluir [FA03]. P02. Sistema grava as informações no banco de dados. P03. Fim do caso de uso.

(Continua)

(Conclusão)

Fluxo alternativo	<p>FA01. Incluir manutenção corretiva: o usuário deve preencher todos os campos para incluir uma nova manutenção corretiva.</p> <p>FA01.1 Se a manutenção corretiva já foi cadastrada no sistema FE01.</p> <p>FA01.2. Se todos os campos não forem preenchidos FE02.</p> <p>FA02. Alterar manutenção corretiva: o usuário deve alterar os dados de uma manutenção corretiva.</p> <p>FA02.1. Se todos os campos não forem preenchidos FE02.</p> <p>FA03. Excluir manutenção corretiva: o usuário deve excluir uma manutenção corretiva do sistema.</p>
Fluxo de exceção	<p>FE01. Não incluir, exibir mensagem “manutenção já cadastrada no banco de dados”.</p> <p>FE02. Não cadastrar, exibir mensagem “preencha todos os campos”.</p>



## Apêndice C. Diagrama de Atividades do sistema

Esta seção apresenta, nas Figuras 12 a 19, a continuação do Diagrama de Atividades do sistema, que tem por finalidade mostrar os fluxos de execução do software proposto.

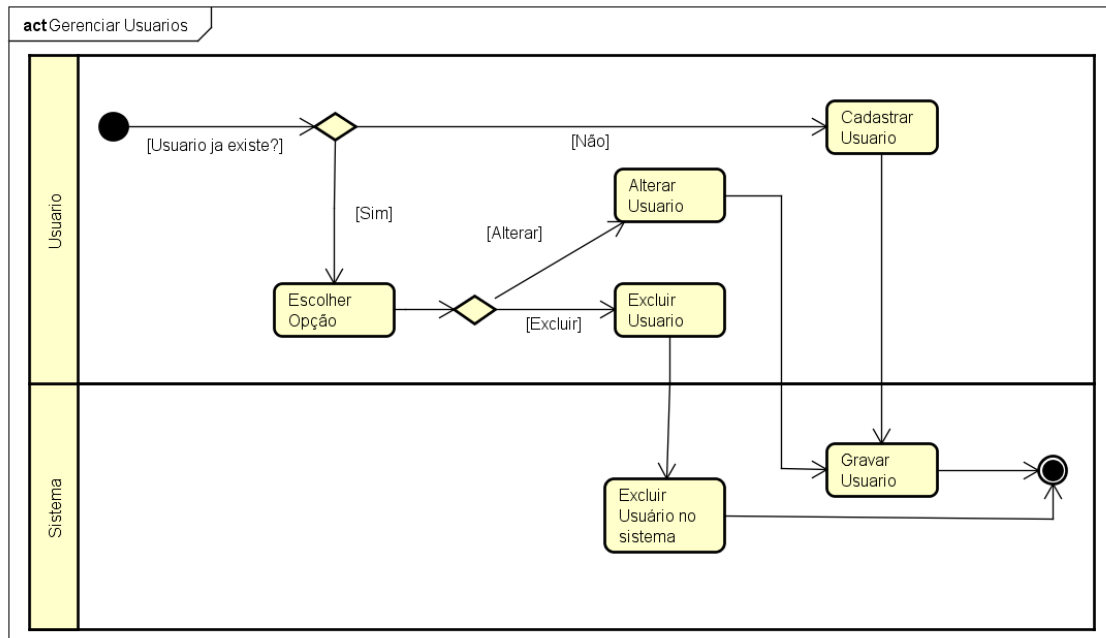


Figura 12. Diagrama de Atividades – Gerenciar Usuários

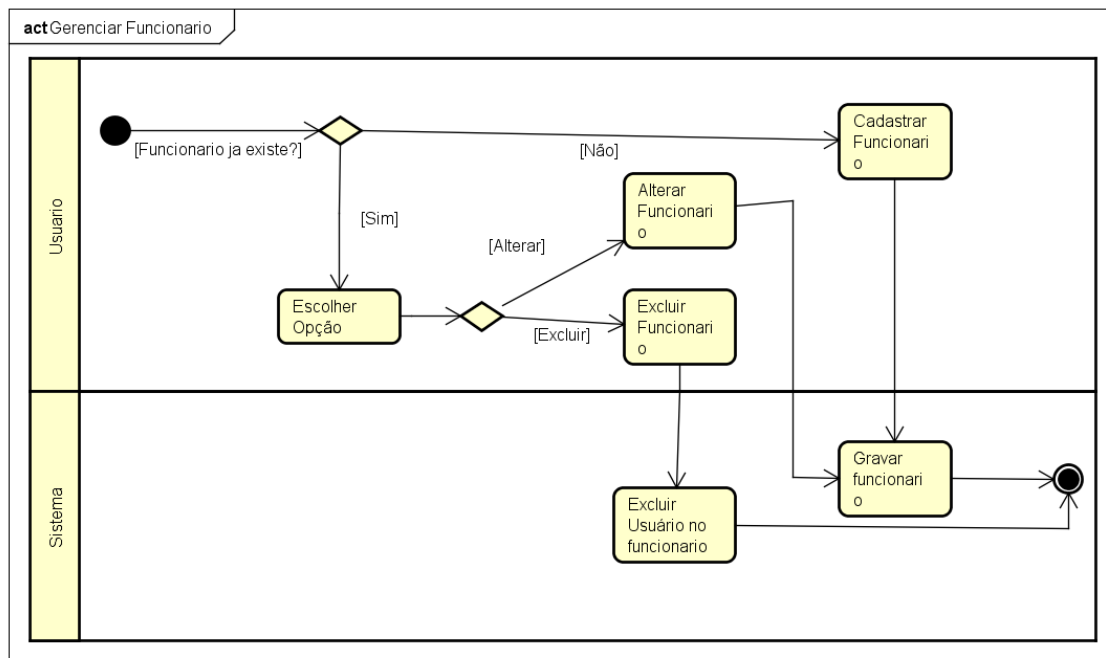


Figura 13. Diagrama de Atividades – Gerenciar Funcionário

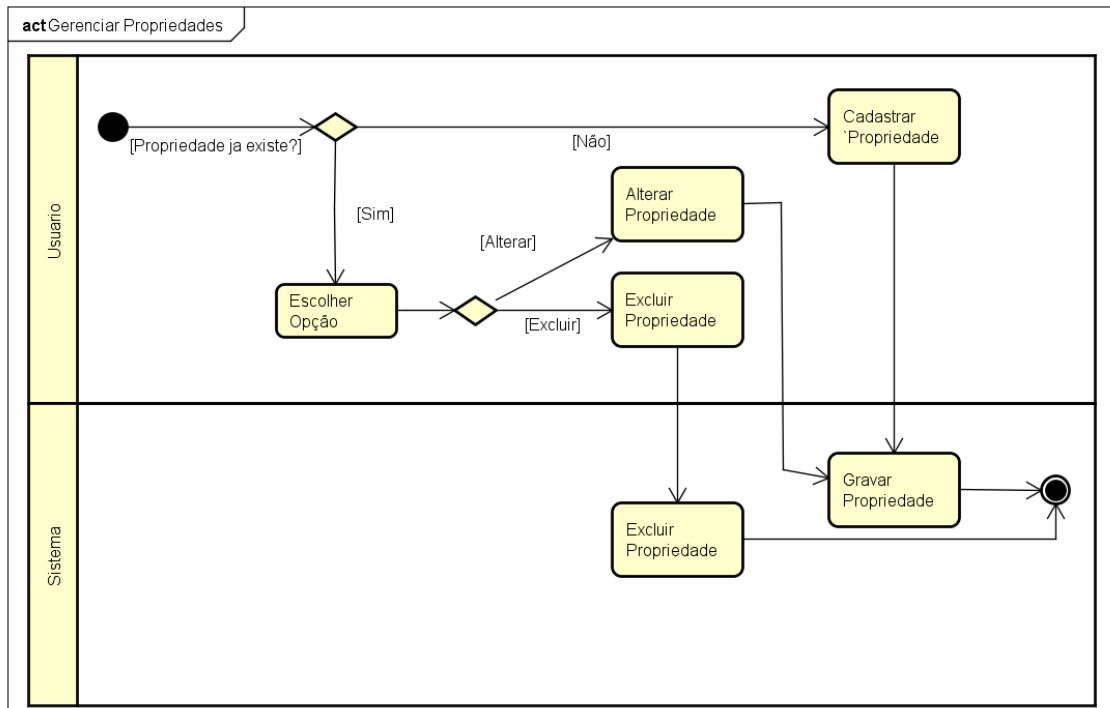


Figura 14. Diagrama de Atividades – Gerenciar Propriedade

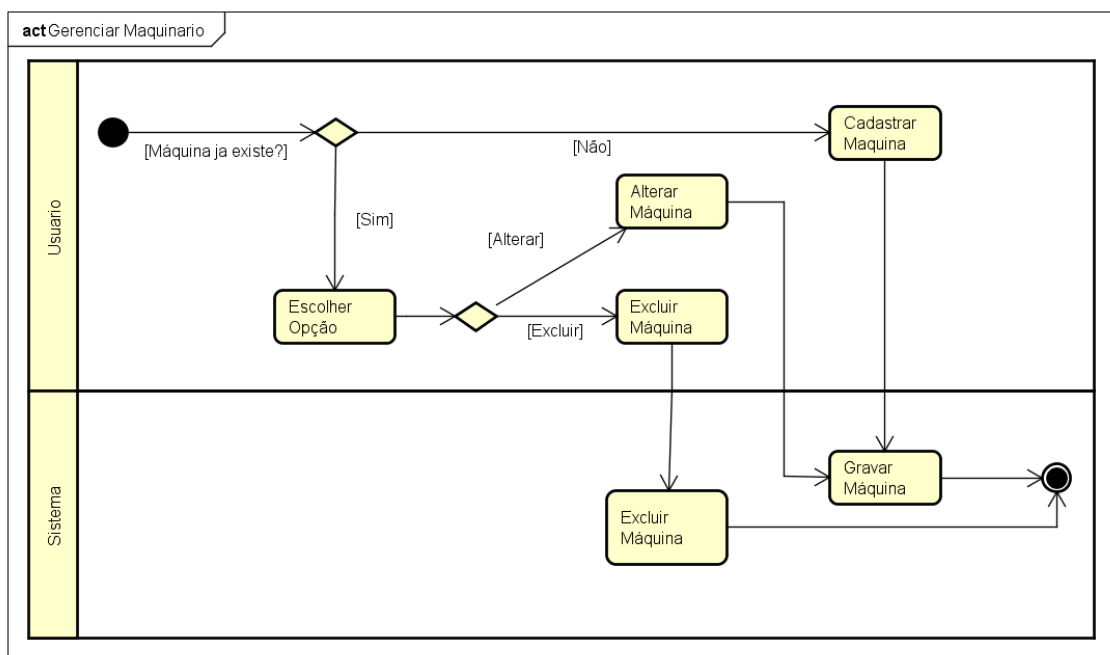


Figura 15. Diagrama de Atividades – Gerenciar Maquinário

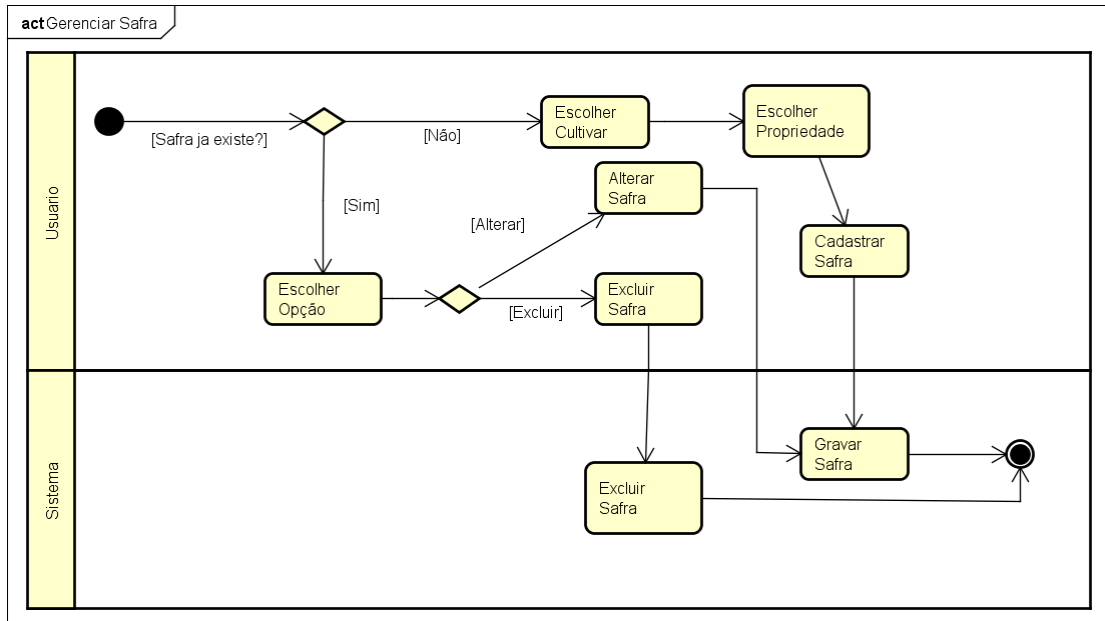


Figura 16. Diagrama de Atividades – Gerenciar Safra

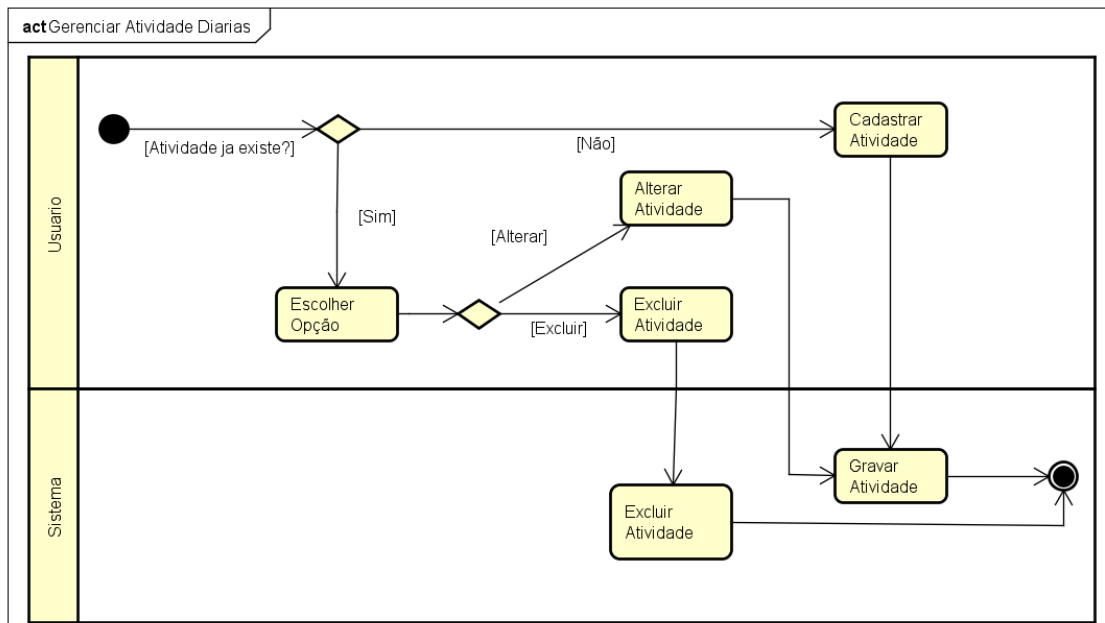


Figura 17. Diagrama de Atividades – Gerenciar Atividades Diárias

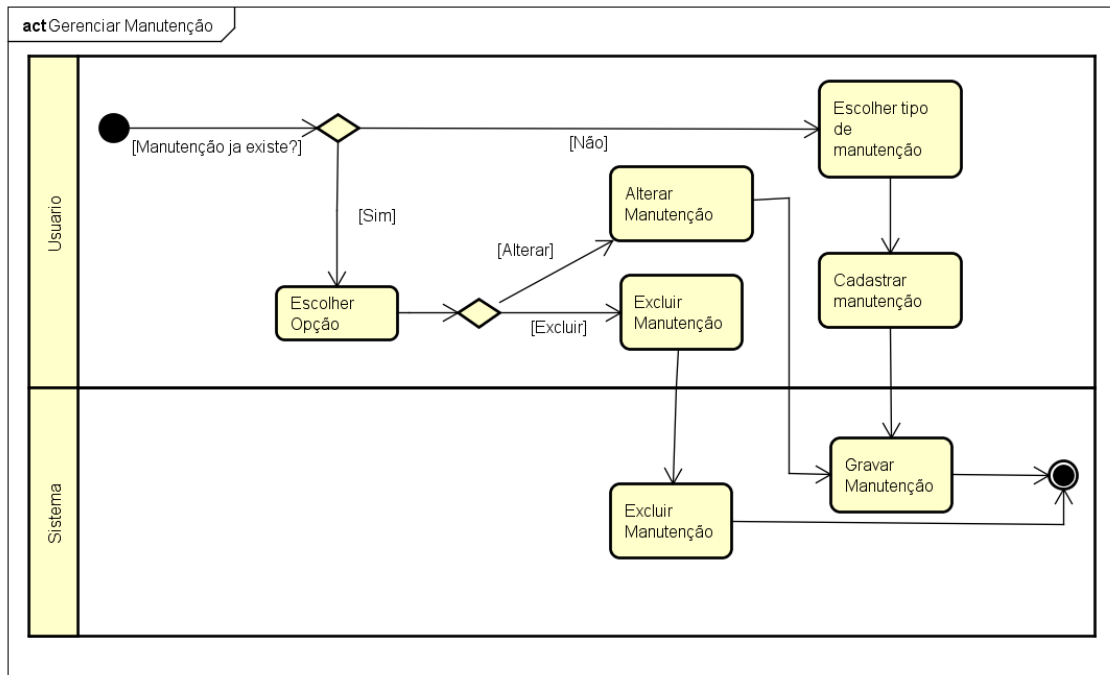


Figura 18. Diagrama de Atividades – Gerenciar Manutenção

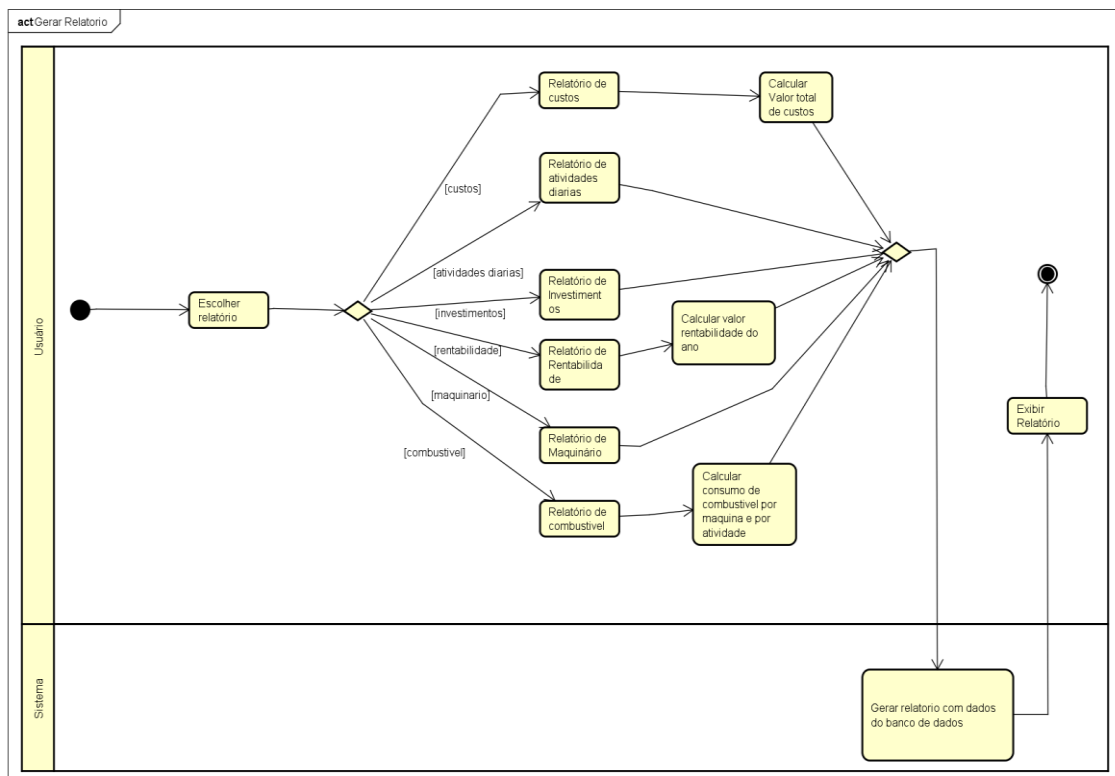


Figura 19. Diagrama de Atividades – Gerar Relatório