Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

Alison Aloisio Brauner

SISTEMA GERENCIADOR DE LACTICINIO

Assis

2014

Alison Aloisio Brauner

SISTEMA GERENCIADOR DE LACTICINIO

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis como requisito parcial a conclusão do Curso de Analise e Desenvolvimento de Sistemas.

Orientador: Alex Sandro Romeo De Souza Poletto.

Área de Concentração:

Assis

2014

FICHA CATALOGRAFICA

BRAUNER, Alison Aloisio.

Sistema Gerenciador de Lacticínio / Alison Aloisio Brauner.

Fundação Educacional do Município de Assis – FEMA- Assis, 2014.

P. 63.

Orientador: Alex Sandro Romeo De Souza Poletto.

Trabalho de Conclusão de Curso – Instituto Municipal De Ensino Superior de Assis – IMESA.

1. Informatização. 2. Registro. 3. Controle.

CDD: 001.61

Biblioteca da FEMA.

SISTEMA DE GERENCIAMENTO DE LACTICÍNIO

ALISON ALOISIO BRAUNER

Trabalho Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito parcial a conclusão do Curso de Analise e Desenvolvimento de Sistemas analisado pela seguinte comissão organizadora.

Orientador: _____.

Analisador: _____.

Assis

2014

DEDICATÓRIA

Este trabalho dedicado primeiramente a Deus ter me concedido a inteligência, paciência e para minha família pelo apoio e incentivo de continuar com os estudos.

AGRADECIMENTOS

Agradeço a Deus, pelo caminho que ele me proporcionou e as oportunidades surgiram.

Ao orientador e Prof. Alex Sandro Romeo De Souza Poletto pela disposição, apoio, paciência para o desenvolvimento deste projeto.

Os meus colegas que me ajudaram em diversos momentos, no decorrer do Curso de Analise e Desenvolvimento de Sistemas.

E claro, a toda minha família que sempre me incentivaram a estudar.

RESUMO

Com o surgimento da globalização, avanços tecnológicos que ocorreram nos últimos tempos houve o desenvolvimento de todas as atividades econômicas, à informática teve um papel fundamental, atuando diretamente na gestão de negócios nesse contexto: suprindo as necessidades dos seres humanos auxiliando desenvolvimento das atividades, realizando grandes movimentações de informações. Juntamente este software aborda diversos conceitos envolvidos na criação do sistema desde analise dos requisitos, diagramas e documentação descrevendo todas suas funcionalidades. Sistema tem como idealização auxiliar o gerenciamento de lacticínio realizando registros das atividades desempenhadas diariamente, devido à falta de softwares voltados para o agronegócio justifica a ideia desenvolver o sistema voltado para este setor econômico. Com isso o produtor terá maior controle de seu negocio, uma vez que existe um sistema auxiliando dispensa o uso muitas vezes excessivo de papeis, podem ser extraviados então se perde a consistência dos dados.

Palavras-Chave: 1.Informatização; 2.Registros; 3.Controle; 4.Lacticinio.

ABSTRACT

With the emergence of globalization, technological advances that have occurred in recent times there has been the development of all economic activities, the computer played a key role, acting directly in business management in this context: meeting human needs aiding development activities, conducting large movements of information. Along this software covers various concepts involved in the creation of the system from requirements analysis, diagrams and documentation describing all its functionalities. System is idealized help manage dairy performing records of activities performed daily, due to lack of software oriented agribusiness justifies the idea to develop the system toward this economic sector. With that the producer will have greater control of their business, since there is a helping relieve the often excessive use of system roles can be lost then loses data consistency.

Keywords: 1.Computerization; 2.Registries; 3.Control; 4. Dairy Product.

LISTA DE ILUSTRAÇÕES

Figura 1- Representa o usuário do sistema. 19
Figura 2 - Balão descreve o requisito como exemplo é “Cadastrar Cliente”. 19
Figura 3 - Representação interação entre usuário e caso de uso. 20
Figura 4 - Mostra o caso de uso geral do sistema.....21
Figura 5- Mostra o caso de uso – Cadastro de usuário do sistema. 22
Figura 6– Mostra o caso de uso – Cadastro dos animais/rebanho. 23
Figura 7– Mostra o caso de uso – Compra de combustível. 24
Figura 8– Mostra o caso de uso – Manter Fornecedor.....26
Figura 9– Mostra o caso de uso – Manter fornecedor de Leite. 27
Figura 10– Mostra o caso de uso – Registro de Insumos. 29
Figura 11- Mostra o caso de uso – Manter Manutenção. 30
Figura 12 - Mostra o caso de uso – Manter Maquinário. 31
Figura 13– Mostra o caso de uso – Manter Motorista. 33
Figura 14 – Mostra o caso de uso – Manter Produção.....34
Figura 15– Mostra o caso de uso – Manter Propriedade. 36
Figura 16– Mostra o caso de uso – Manter Veículos. 37
Figura 17– Mostra o caso de uso – Manter Funcionários. 38
Figura 18– Mostra o caso de uso – Manter Histórico de Compra Combustível. 39
Figura 19– Mostra o caso de uso – Manter Histórico de Produção.....40
Figura 20 - Diagrama de Sequencia - Login de usuário do sistema.....42
Figura 21 - Diagrama de sequencia - Cadastro de Propriedade. 43
Figura 22 - Diagrama de Sequencia - Registro de Fornecedor. 43
Figura 23 - Diagrama de Sequencia - Cadastro Fornecedor de Leite. 44
Figura 24 - Diagrama de Sequencia - Registro Compra de Combustível.....44
Figura 25- Diagrama de Sequencia - Registra os funcionários. 45
Figura 26 - Diagrama de Sequencia - Manter Motoristas dos veículos de transporte mercadoria. 45
Figura 27- Diagrama de Sequencia - Registro de Produção.....46
Figura 28 - Diagrama de Sequencia - Controle de Rebanho 46
Figura 29 - Diagrama de Sequencia - Registro de Insumos.....47
Figura 30 - Diagrama de Sequencia - Controle de Manutenções Maquinário. 47
Figura 31 - Diagrama de Sequencia - Registra os Maquinários da propriedade. 48
Figura 32 - Diagrama de Sequencia - Registro de veículos pertencem à empresa. .48
Figura 33 - Diagrama de Sequencia - Histórico de Produção. 49
Figura 34 - Diagrama de Sequencia - Histórico de Compras de combustível. 49
Figura 35 - Diagrama de Classes - Contem as classes do sistema. 51
Figura 36 - Diagrama - EAP - Estrutura Analítica Projeto. 52
Figura 37- Diagrama Banco de Dados. 54

Figura 38 – Camadas.....56
Figura 39 - Classe de Compra de combustível. 57
Figura 40 - BLL de Fornecedor Produtos. 60
Figura 41 - Validação de CPF. 61
Figura 42 – Código do evento validação do botão validar.....62

SUMÁRIO

1- INTRODUÇÃO 13
1.1. OBJETIVOS 13
1.2. JUSTIFICATIVAS e MOTIVAÇÕES 14
1.3. PUBLICO ALVO 14
2. METODO DE DESENVOLVIMENTO 15
2.1. METODOLOGIA DE ANALISE. 15
2.3. C# 15
2.4. DBDesigner 16
2.5. MICROSOFT SQL SERVER 2012 16
2.6. ASTAH Community (Comunidade) 17
2.7. VISUAL STUDIO EXPRESS 2012 17
2.8. MODELAGEM DO SISTEMA 18
3. ANÁLISE DO SISTEMA 18
3.1. DIAGRAMA DE CASO DE USO GERAL 20
3.2. NARRATIVAS DE CASO DE USO 22
3.3. DIAGRAMAS DE SEQUENCIA 42
3.4. DIAGRAMA DE CLASSE 50
3.5. WBS – WORK BREAKDOWN STRUCTURE 52
3.6. DIAGRAMA (E-R) ENTIDADE RELACIONAMENTO 53
4.1. DATA ACCESS LAYER (DAL) 55
4.2. BUSINESS LOGIC LAYER (BLL) 55
4.3. USER INTERFACE (UI) 55
5. CONSIDERAÇÕES FINAIS 62
6. REFERENCIAS 63

1- INTRODUÇÃO

A Agropecuária expressão que descreve de maneira agrupada a pecuária e a agricultura há muito desempenha papel importante no cenário econômico nacional, além de ser sida uma das primeiras atividades econômicas a ser desenvolvida em nosso país, mas de forma geral a produção rural esta em expansão. Neste setor a produção de leite de vaca no Brasil vem crescendo desde 1974 até os dias atuais, naquele ano foi produzido 7,1 bilhões de litros de leite, já em 2011 o Brasil alcançou 32,1 bilhões de litros apresentado crescimento superior a 350% neste período segundo dados de pesquisa do (IBGE – Instituto Brasileiro de Geografia e Estatística). Todo este crescimento econômico merece nosso respeito e incentivo para que não pare de crescer, e para que isto melhore ainda mais aplicação de tecnologia, pode contribuir para a melhoria das condições de produção do campo.
Pois quando se fala em softwares que dão suporte para o produtor, existe poucas opções muitas vezes essas incompletas, deixam a desejar ou seja não há disponível softwares ajudam no desenvolvimento das atividades econômicas oferecendo suporte para a tomada de decisões. Algo que já foi implantando e que vem colendo resultados positivos é agricultura de precisão, por isso deve haver uma gama maior de produtos computacionais que ajudem no dia-a-dia do produtor rural.

1.1. OBJETIVOS

Produzir um sistema que auxilie o produtor rural, nas atividades diárias realizadas em sua propriedade ajudando na sua melhor gestão, contribuir para um melhor ambiente de trabalho no campo. Possibilitando maior controle o software ira ajudar o pecuarista a gerir os recursos utilizados sejam eles humanos, financeiros ou materiais.

1.2. JUSTIFICATIVAS e MOTIVAÇÕES

São muitos os produtos, porém, são poucos os que conhecem ou fazem uso de software, que seja voltado para sua atividade econômica, seja por não saberem e pela falta de produtos no mercado. Com o aumento da demanda teve que haver maior produção consequente os gastos são maiores, necessita de um controle eficaz para apurar os custos envolvidos. Tornou-se importante existência de sistema para cooperar com a tomada de decisão.

1.3. PUBLICO ALVO

Este software tem como ideal auxiliar o produtor agropecuário, mas especificamente os que atuam no setor de gado leiteiro a desempenhar as atividades rurais através de dados coletados, em seguida processados gerando registros ou seja produz informação.

2. METODO DE DESENVOLVIMENTO

2.1. METODOLOGIA DE ANALISE.

O analise e desenvolvimento deste software foi empregado à programação orientada a objetos, na qual consiste em organizar objetos do mundo real nos códigos e incorporando sua estrutura, simulando seu real comportamento.

Segundo C.Lee et al., (2001, p.24) “definiremos um objeto como uma unidade de software que consiste de atributos (dados) e dos métodos (código) que atuam sobre aqueles dados. Os dados não são diretamente acessíveis aos usuários do objeto.”

2.3. C#

O C# linguagem de programação desenvolvida pela Microsoft é uma das mais utilizadas no mundo quando se diz em termos de software comercial competindo diretamente com o Java criado pela equipe liderada por Anders Heijlsberg e Scott Wiltamuth. Com Visual Studio e plataforma. NET, C# é a evolução do C e C++ fornece diversos recursos importantes para a o programador como programação orientada a objetos, elementos gráficos, interface gráfica com usuário (GUI), String, tratamento de exceções. Integrada com a plataforma .NET possibilita aplicativos na Web possam ser utilizados, em diversos dispositivos desde celulares até computadores de mesa ou seja facilitando a migração entre diapositivos uma vez que C# tem raízes em C, C++ e Java.

Segundo Sharp (2007)

> O Microsoft Visual C# é uma poderosa linguagem, mas simples, voltada principalmente para os desenvolvedores que criam aplicativos usando o Microsoft .NET Framework. Ela herda grande parte dos melhores recursos do C + + e Microsoft Visual Basic [...] resultando em uma linguagem mais limpa e lógica.
>
> (Sharp, 2007, 4)

C# linguagem de programação orientada a objetos, os programas são criados usando-se um IDE (Integrated Development Environment – ambiente desenvolvimento integrado) possibilita que o programador possa criar executar, testar, e depurar programas ajuda a diminuir o tempo de desenvolvimento os processos de envolvidos que utilizam IDE é normalmente denominado de RAD (Rapid Application Development – desenvolvimento rápido de aplicativos). Além disso, também permite grau de intercambio entre linguagens: componentes de software de diferentes linguagens podem interagir entre-se, C# também interagi pela Internet com padrões como SOAP (Simple Object Access Protocol – protocolo de acesso simples) e XML (Extensible Markup Languague – linguagem de marcação extensível), portanto a incorporação do .NET no C# trouxe diversos avanços na programação.

2.4. DBDesigner

Segundo o artigo "DBDesigner[1] – Sistema de design para banco de dados integrando o projeto do banco auxiliando em varias etapas desenvolvimento desde a modelagem, criação das tabelas e manutenção em um só ambiente de desenvolvimento. Combinando características profissionais com a interface simples clara e objetiva, sendo a forma mais eficiente para lidar com banco de dados, além disso, DBDesigner é plataforma OpenSource."

2.5. MICROSOFT SQL SERVER 2012

Sql Server [2] trata-se de sistema SGBD (Sistema Gerenciador De Banco De Dados) desenvolvido pela Microsoft um software cuja sua principal é função armazenar, recuperar informações solicitadas por outros programas computacionais, seja aqueles no mesmo computador ou execução em outra maquina está opera através

[1] DBDesigner disponível em http://fabforce.net/dbdesigner4/
[2] Segundo artigo disponível no link http://www.microsoft.com/pt-br/server-cloud/products/sql-server/

de uma rede, SQL Server sistema de banco de dados de alto desempenho projetado para ser executado nos sistemas Windows NT Server. A Microsoft integrou o SQL Server com Internet Information Server (IIS ou Servidor de Informações da Internet) ajudar a criar aplicações complexas baseadas na Web.

Existem diferentes edições do Microsoft SQL Server destinadas a diferentes públicos e varias cargas de trabalho variam desde pequenas aplicações buscam dados no mesmo computador, aquelas que são acessadas por milhões de usuários manipulam grandes quantidades de dados.

2.6. ASTAH Community (Comunidade)

Ferramenta poderosa para modelagem UML (Unified Modeling Language) do sistema integrando os diagramas de fluxo de dados, gerando uma visão do escopo da estrutura do sistema, Astah[3] está pronto para uso no momento em que você baixa-lo para ser rápida e facilmente reforçada, refinar o processo de desenvolvimento da modelagem dos dados.

2.7. VISUAL STUDIO EXPRESS 2012

Segundo Marcos (2012), “fermenta desenvolvida pela Microsoft conhecida como uma IDE (Integrated Development Environment) Visual Studio software com editor texto muito poderoso, onde você digita seus códigos em uma dada linguagem de programação possuidor de diversas ferramentas auxiliares para depuração (detecção erros de sintaxe do código e de logica de seus programadores), integrado com NET.Framework facilidade na interação com banco de dados.”

[3] Disponível no link http://translate.google.com.br/translate?hl=pt-BR&sl=en&u=http://astah.net/editions/community&prev=/search%3Fq%3Dastah%2Bcommunity%26biw%3D1366%26bih%3D657

Vale ressaltar que existem versões pagas, utilizadas por empresas, para produzirem softwares, contudo a Microsoft disponibiliza versões simplificadas da ferramenta, para uso nível domestico e não comerciais chamadas de versões Express.

2.8. MODELAGEM DO SISTEMA

Para a modelagem do sistema foi utilizado a Linguagem de Modelagem Unificada (UML) serve para criar a visão do sistema, ou seja, gera de forma visual o seu escopo, descreve o sistema de computador abordando vários pontos.

Segundo C.Lee et al., (2001, p.505) "A UML é uma linguagem de modelagem para documentar e visualizar os artefatos que especificamos e construímos na análise e desenho de um sistema."

3. ANÁLISE DO SISTEMA

Casos de uso são utilizados para visualizar o sistema formando espécie de escopo do projeto. Eles definem os requisitos para que o sistema de computador possa ser entendido por usuário e desenvolvedores. Possibilitando uma prévia de como será o sistema, após sua implantação, visualizando os seus objetos e modo como eles interagem entre si.

São compostos basicamente por quatro partes:

- Cenário: Descreve a sequencia dos eventos, ocorrem conforme o usuário interage com o sistema.
- Ator: A **Figura 1** abaixo ilustra o usuário do sistema, quem ira interagir com o mesmo, representado pelo boneco, dependendo do sistema pode ter mais de um ator.

Segundo M. Tepfenhart (2001, p.508):

> Um caso de uso individual pode também ser visto como um procedimento por meio do qual um ator externo pode utilizar o sistema. Tomados em conjunto, os casos de uso definem a funcionalidade plena do sistema [...] podem ser utilizados como base para desenvolvimento de testes do sistema.
> Os dois conceitos [...] diagrama de caso de uso são: Ator: Representa usuários de sistema, incluindo humanos e outros sistemas. Caso de uso: Representa serviços ou a funcionalidade provida pelo sistema aos usuários.

Figura 1- Representa o usuário do sistema.

- Use Case: A **figura 2** trata-se de uma tarefa ou funcionalidade realizada pelo usuário representa interação entre usuário e sistema descrita pelo balão, dentro é descrito o requisito ou caso de uso.

Figura 2 - Balão descreve o requisito como exemplo é “Cadastrar Cliente”.

]

- Comunicação: **Figura 3** liga o ator com o caso de uso, havendo uma interação entre os dois.

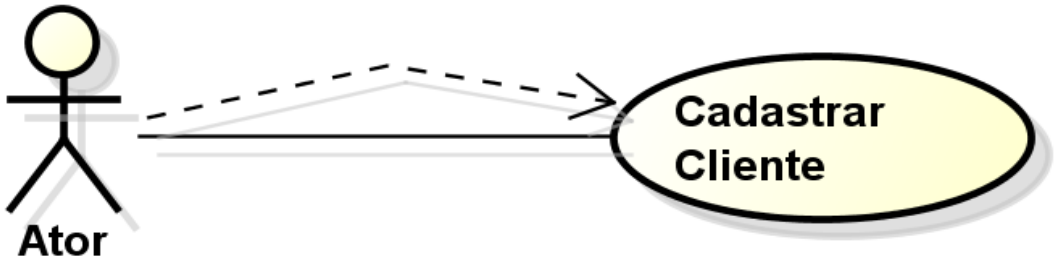

Figura 3 - Representação interação entre usuário e caso de uso.

3.1. DIAGRAMA DE CASO DE USO GERAL

Como já mencionado no inicio deste tópico, os casos de uso descrevem as funcionalidades ou requisitos dos sistemas, ou seja, todas as tarefas que serão realizadas por ele, porém os casos de uso do inglês UC(Use Case), não especificam como fazer e sim o que será feito.

O diagrama de caso de uso geral abaixo ilustrado pela **Figura 4** reúne todos os casos de uso, em uma só figura, ele aborda todas as funcionalidades serão realizadas pelo sistema computacional, juntamente traz (o) ou (os) atores do sistema.

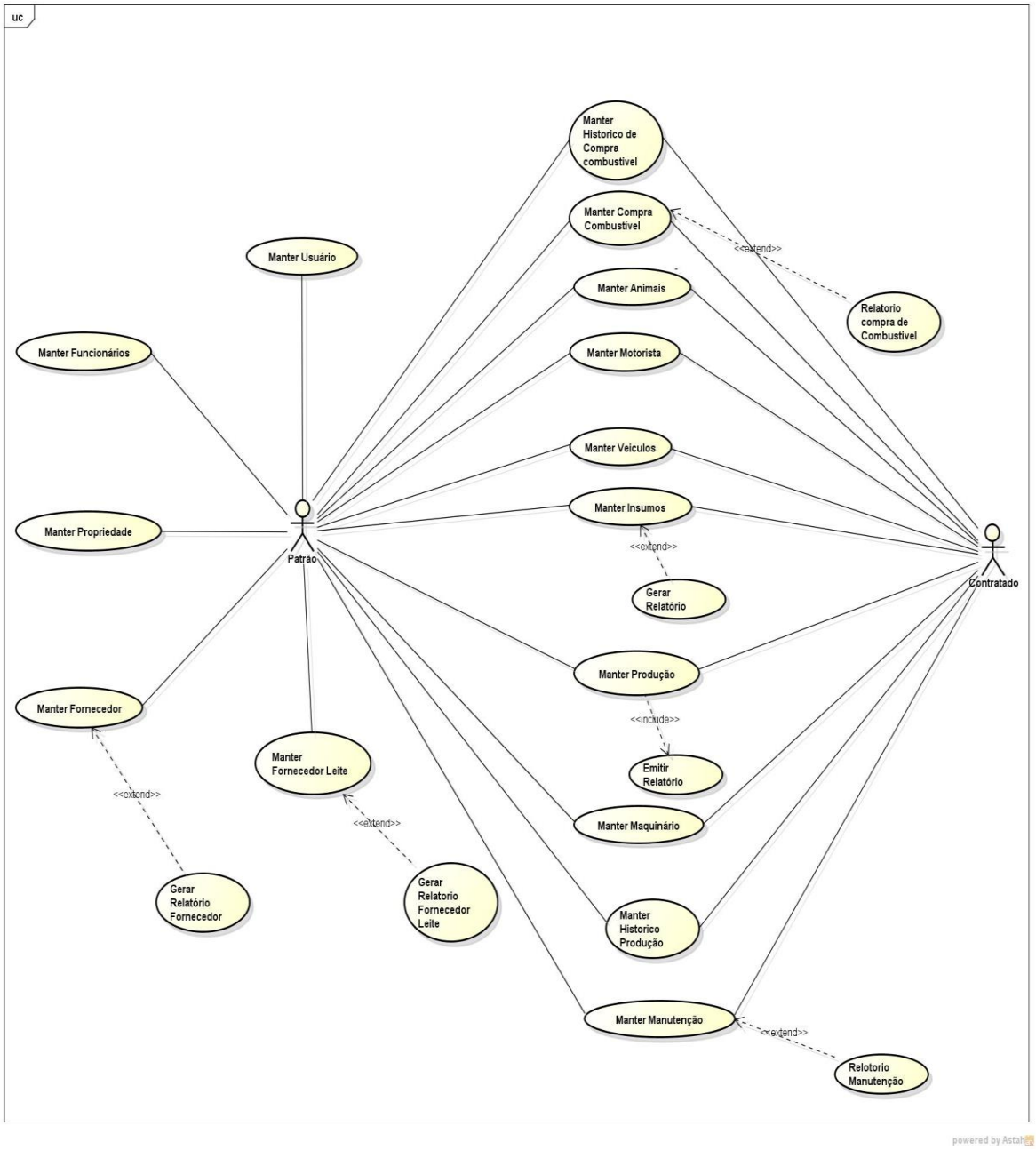

Figura 4 - Mostra o caso de uso geral do sistema.

3.2. NARRATIVAS DE CASO DE USO

Figura 5 demonstra o cadastro de usuários do sistema, de forma controlar o acesso ao sistema.

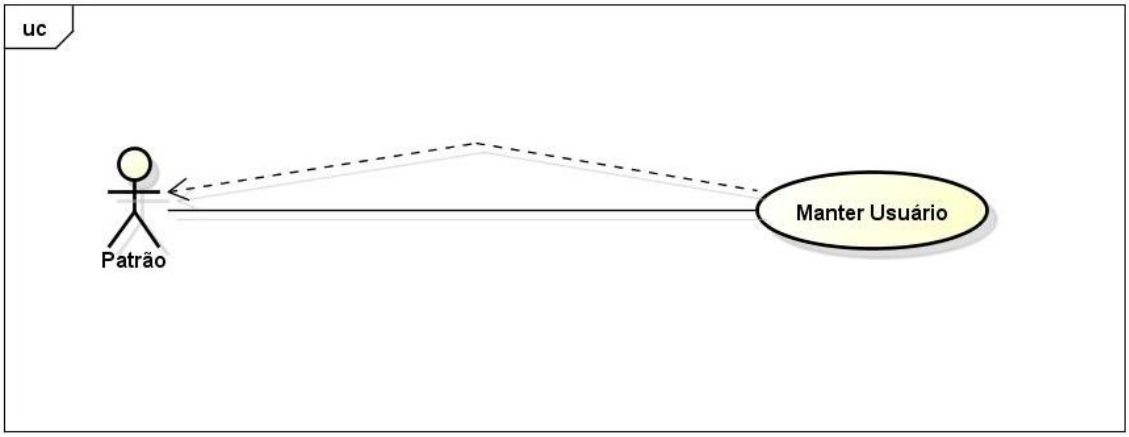

Figura 5- Mostra o caso de uso – Cadastro de usuário do sistema.

1. **Finalidade/Objetivo**
   - Permite o usuário final poder cadastrar novos usuários do sistema.
2. **Ator**
   - Patrão ou dono da respectiva propriedade**.**
3. **Precondições**
   - Ter efetuado login no sistema.
4. **Fluxo Principal**

   A1. Usuário executa sistema, preenche os campos login e senha.

   A2. Em seguida seleciona a opção cadastrar usuário.
   - Sistema salva as informações e continua execução.
5. **Fluxo Alternativo**

   A1. Usuário informa nome e senha, solicita a opção cadastrar novo usuário.
   - Sistema então salva as informações no banco de dados, para acessa – lá quando for preciso.

   A2. Usuário clica na opção Fechar .

- Execução do sistema é interrompida.

6. **Casos de Teste**
    - Verificar a busca de senha e login no banco de dados está correto.

Abaixo na **Figura 6**, são inseridos no sistema os animais que compõem o rebanho do laticínio, através estes dados resultam em maior controle da produção.

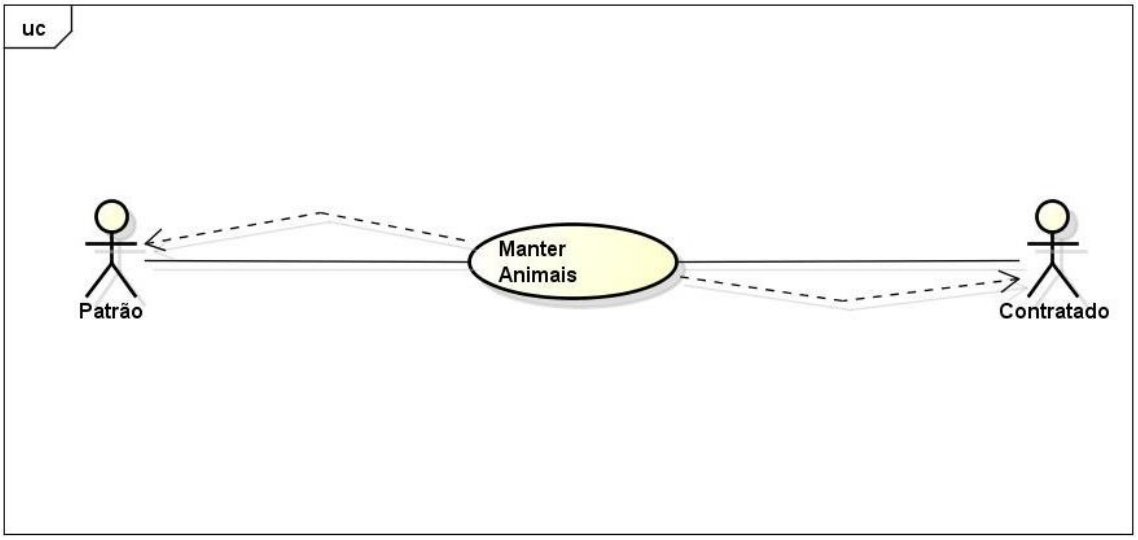

Figura 6– Mostra o caso de uso – Cadastro dos animais/rebanho.

1. **Finalidade/Objetivo**
    - Registrar no sistema todos os animais que compõe o rebanho da propriedade.
2. **Ator**
    - Patrão.
    - Contratado.
3. **Precondições**
    - Ter efetuado o login no sistema.
4. **Fluxo Principal**

    [A1] Usuário seleciona no menu de funções, a opção Rebanho.

    [A2] Em seguida clica sobre Inserir.

    - O sistema solicita os dados precisos para o cadastro de animais.

[A3] Usuário fornece as informações solicitadas.

- Sistema verifica se todas as informações foram preenchidas, em seguida exibe uma confirmação dos dados.

5. **Fluxo Alternativo**

A1. Usuário clica sobre Excluir.

- Sistema solicita o código do animal a ser excluído.

A2. Usuário escolhe opção Alterar Animal.

- Sistema solicita o código a ser alterado

A3. Usuário seleciona a opção Fechar.

- Sistema retorna ao **[A1]** do Fluxo Principal.

6. **Casos de Teste.**

- Verificação dos campos preenchidos.

Figura 7 controla os abastecimentos de veículos, maquinários que são realizados utilizando o combustível estão armazenados na respectiva propriedade.

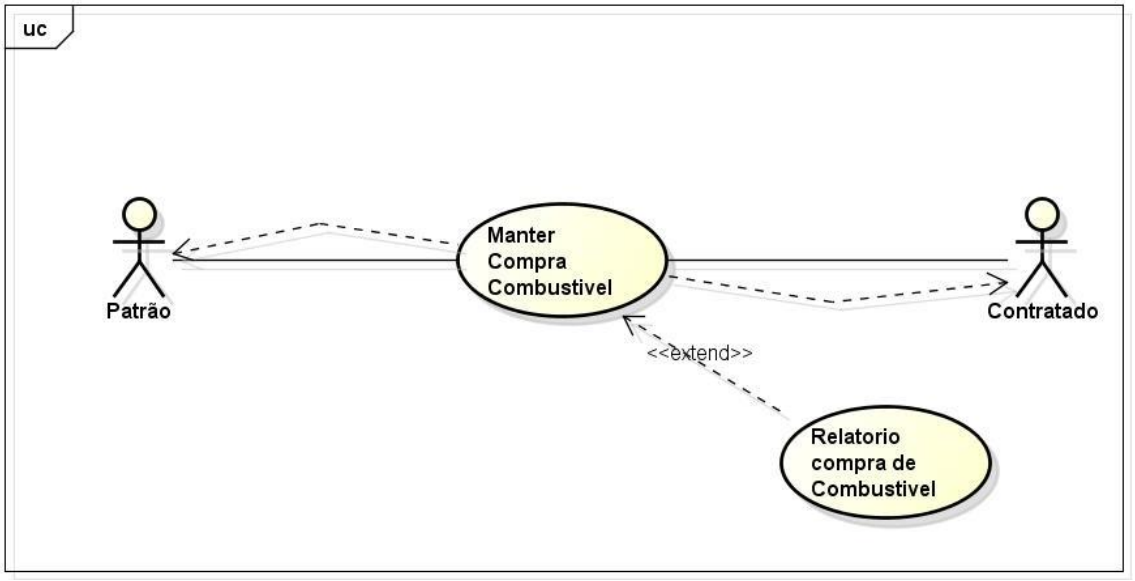

Figura 7– Mostra o caso de uso – Compra de combustível.

1. **Finalidade/Objetivo**

- Obter registro das compras de combustível.

2. **Ator**
    - Patrão.
    - Contratado.
3. **Precondições**
    - Ter efetuado o login no sistema.
4. **Fluxo Principal**

    [A1] Usuário escolhe no menu principal, opção compra combustível.

    [A2] Em seguida clica em Inserir.
    - Sistema solicita todas as informações, para efetuar o registro.

    [A3] Usuário fornece todas as informações.
    - Sistema verifica todos os campos foram informados, em seguida mostra mensagem de confirmação.
5. **Fluxo Alternativo**

    A1. Usuário clica sobre Excluir.
    - Sistema solicita o código da compra a ser excluído.

    A2. Usuário escolhe opção Alterar.
    - Sistema solicita o código a ser alterado.

    A3. Usuário opta pela opção de gerar relatório de aquisição de combustível.
    - Sistema solicita os dados necessários, ao usuário.

    A4. Usuário fornece os dados e confirma a operação.
    - Sistema exibe relatório na tela para usuário.

    A5. Usuário seleciona opção Fechar.
    - Sistema retorna ao **[A1]** do Fluxo Principal.
6. **Casos de Teste.**
    - Verificar os campos preenchidos

Figura 8 abaixo tem por objetivo controlar as empresas fornecem insumos agropecuários, que são todos os produtos essenciais utilizados nas atividades rurais desde adubos, plantas, tratores, agrotóxicos, etc.

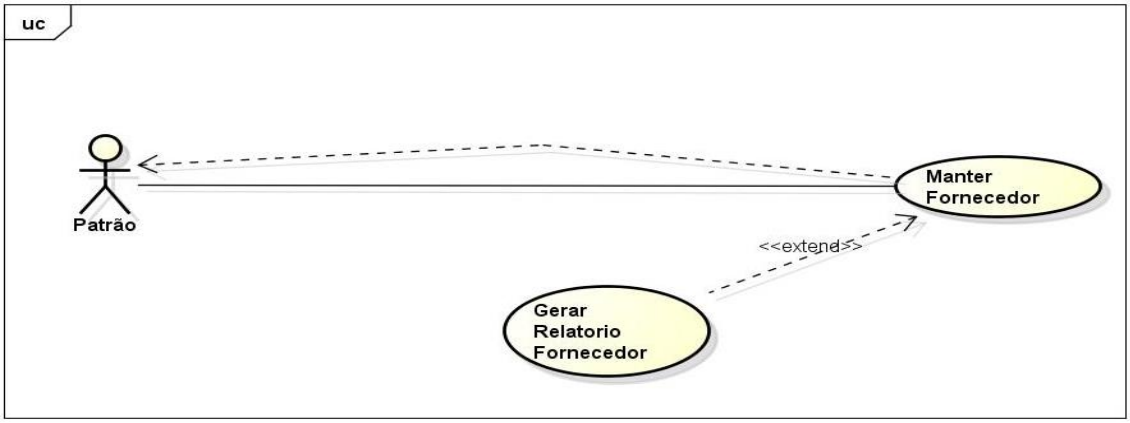

Figura 8– Mostra o caso de uso – Manter Fornecedor.

1. **Finalidade/Objetivo**
   - Registrar quem e quais os fornecedores vendem produtos para o laticínio.
2. **Ator**
   - Patrão.
   - Contratado.
3. **Precondições**
   - Ter efetuado o login no sistema.
4. **Fluxo Principal**

   [A1] Usuário escolhe na tela principal, opção Fornecedor Produtos.

   [A2] Em seguida clica na opção Inserir.
   - Sistema solicita os dados, para efetuar o registro.

   [A3] Usuário fornece todas as informações.
   - Sistema verifica todos os campos se foram informados, em seguida mostra uma mensagem de confirmação.
5. **Fluxo Alternativo**

   A1. Usuário clica sobre a opção Excluir, fornecedor de produtos.
   - Sistema então solicita o código do registro a ser excluído.

   A2. Usuário escolhe a opção Alterar.
   - Sistema solicita o código a ser alterado.

   A3. Usuário opta pela função do sistema, gerar relatório dos fornecedores.
   - Sistema pede ao usuário às informações necessárias.

A4. Usuário informa os dados foram pedidos e confirma a operação **A3** do Fluxo Alternativo.

- Sistema exibe na tela o relatório.

A5. Usuário opta pela opção Fechar.

- Sistema retorna ao **[A1]** do Fluxo Principal.

6. **Casos de Teste**
    - Verificar os campos preenchidos.

Figura 9 abaixo mostra o relatório que posteriormente será gerado, nele conterá os dados de quem esta vendendo leite ao laticínio dados esses são código, nome, quantidade de litros, data referente a venda, situação do pagamento.

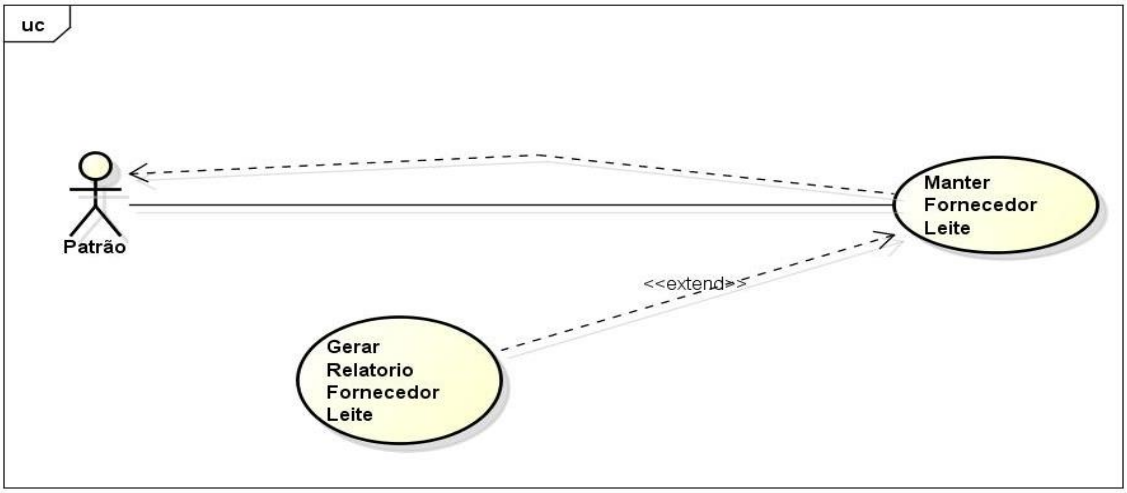

Figura 9– Mostra o caso de uso – Manter fornecedor de Leite.

1. **Finalidade/Objetivo**
    - Registrar no sistema quem e quais as propriedades rurais vendem leite para o lacticínio.
2. **Ator**
    - Patrão.

3. **Precondições**
    - Ter efetuado o login no sistema.
4. **Fluxo Principal**

    [A1] Usuário escolhe no menu de funções a, opção Manter Fornecedor.

    [A2] Em seguida clica em Inserir.
    - Sistema solicita todas as informações, para efetuar o registro fornecedor de leite.

    [A3] Usuário fornece todas as informações.
    - Sistema verifica se todos os campos foram preenchidos, exibe mensagem de confirmação.
5. **Fluxo Alternativo**

    A1. Usuário escolhe a opção Excluir.
    - Sistema solicita o código do fornecedor a ser apagado.

    A2. Usuário escolhe opção Alterar.
    - Sistema pede o código a ser alterado.

    A3. Usuário opta a opção gerar relatório fornecedor leite.
    - Sistema solicita os dados necessários.

    A4. Usuário preenche os dados, foram solicitados pelo sistema e confirma a operação **A3** do Fluxo Alternativo.

    A5. Usuário seleciona opção Fechar.
    - Sistema então retorna, ao **[A1]** do Fluxo Principal.
6. **Casos de Teste**
    - Verificar os campos preenchidos.

Figura 10 abaixo ilustra estoque, através de baixas que são realizadas, conforme serão sendo utilizados.

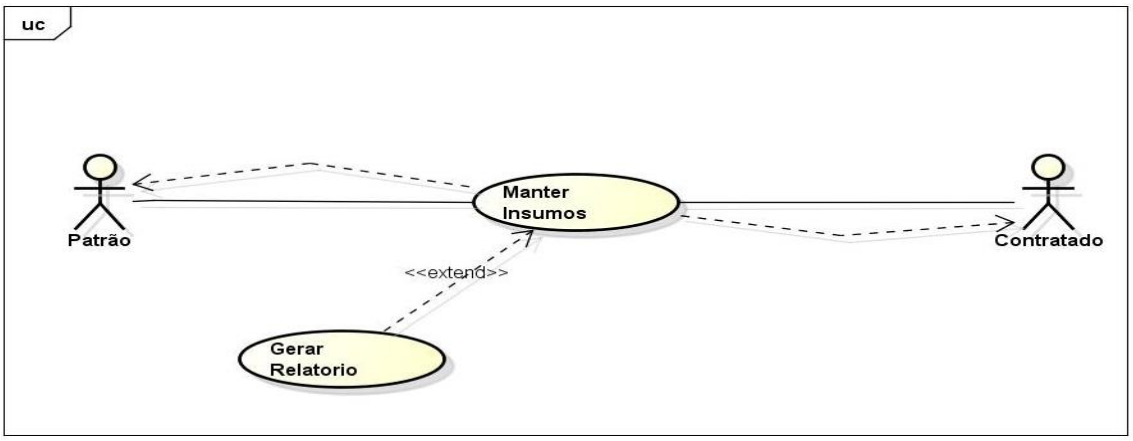

Figura 10– Mostra o caso de uso – Registro de Insumos.

1. **Finalidade/Objetivo**
   - Ter controle sobre o estoque registrando as entradas e saídas, de insumos.
2. **Ator**
   - Patrão.
   - Contratado.
3. **Precondições**
   - Ter efetuado login no sistema.
4. **Fluxo Principal**

   [A1] Usuário escolhe no menu de funções, opção Insumos.

   [A2] Em seguida clica em cadastrar.

   - Sistema solicita as informações para registrar insumos.

   [A3] Usuário fornece as informações precisas.

   - Sistema verifica se todos os campos forma preenchidos, exibe uma mensagem de confirmação.
5. **Fluxo Alternativo**

   A1. Usuário seleciona opção Excluir.

   - Sistema solicita o código do Insumo a ser exclui-lo.

   A2. Usuário opta pela opção Alterar o registro do insumo.

   - Sistema solicita o código a ser alterado.

   A3. Usuário opta pela opção de gerar relatório, do estoque de insumos.

   - Sistema solicita as informações necessárias.

A4. Usuário informa os dados, foram solicitados e confirma a operação **A3** do Fluxo Alternativo.

A5. Usuário opta pela opção Fechar.

- Sistema retorna ao **[A1]** do Fluxo Principal.

6. **Casos de Teste**
   - Verificar os campos preenchidos.

Figura 11 logo abaixo controla as manutenções realizadas nos maquinários e veículos, ou seja, as trocas de peças, ou manutenção de rotina, descrevendo os itens e valor.

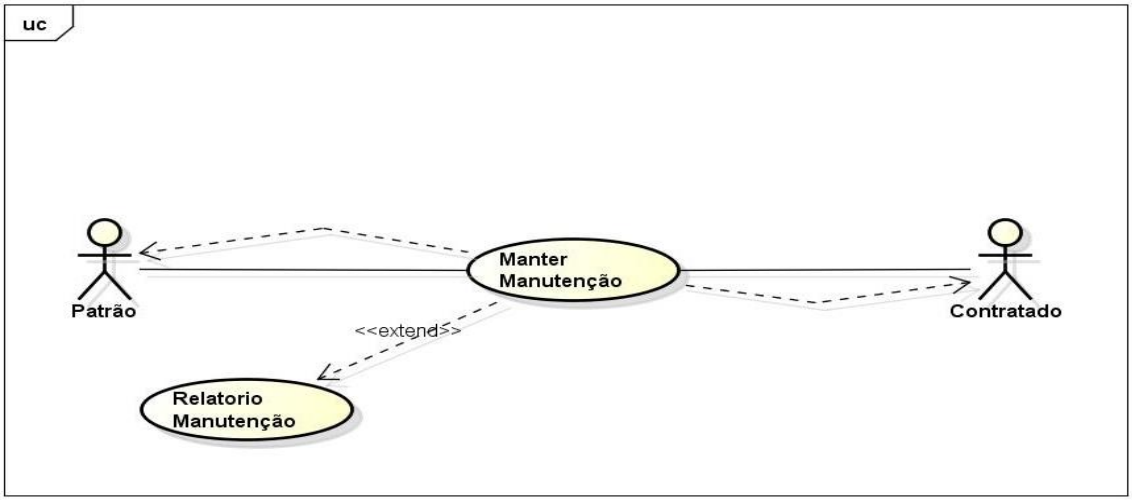

Figura 11- Mostra o caso de uso – Manter Manutenção.

1. **Finalidade/Objetivo**
   - Registrar as manutenções, realizadas nos maquinários da propriedade.
2. **Ator**
   - Patrão.
   - Contratado.
3. **Precondições**
   - Ter efetuado login no sistema.
4. **Fluxo Principal**

[A1] Usuário no menu principal opta, pela opção Manutenções.

[A2] Em seguida escolhe a opção Inserir.

- Sistema solicita os dados, para realizar o registro de manutenção.

[A3] Usuário informa as informações necessárias.

- Sistema verifica se os campos foram preenchidos, em seguida exibe uma mensagem de confirmação.

5. **Fluxo Alternativo**

A1. Usuário opta pela opção Excluir o registro de manutenção.

- Sistema solicita o código de respectivo registro a ser excluído.

A2. Usuário escolhe opção Editar o registro de manutenção.

- Sistema pede o código do registro a ser excluído.

A3. Usuário clica na opção Fechar.

- Sistema retorna ao **[A1]** do Fluxo Principal.

6. **Casos de Teste**

- Verificar os campos preenchidos.

Figura 12 com este registro o dono da propriedade tem o controle de todos os equipamentos dispostos na sua propriedade, juntamente com as manutenções pode apurar os gastos com determinado maquinário.

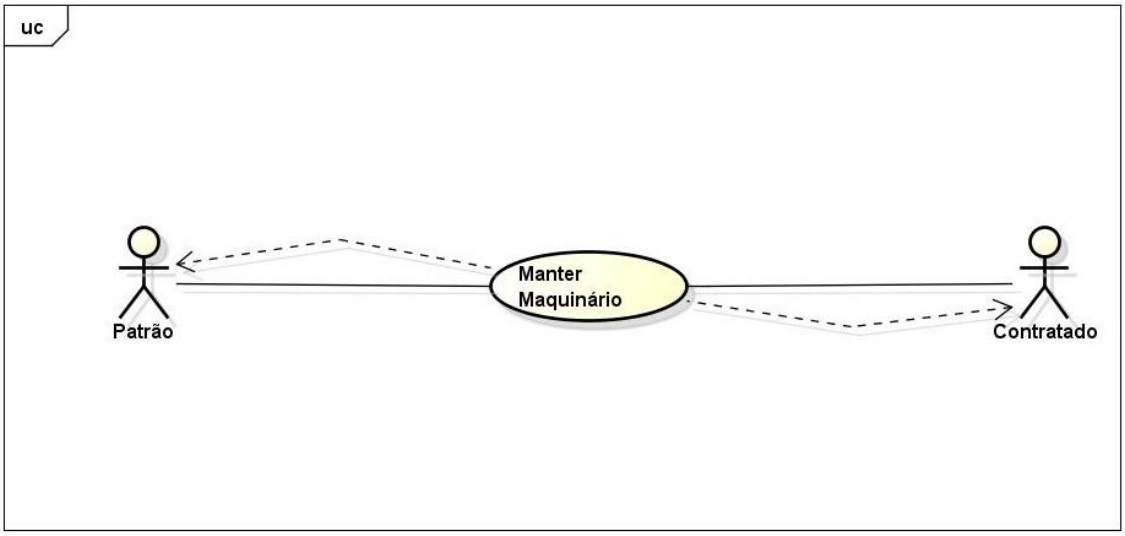

Figura 12 - Mostra o caso de uso – Manter Maquinário.

1. **Finalidade/Objetivo**

- Registrar as maquina agrícolas da propriedade, que já pertenciam ou foram adquiridos.

2. **Ator**

- Patrão
- Contratado

3. **Precondições**

- Ter efetuado login no sistema.

4. **Fluxo Principal**

[A1] Usuário escolha no menu principal do sistema, opção Maquinários.

[A2] Usuário em seguida clica na opção Inserir.

- Sistema, então solicita os dados para efetuar o registro.

[A3] Usuário informa os dados necessários para operação.

- Sistema verifica se os campos do registro foram preenchidos, em seguida exibe uma mensagem de confirmação.

5. **Fluxo Alternativo**

A1. Usuário opta pela opção Excluir.

- Sistema solicita o código para exclusão do registro de maquinário.

A2. Usuário seleciona opção Alterar.

- Sistema então pede ao usuário informe o código do registro que será exclui-lo.

A3. Usuário seleciona a opção Fechar.

- Sistema retorna ao **[A1]** do Fluxo Principal.

6. **Casos de Teste**

Verificar os campos preenchidos.

Figura 13 abaixo possibilita saber qual o funcionário, que esta responsável pelo veiculo utilizado nas coletas de produções das demais propriedades que vendem a sua produção de leite ao laticínio, no caso estas propriedades são os fornecedores de leite.

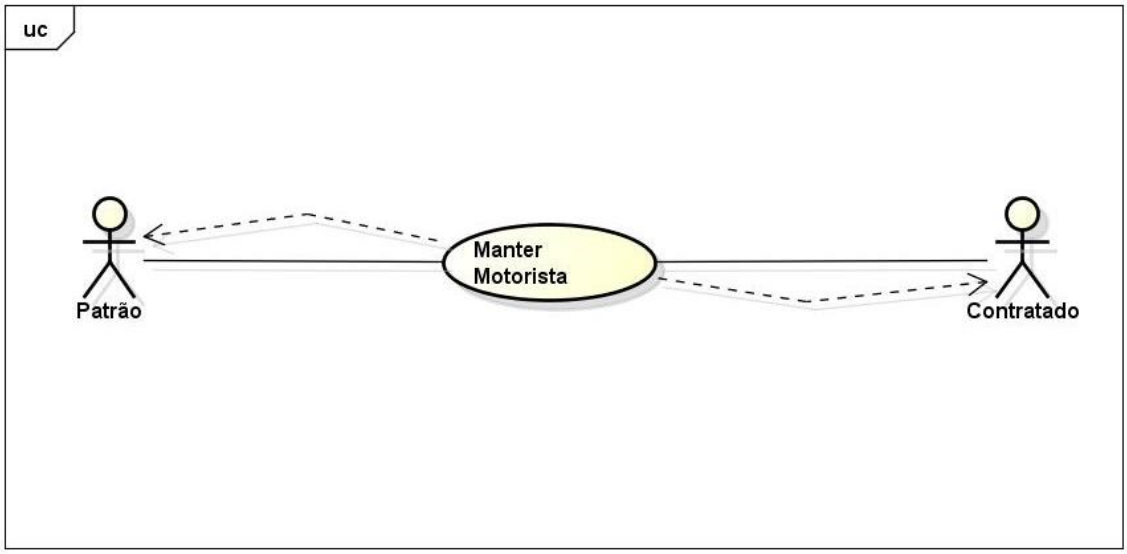

Figura 13– Mostra o caso de uso – Manter Motorista.

1. **Finalidade/Objetivo**
   - Ter controle saber qual é o veiculo, motorista e local destino.
2. **Ator**
   - Patrão.
   - Contratado.
3. **Precondições**
   - Ter feito o login no sistema.
4. **Fluxo Principal**

   [A1] Usuário no menu principal do sistema, seleciona a opção Motorista.

   [A2] Em seguida clica sobre a opção Inserir.
   - Sistema solicita todos os dados relacionados ao registro de motorista.

   [A3] Usuário informa todos os campos de dados.
   - Sistema exibe uma mensagem de confirmação.
5. **Fluxo Alternativo**

   A1. Usuário opta pela opção, Excluir.
   - Sistema solicita o código do registro para exclui - lo.

   A2. Usuário seleciona opção Alterar.
   - Sistema solicitado o código do registro para ser alterado.

A3. Usuário escolhe opção Fechar.

- Sistema retorna ao, **[A1]** do Fluxo Principal.

6. **Casos de Teste**
    - Verificar os campos preenchidos.

Figura 15 abaixo ilustra as produções de leite no laticínio, permitindo o controle sobre leite que é produzido, além disso, calcula a media de produção por cada animal.

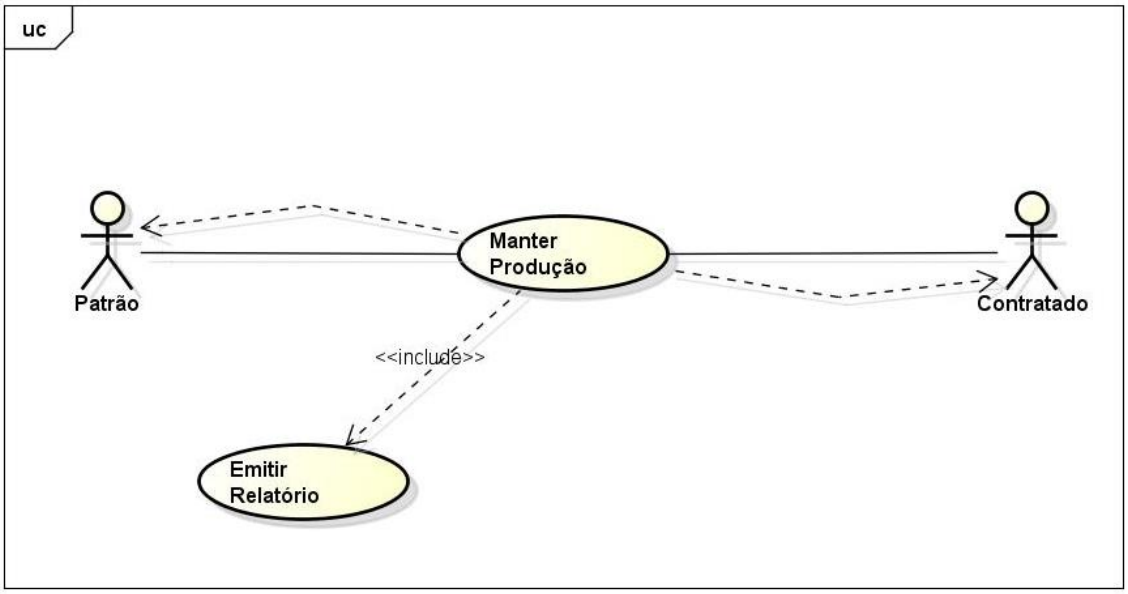

Figura 14 – Mostra o caso de uso – Manter Produção.

1. **Finalidade/Objetivo**
    - Registrar quantidade de leite produzido.
2. **Ator**
    - Patrão.
    - Contratado.
3. **Precondições**
    - Ter efetuado login no sistema.
4. **Fluxo Principal**

[A1] Usuário escolhe na tela de menu a opção Produção Leite.

[A2] Em seguida o usuário clica sobre Inserir.

- Sistema então solicita os dados relativos produção.

[A3] Usuário informa todos os campos solicitados.

- Sistema exibe uma mensagem de confirmação, para inserir os dados.

5. Fluxo Alternativo

A1. Usuário opta pela opção Excluir.

- Sistema pede para informar o código registro para exclui – lo.

A2. Usuário opta pela opção Alterar.

- Sistema pede o código a ser alterado.

A3. Usuário opta pela opção de gerar relatório, da .

- Sistema solicita as informações necessárias.

A4. Usuário informa os dados, foram solicitados e confirma a operação **A3** do Fluxo Alternativo.

A3. Usuário opta pela opção Fechar.

- Sistema retorna ao, **[A1]** do Fluxo Principal.

6. Casos de Teste

- Verificar os campos preenchidos

Figura 16 logo abaixo associa ao fornecedor de leite a sua propriedade, pois muitas vezes o dono da propriedade não reside na mesma.

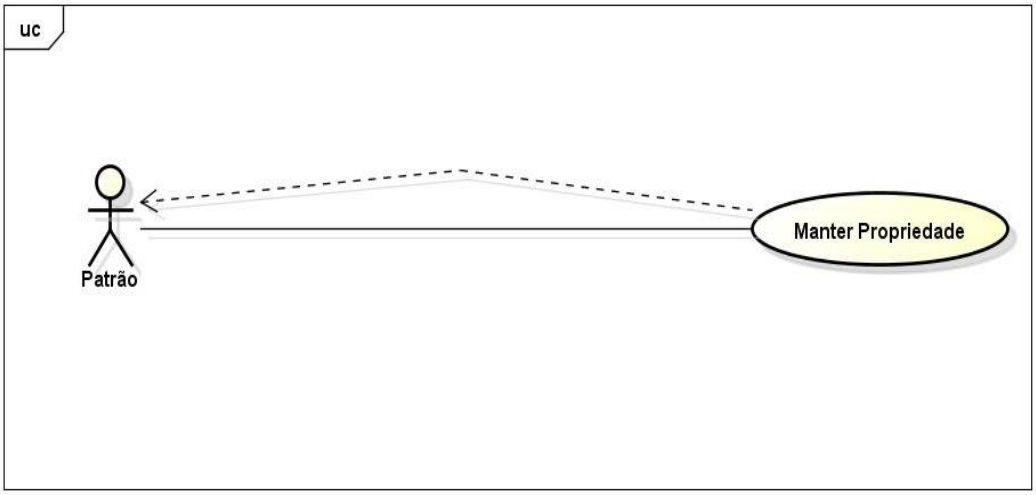

Figura 15– Mostra o caso de uso – Manter Propriedade.

1. **Finalidade/Objetivo**
    - Possuir controle de uma e/ou varias propriedades distintas.
2. **Ator**
    - Patrão.
    - Contratado.
3. **Precondições**

    Ter feito o login no sistema.
4. **Fluxo Principal**

    [A1] Usuário escolhe na tela principal opção Propriedade.

    [A2] Usuário em seguida seleciona opção Inserir.
    - Sistema solicita os dados precisos para efetuar registro no sistema.

    [A3] Usuário informa todos os campos necessários.
    - Sistema então exibe uma mensagem de confirmação.
5. **Fluxo Alternativo**

    A1. Usuário opta pela opção Excluir.
    - Sistema solicita a ele o código a ser excluído.

    A2. Usuário escolhe então Alterar.
    - Sistema então solicita o código do registro a ser excluído.

    A3. Usuário solicita ao sistema operação Fechar.
    - Sistema retorna ao **[A1],** do Fluxo Principal.
6. **Casos de Teste**
    - Verificar os campos preenchidos.

Figura 17 logo abaixo possibilita saber quantos e quais veículos fazem parte da frota, somada aos registros de manutenções possibilita maior controle de gastos.

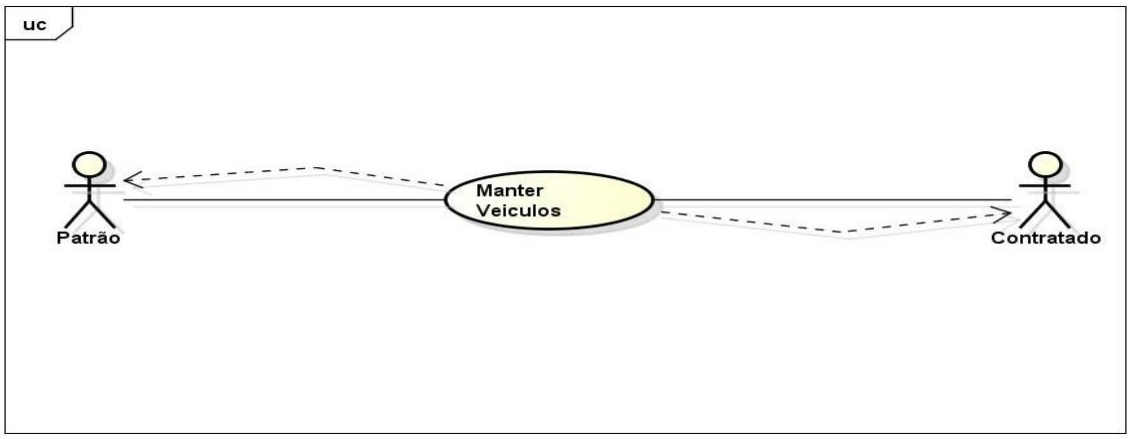

Figura 16– Mostra o caso de uso – Manter Veículos.

1. **Finalidade/Objetivo**
   - Obter informações sobre os veículos utilizados no cotidiano da empresa.
2. **Ator**
   - Patrão
   - Contratado.
3. **Precondições**
   - Ter efetuado login no sistema.
4. **Fluxo Principal**

   [A1] Usuário no menu principal clica sobre a opção Veículos.

   [A2] Logo após usuário seleciona opção Inserir.
   - Sistema solicita ao usuário todas as informações.

   [A3] Usuário preenche todos os dados solicitados pelo sistema.
   - Sistema exibe um mensagem de confirmação da operação.
5. **Fluxo Alternativo**

   A1. Usuário opta pela operação Excluir.
   - Sistema pede ao usuário informar código de exclusão.

   A2. Usuário escolhe opção Alterar.
   - Sistema então solicita o código de registro para exclui-lo.

   A3. Usuário escolhe a opção Fechar.
   - Sistema então retorna ao **[A1]** do Fluxo Principal.
6. **Casos de Teste**
   - Verificar os campos preenchidos.

Figura 18 abaixo possibilita saber o quadro de funcionários com isto o proprietário pode ter os registros eletrônicos de seus colaboradores de sua propriedade.

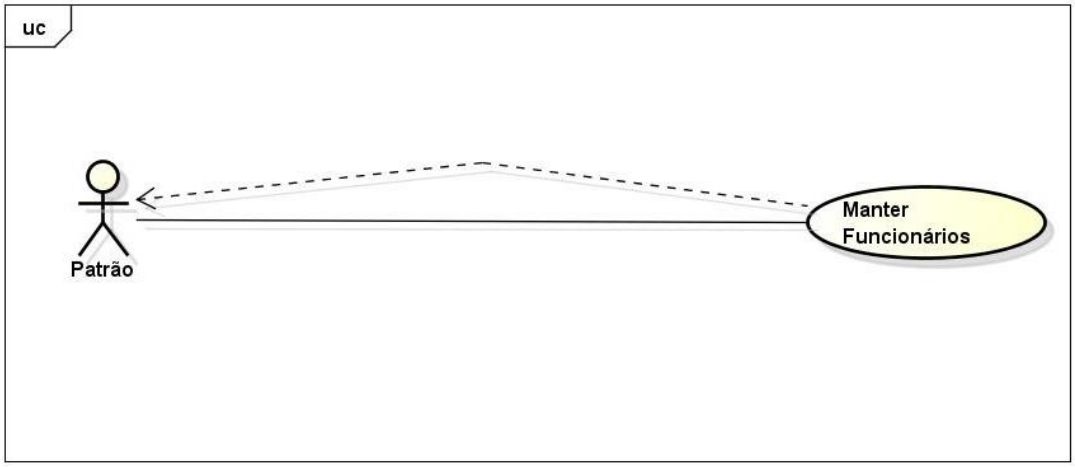

Figura 17– Mostra o caso de uso – Manter Funcionários.

1. **Finalidade/Objetivo**

    - Saber seu quadro de funcionários que estão trabalhando na propriedade.

2. **Ator**

    - Patrão.

3. **Precondições**

    - Ter efetuado login no sistema.

4. **Fluxo Principal**

    [A1] Usuário na tela de menu escolhe a opção Funcionários.

    [A2] Em seguida usuário clica na opção Inserir.

    - Sistema então solicita o preenchimento dos dados.

    [A3] Usuário informa os dados solicitados.

    - Sistema então exibe uma mensagem de confirmação.

5. **Fluxo Alternativo**

A1. Usuário escolhe opção Excluir o registro de funcionário.

- Sistema pede ao usuário informe o código para exclusão.

A2. Usuário escolhe opção Alterar.

- Então sistema solicita o código para realizar alteração.

A3. Usuário clica em Fechar.

- Sistema então retorna ao **[A1]** do Fluxo Principal.

6. **Casos de Teste**

- Verificar os campos preenchidos.

Figura 19 abaixo ambos os usuários podem manter os registros mais antigos para compara-los com os mais recentes.

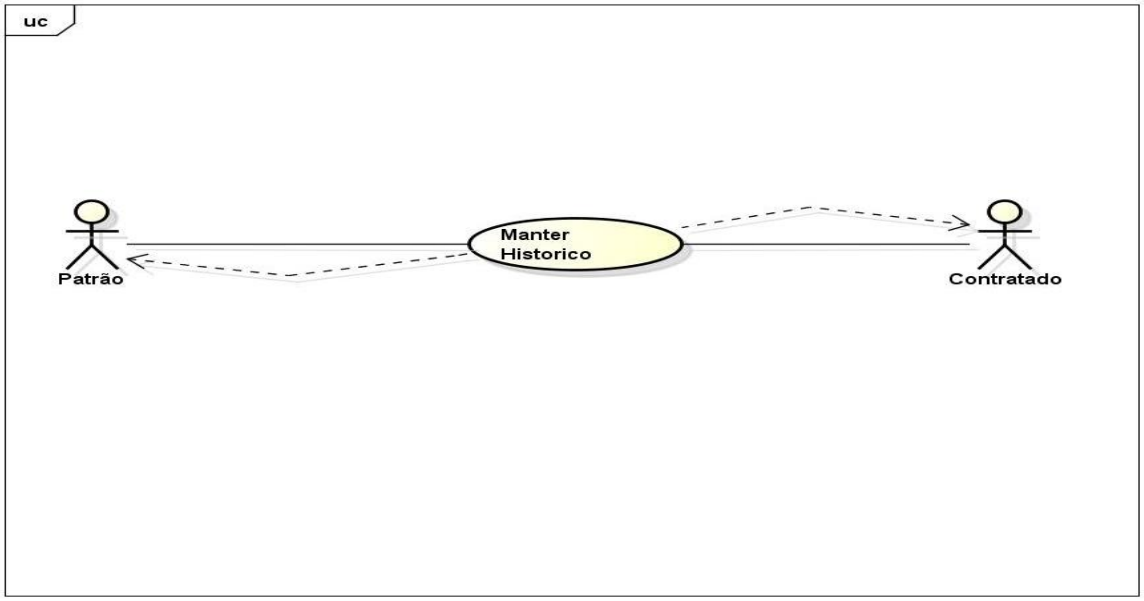

Figura 18– Mostra o caso de uso – Manter Histórico de Compra Combustível.

1. **Finalidade/Objetivo.**

- Obter a relação dos custos, sejam os mais recentes ou de algum certo tempo anterior.

2. **Ator.**
    - Patrão.
    - Contratado.
3. **Precondições.**
    - Ter efetuado login no sistema.
4. **Fluxo Principal.**

    [A1] Usuário na tela de menu escolhe a opção Histórico Compra Combustível..

    [A2] Em seguida usuário clica na opção Inserir.
    - Sistema então solicita o preenchimento dos dados para gerar histórico.

    [A3] Usuário informa os dados solicitados.
    - Sistema então exibe uma mensagem de confirmação.

5. **Fluxo Alternativo**

    A1. Usuário opta por Excluir.
    - Sistema então solicita o respectivo código, histórico a ser apagado.

    A2. Usuário opta pela alteração de histórico, clica sobre Alterar.
6. **Casos de Teste.**
    - Verificar os campos preenchidos.

Figura 19 abaixo possibilita manter o registros realizados a mais tempo, caso necessite compara lós com mais recentes.

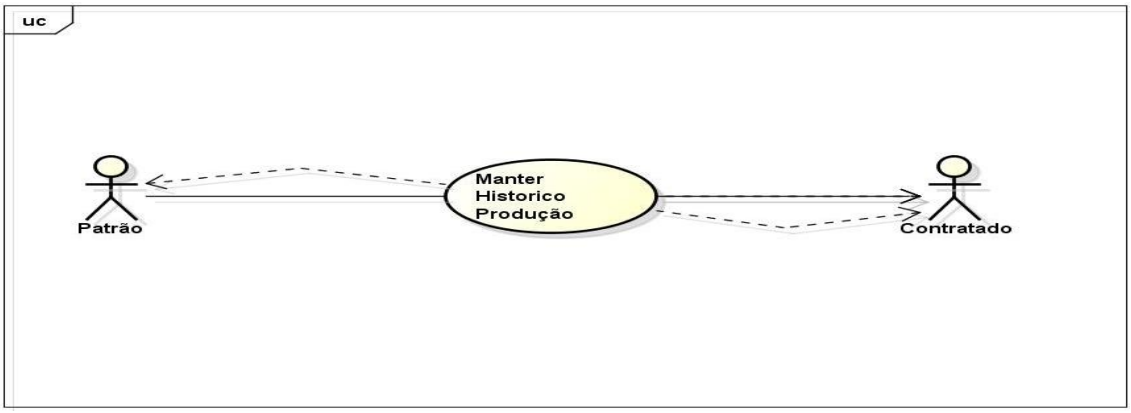

Figura 19– Mostra o caso de uso – Manter Histórico de Produção.

1. **Finalidade/Objetivo.**
    - Saber o quanto já foi produzido em determinado tempo anterior, poder fazer uma comparação.
2. **Ator.**
    - Patrão.
    - Contratado.
3. **Precondições.**
    - Ter efetuado login no sistema.
4. **Fluxo Principal.**

    [A1] Usuário na tela de menu escolhe a opção Histórico de Produção.

    [A2] Em seguida usuário clica na opção Cadastrar.
    - Sistema então solicita o preenchimento dos dados para gerar histórico produção.

    [A3] Usuário informa os dados solicitados.
    - Sistema então exibe uma mensagem de confirmação e usuário confirma.

5. **Fluxo Alternativo**

    A1. Usuário opta por Excluir.
    - Sistema então solicita o respectivo código, histórico a ser apagado.

    A2. Usuário opta pela alteração de histórico produção, clica sobre Alterar.
6. **Casos de Teste.**
    - Verificar os campos preenchidos.

3.3. DIAGRAMAS DE SEQUENCIA

Segundo M. Tepfenhart (2001, p.521):

> Um diagrama de sequencia é um diagrama que mostra objetos reais e interações entre objetos no sentido horizontal, e sequencia no vertical. As linhas pontilhadas verticais representam a linha de vida do objeto e as setas horizontais representam as interações ou mensagem entre objetos [...]. Mensagens podem incluir números de sequencia, nomes de operação e parâmetros reais. Caixas alongadas estreitas nas linhas de vida de um objeto representam sua "ativação" quando interações são sequenciais e representam chamadas de operações [...] permanece ativa até que todas as operações sequenciais que ele invoca sejam finalizadas e devolvidas.

Figura 20 abaixo, temos o ator principal que é o Patrão no caso o proprietário do laticínio, realizando o acesso aos usuários já locados no sistema ou adicionando um novo.

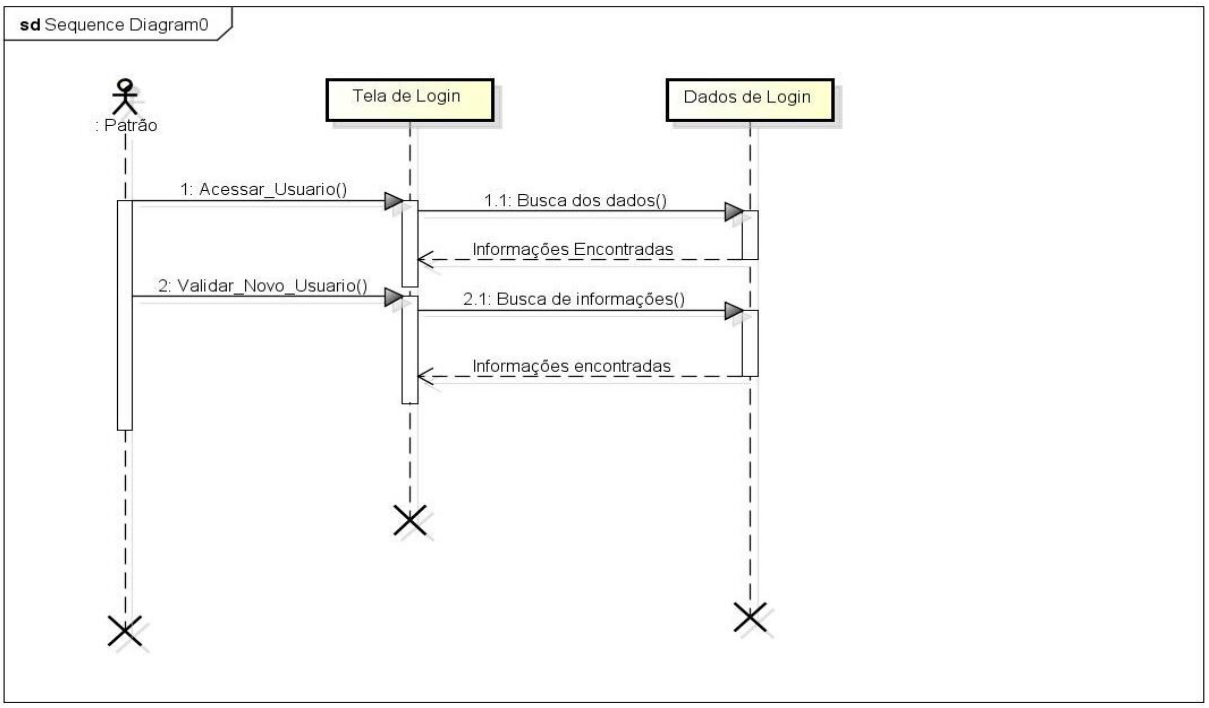

Figura 20 - Diagrama de Sequencia - Login de usuário do sistema

Figura 21 abaixo vemos o usuário denominado de Patrão realizando o login, após isso registrando uma nova propriedade, que ira vender o leite ao laticínio.

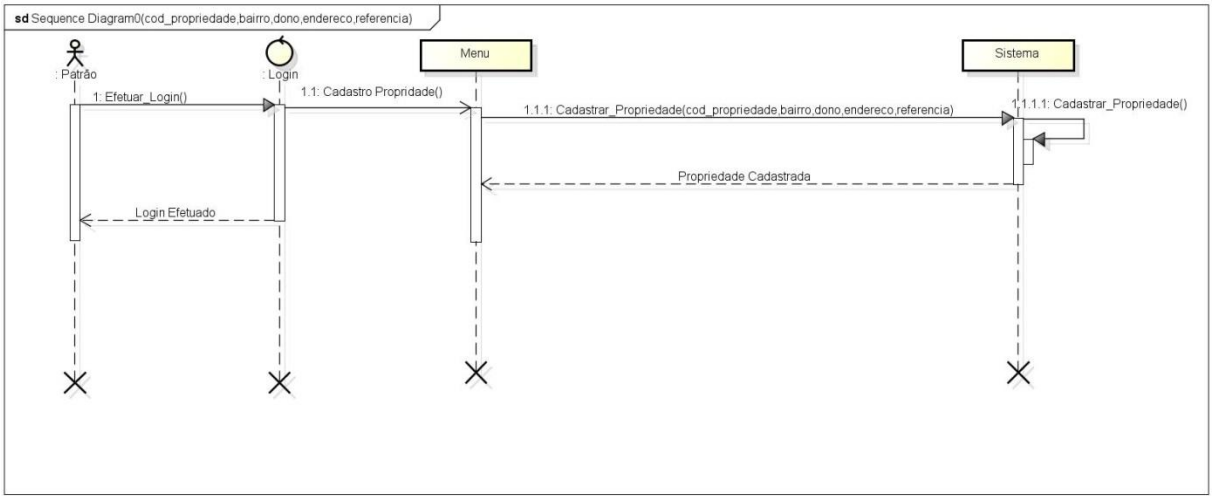

Figura 21 - Diagrama de sequencia - Cadastro de Propriedade.

Figura 22, abaixo veremos o respectivo dono efetua o login, em seguida acessa o menu principal do sistema, registra um novo fornecedor de insumos agrícolas, em seguida tem a opção de gerar relatório.

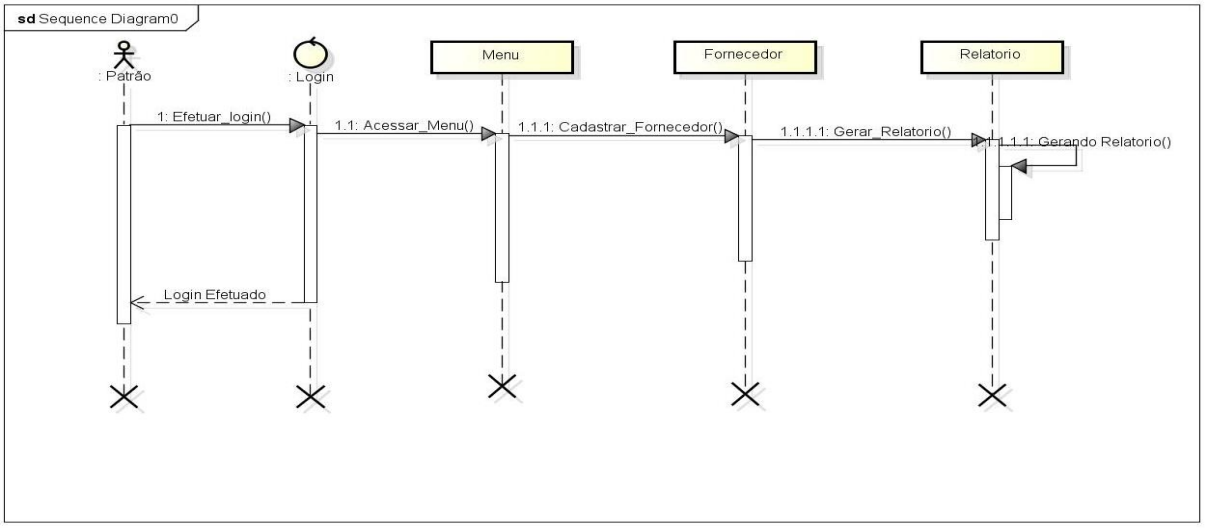

Figura 22 - Diagrama de Sequencia - Registro de Fornecedor.

Figura 23, dono da propriedade após ter efetuado o login, acessa o menu principal, na opção de Fornecedor Leite, pode inserir, editar, consultar e gerar relatório que conterá as informações a respeito das pessoas e suas vendas realizadas para o laticínio.

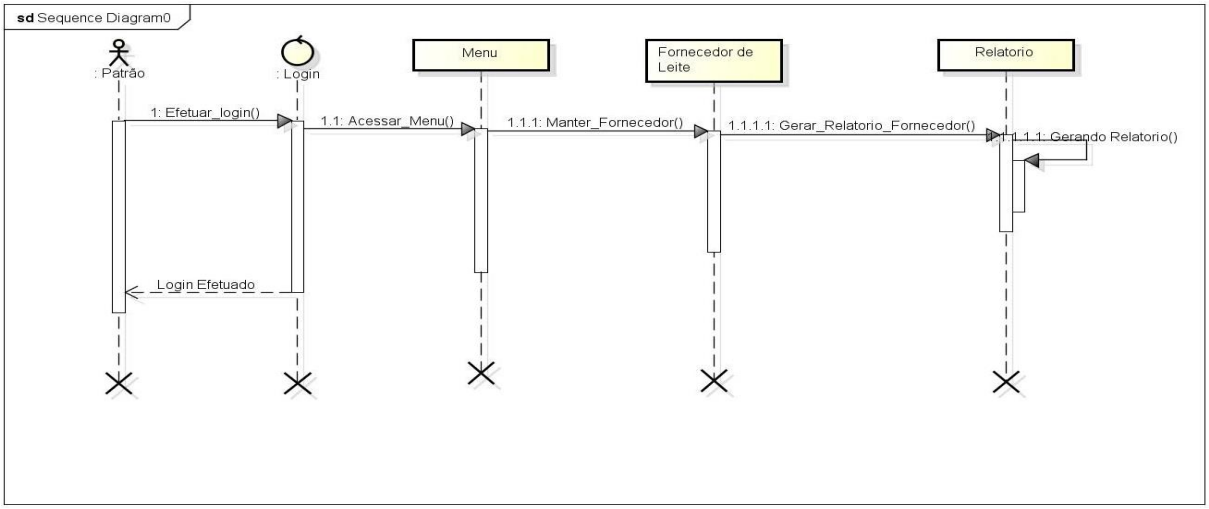

Figura 23 - Diagrama de Sequencia - Cadastro Fornecedor de Leite.

Figura 24 segue logo abaixo tanto o dono como seu funcionário, podem registrar as compras de combustível que são efetuadas, seria quando o laticínio encomenda determinada quantia de combustível diretamente do fornecedor.

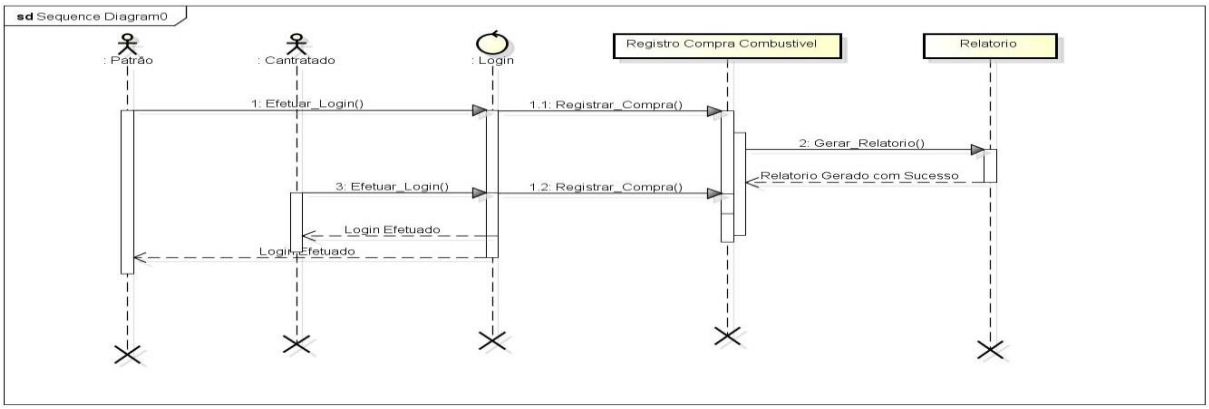

Figura 24 - Diagrama de Sequencia - Registro Compra de Combustível.

Figura 25, localizada abaixo ilustra o dono da propriedade inserindo um novo funcionário recém-contratado, desta além de saber que esta trabalhando, contara com os registros.

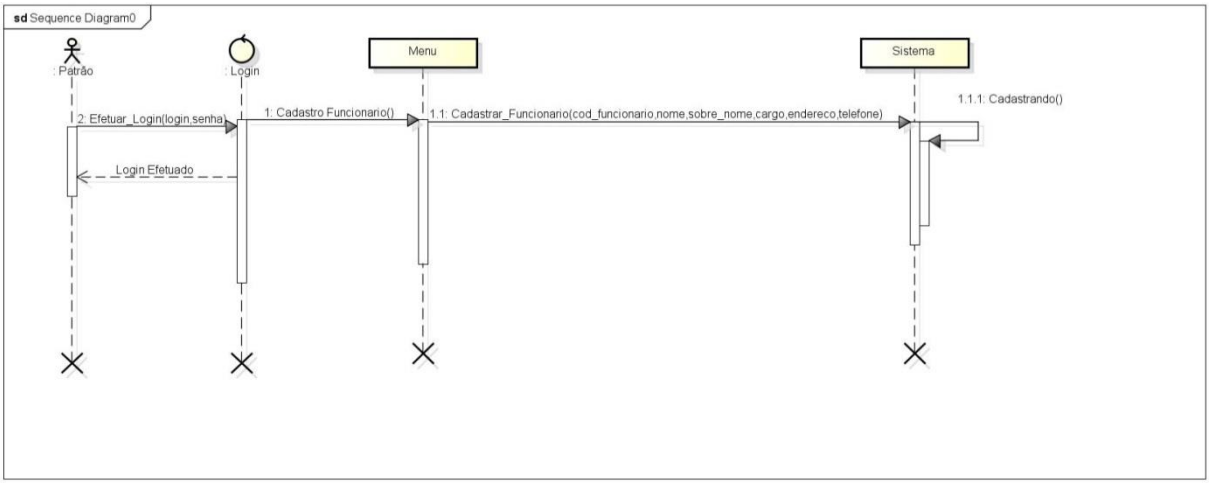

Figura 25- Diagrama de Sequencia - Registra os funcionários.

Figura 26, logo abaixo o dono da propriedade ou funcionário efetuam o login a partir de então registram qual funcionário será motorista responsável por guiar o caminhão na coleta de leite pelas demais propriedades.

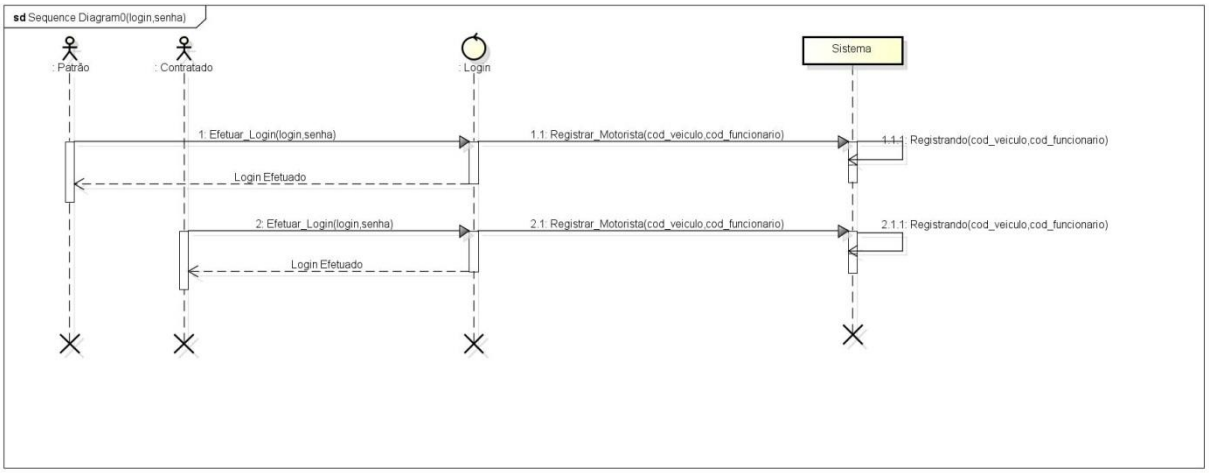

Figura 26 - Diagrama de Sequencia - Manter Motoristas dos veículos de transporte mercadoria.

Figura 27, logo abaixo ilustra o dono ou funcionário, registrando nova produção de leite dentro do laticínio, contendo as informações sobre o determinada ordenharia que acabou de ocorrer.

Figura 27- Diagrama de Sequencia - Registro de Produção

Figura 28, registro de rebanho do laticínio ou de outra propriedade, pode ser realizado por dono ou funcionário, assim terá o conhecimento de onde são os animais que estão produzindo o leite.

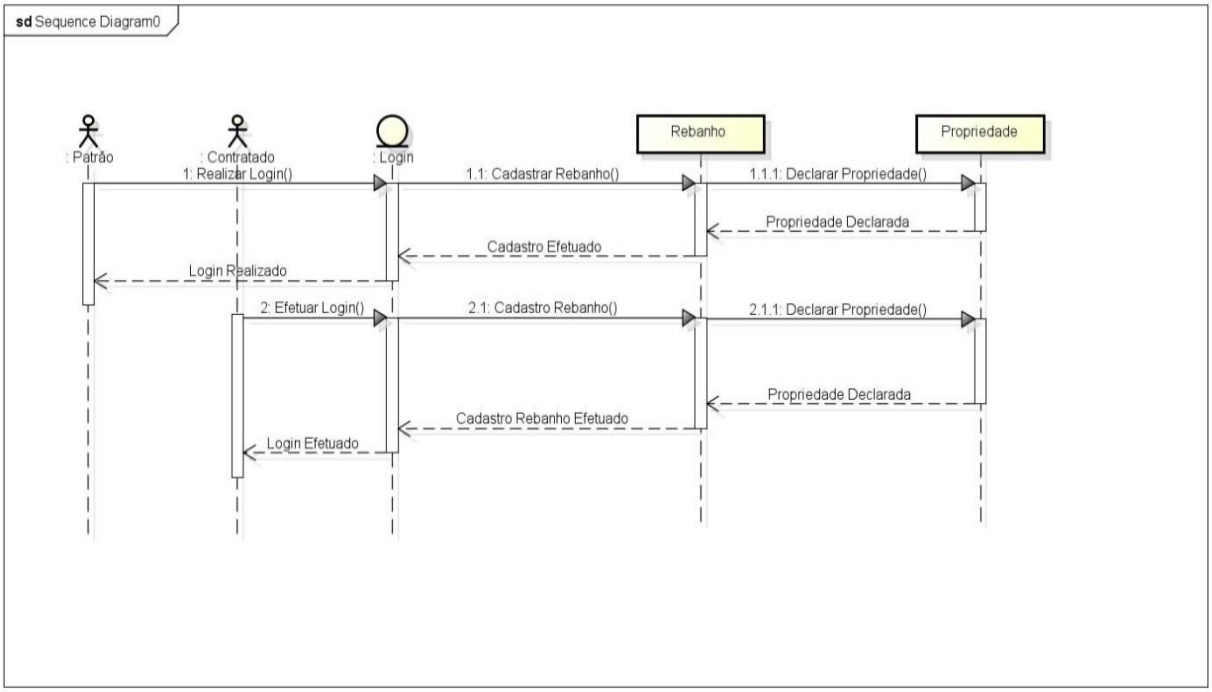

Figura 28 - Diagrama de Sequencia - Controle de Rebanho

Figura 29, localizada abaixo o controle dos insumos que são utilizados na propriedade no desenvolvimento das atividades controlando assim a quantidade disponível.

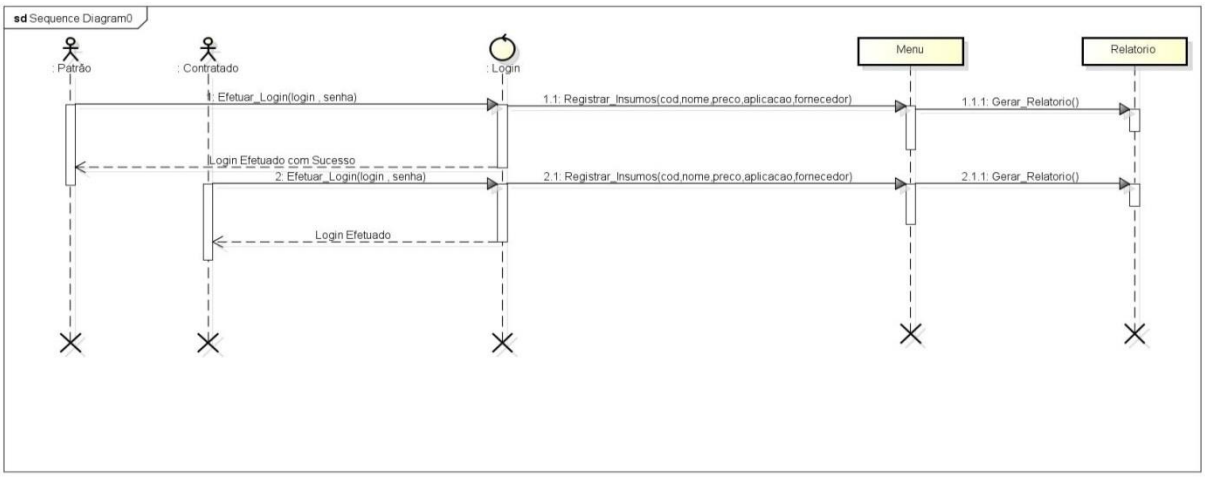

Figura 29 - Diagrama de Sequencia - Registro de Insumos.

Figura 30, abaixo o dono ou funcionário efetuam login, podem registrar as manutenções que são efetuadas nos maquinários, e também podem estar gerando o relatório através dele ter a relação do procedimento e custo.

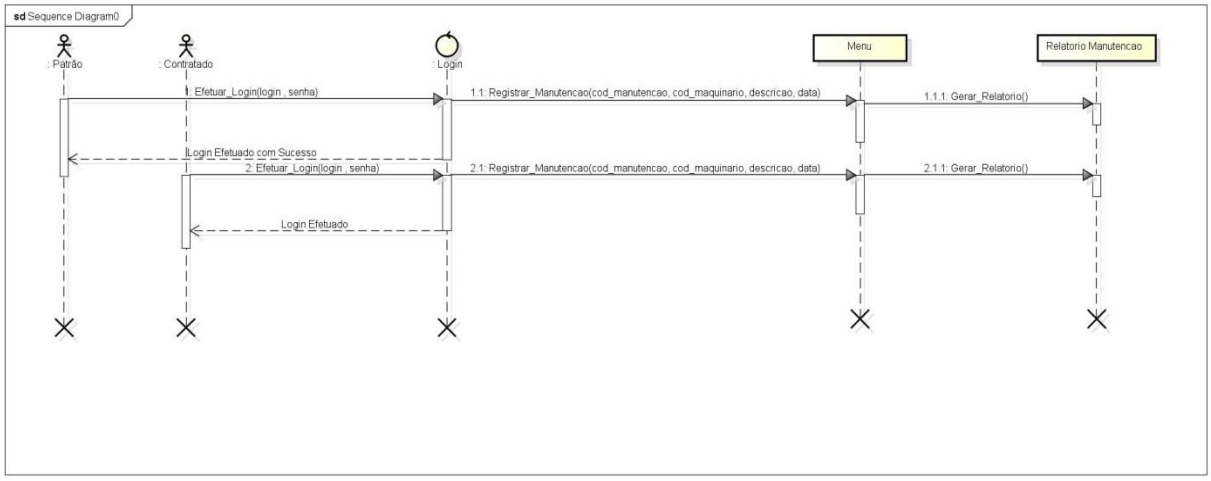

Figura 30 - Diagrama de Sequencia - Controle de Manutenções Maquinário.

Figura 31, logo abaixo realizando o controle dos maquinarios fazem parte da propriedade, tendo esses registros o dono sabe o que esta a disposição para desempenhar os trabalhos exigem mecanização.

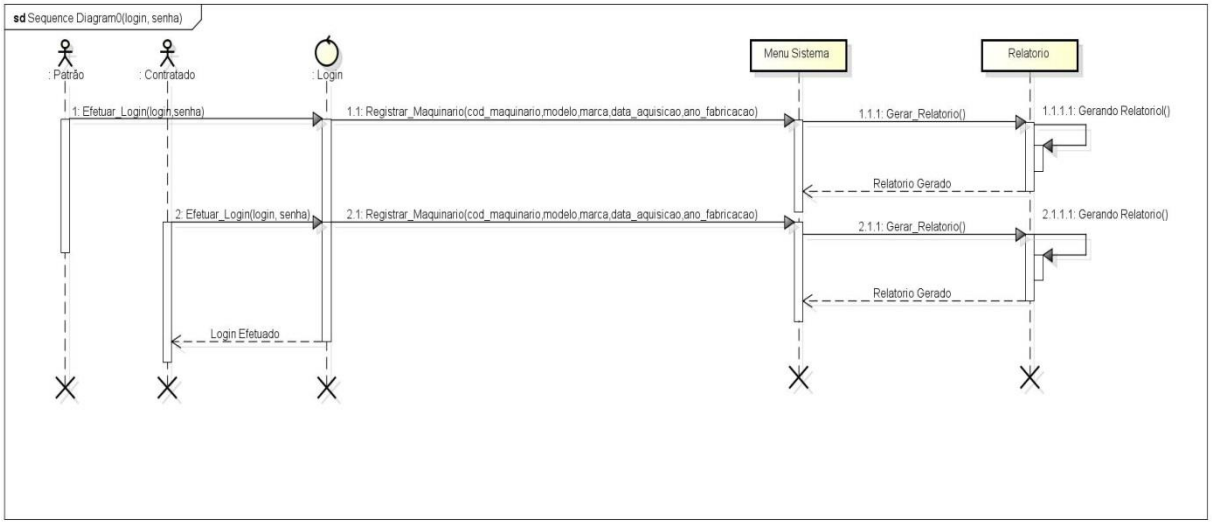

Figura 31 - Diagrama de Sequencia - Registra os Maquinários da propriedade.

Figura 32 abaixo ilustra o dono ou um de seus funcionários acessando o menu principal em seguida registrando um novo veiculo.

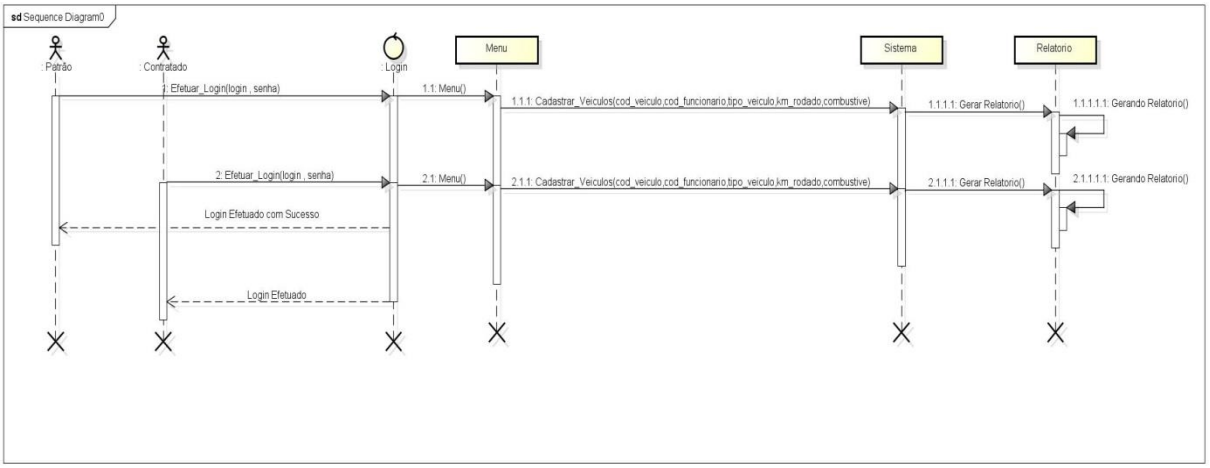

Figura 32 - Diagrama de Sequencia - Registro de veículos pertencem à empresa.

Figura 33 o respectivo dono ou um de seus funcionários efetuam o login e então podem consultar o Histórico de Produção.

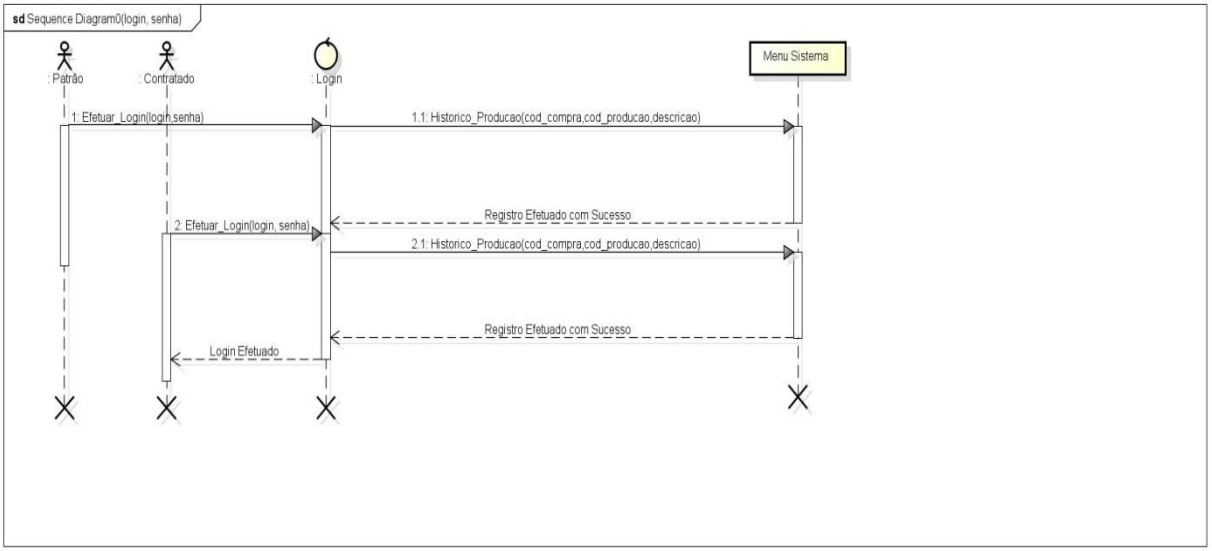

Figura 33 - Diagrama de Sequencia - Histórico de Produção.

Figura 34 o respectivo dono ou um de seus funcionários efetuam o login e então podem consultar o Histórico de Compras de Combustível.

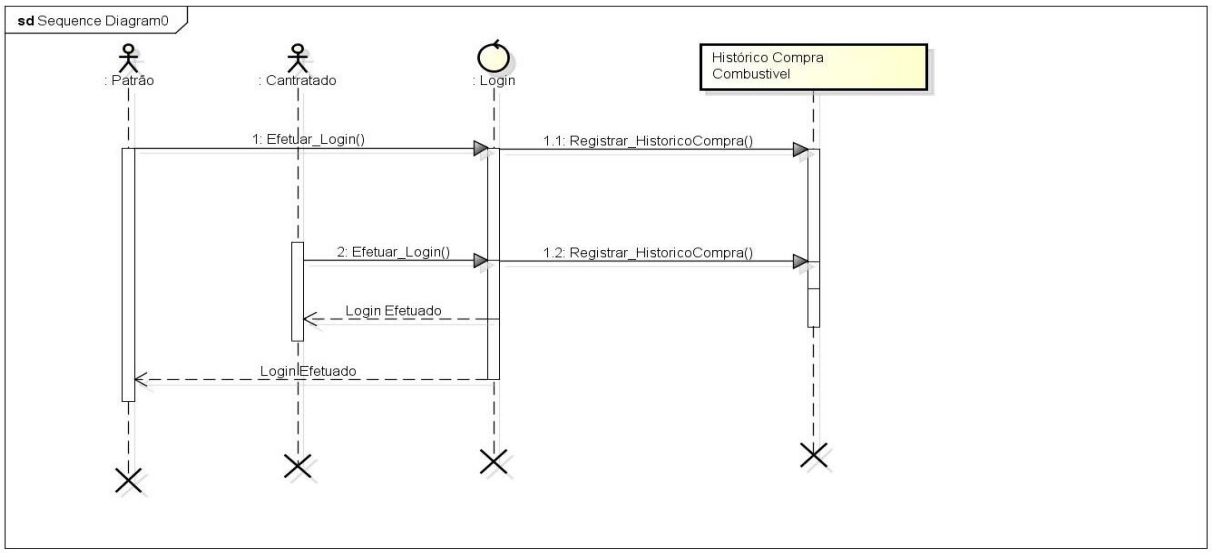

Figura 34 - Diagrama de Sequencia - Histórico de Compras de combustível.

3.4. DIAGRAMA DE CLASSE

Representa de forma abstrata estrutura das entidades envolvidas no sistema, e suas relações, por exemplo, se pensarmos em funcionário em uma aplicação, é representado através de uma classe que conterá os seus atributos por exemplo: nome, cidade, telefone, cargo, salario, endereço, etc. E os métodos são as ações podem ser realizadas exemplo atualizar, consultar, novo funcionário.

Segundo M. Tepfenhart (2001, p.510):

> Um diagrama de estrutura estática descreve a estrutura estática de um sistema. Em outras palavras, ele descreve como o sistema é estruturado e não como ele se comporta. Descreve “o que são as coisas” e seus relacionamentos estáticos com outras coisas [...] Classes definem os tipos de objeto que existem dentro do sistema [...] ter atributos que são geralmente membros de dados primitivos de objetos e operações podem [...] ser aplicados sobre os objetos. Diagramas de classe também mostram pacotes que agrupam classes, dependências entre classes e dependências entre os pacotes que as contem. Relacionamentos de generalização / especialização que correlacionam classes em um relacionamento do tipo supertipo / subtipo também podem ser incluídos.

Figura 35 abaixo representa o diagrama de classes neste são descritas todas as classes contidas, no sistema e trazem também os atributos, relacionamentos ou ligações entre as mesmas.

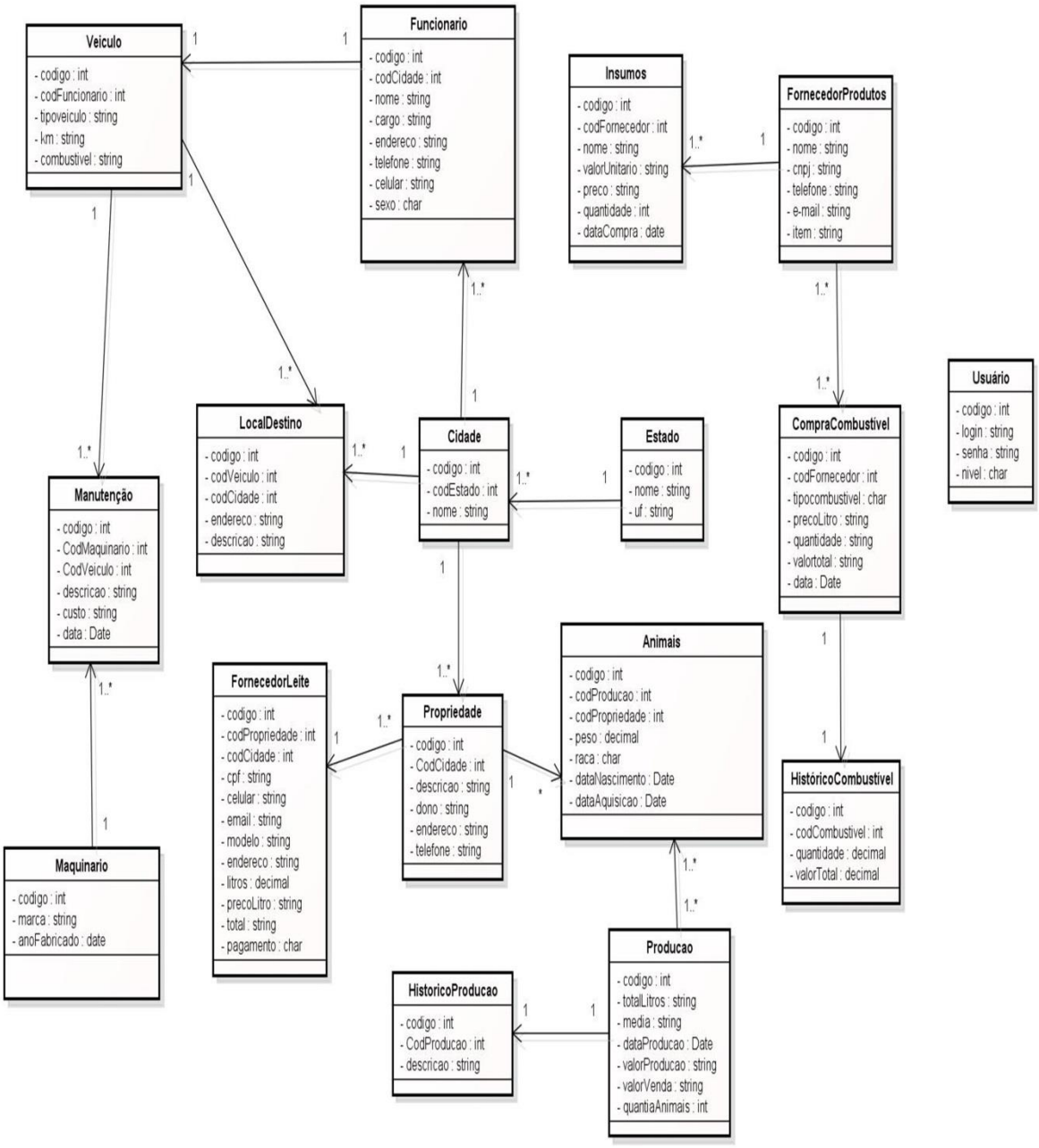

Figura 35 - Diagrama de Classes - Contem as classes do sistema.

3.5. WBS – WORK BREAKDOWN STRUCTURE

Diagrama abaixo auxilia na gerencia de projetos trata-se do EAP- (Estrutura Analítica de Projetos), em inglês Work breakdown structure consiste na divisão do trabalho em etapas.

Segundo o Guia PMI de Gerenciamento de Projetos (2004, p. 112):

> A WBS ou Estrutura Analítica de Projeto é uma decomposição hierárquica orientada à entrega do trabalho[...]. A EAP subdivide o trabalho do projeto em partes menores a mais facilmente gerenciáveis, em que cada nível descendente da EAP representa definição cada vez mais detalhada do projeto.

Ilustração **Figura - 37** abaixo descrevem as etapas do projeto, a partir de uma gera a próxima derivada da anterior. Retângulo a esquerda rotulado de “Fase Inicial” antecede as etapas subsequentes “Levantamento de Requisitos” em seguida “Especificação de requisitos”.

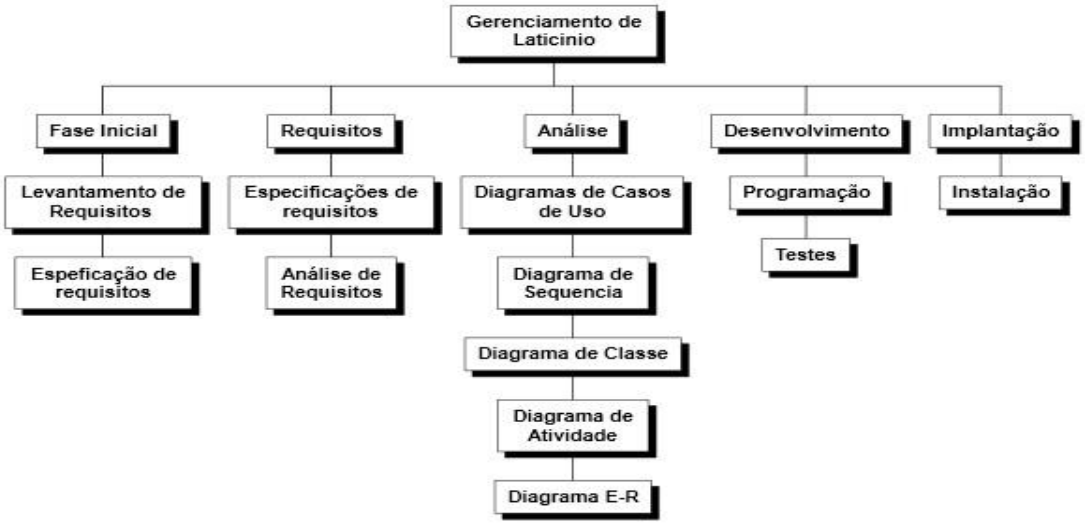

Figura 36 - Diagrama - EAP - Estrutura Analítica Projeto.

3.6. DIAGRAMA (E-R) ENTIDADE RELACIONAMENTO.

Segundo e Odell (1996):

> Para os diagramas ER, as associações são expressas ou dentro ou fora de tipos de entidade.[...] dentro são denominadas tipos de atributos. Aquelas que estão fora são representadas em termos de tipos de relação com outros tipos de entidade.
> [...] objetos podem ser expressos como um diagrama ER que descreve tipos de entidade com tipos de atributos ER e tipos de relação. (Martin e Odell, 1996, 569).

Figura 37 - Representa estrutura relacional do banco de dados mostrando todas as tabelas contidas no banco de dados, juntamente as suas relações, com as demais descreve os seus atributos o seu tipo, compõe as suas colunas.

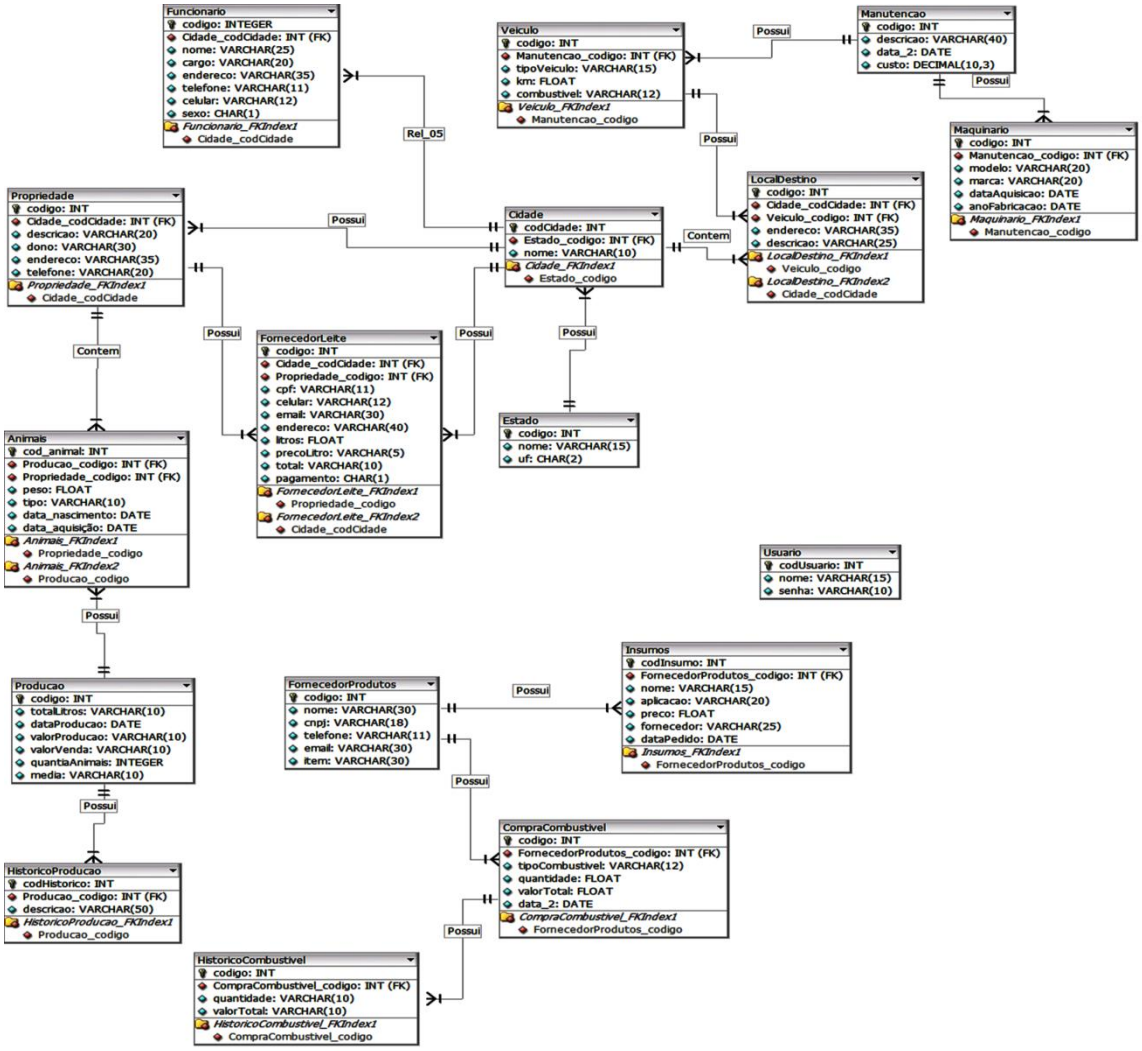

Figura 37- Diagrama Banco de Dados.

4. ESTRUTURA DE IMPLEMENTAÇÃO

Metodologia de desenvolvimento utilizada é dividida em três partes, no conceito de programação, são essas partes são denominadas de camadas, divide-se em três cada uma no qual tem sua função, vantagem resulta em uma aplicação mais organizada com códigos mais “limpos”, facilita a manutenção do software uma vez que está dividido em varias partes.

Veremos abaixo o nome, e sua função:

4.1. DATA ACCESS LAYER (DAL)

Única camada realiza o acesso aos dados, fazendo as operações no banco de dados, boa prática de programação nesta camada, evite colocar validações classe ou qualquer outro código que não seja relaciona diretamente com acesso.

4.2. BUSINESS LOGIC LAYER (BLL)

Camada contem as logicas das regras do negocio, ira fazer verificações e validações dos dados gerados posteriormente vão para o banco. Vale ressaltar que existe diferença entre regras de negocio e validações básicas, pois existem validações que não precisam ser efetuadas nesta classe. Exemplo validação do de CPF operação feita de forma geral, não especifica do sistema, pode ser implantada na camada de interface.

4.3. USER INTERFACE (UI)

Camada no qual há conjunto de comandos ou de menus através do qual haverá interação direta com o usuário final do sistema, esta parte nada mais é que as telas do sistema.

A estrutura do projeto divide-se em três partes conforme (**Figura 38**), onde existe pasta Camadas, dentro dela existem as demais partes compõe o sistema que são: BLL, DAL e MODEL contidos dentro das mesmas estão as classes.

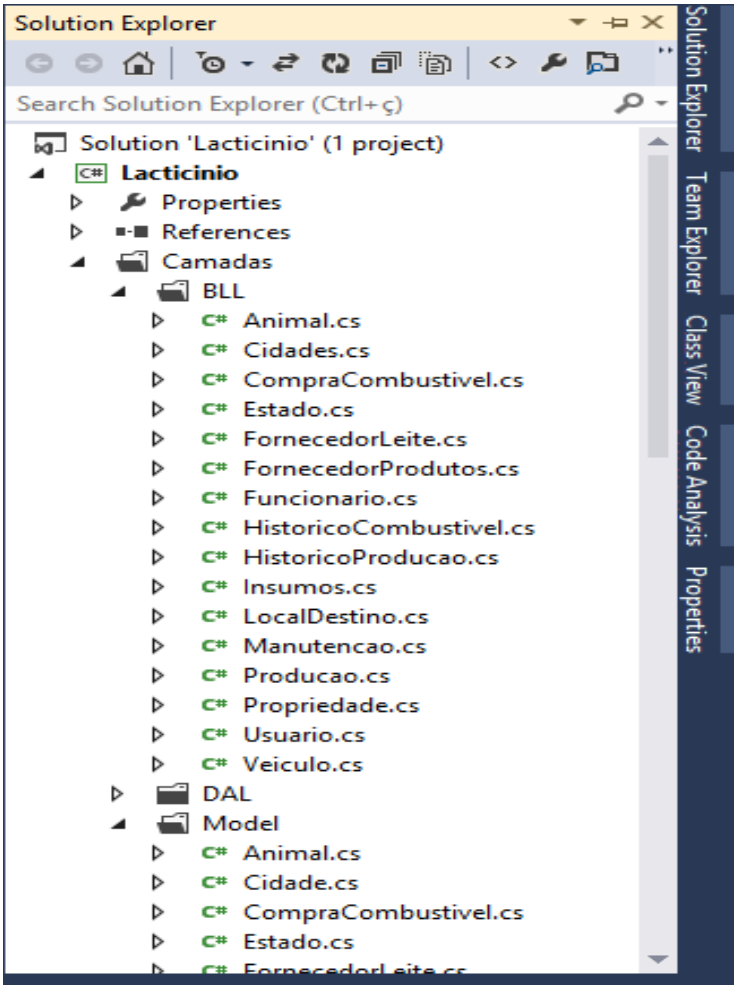

Figura 38 – Camadas

Como mencionado anteriormente a (**Figura - 39**) abaixo, trata-se de uma das classes estão inseridas dentro das partes que fazem parte do projeto, esta aqui mais especificamente falando, encontra-se na camada MODEL, traduzindo para o português significa MODELO, na programação o termo refere-se ao modelo de dados.

Onde definimos o objeto, os tipos de dados, seus atributos e geramos os métodos “get” para acesso e “set” para modificar.

```
using System.Text;
```

```
using System.Threading.Tasks;

namespace Lacticinio.Camadas.Model
{
  public  class CompraCombustivel
    {
        private int _codigo;
        private int _codFornecedor;
        private char _tipoCombustivel;
        private string _quantidade;
        private string _valorTotal;
        private DateTime _data;


        public int codigo
        {
            get { return _codigo; }
            set { _codigo = value; }
        }

        public int codFornecedor
        {
            get { return _codFornecedor; }
            set { _codFornecedor = value; }
        }


        public char tipoCombustivel
        {
            get { return _tipoCombustivel; }
            set { _tipoCombustivel = value; }
        }

        public string quantidade
        {
            get { return _quantidade; }
            set { _quantidade = value; }
        }

        public string valorTotal
        {
            get { return _valorTotal; }
            set { _valorTotal = value; }
        }
        public DateTime data
        {
            get { return _data; }
            set { _data = value;}
        }
```

Figura 39 - Classe de Compra de combustível.

Na BLL camada está expressa na (**Figura 40**), onde contem as regras do negocio sobre todas as operações Inserção, Atualização, Exclusão. Regra do código aqui

descrito não permite a exclusão de fornecedor, se houver uma compra efetuada em seu nome.

```
namespace Lacticinio.Camadas.BLL
{
  public  class FornecedorProdutos
    {
      public List<Model.FornecedorProdutos> Select()
      {
          DAL.FornecedorProdutos dalFornecedorProdutos = new DAL.FornecedorProdutos();
          return dalFornecedorProdutos.Select();
      }

      public Model.FornecedorProdutos SelectCod(int wCod) // regra de negocio busca
por codigo
      {
          DAL.FornecedorProdutos dalFornecedorProdutos = new DAL.FornecedorProdutos();
          return dalFornecedorProdutos.SelectCodFornecedor(wCod);
      }

     public List<Model.FornecedorProdutos> SelectNome(string wNome) // regra de
negocio de busca por nome
      {
          DAL.FornecedorProdutos dalFornecedorProdutos = new DAL.FornecedorProdutos();
          return dalFornecedorProdutos.SelectNome(wNome);
      }

      public List<Model.FornecedorProdutos> SelectItem(string wItem)
      {
          DAL.FornecedorProdutos dalFornecedorProdutos = new DAL.FornecedorProdutos();
          return dalFornecedorProdutos.SelectItem(wItem);
      }

      public void Insert(Model.FornecedorProdutos oFornPro)
      {
          if (oFornPro.codigo != 0 && oFornPro.item != string.Empty)
          {
              DAL.FornecedorProdutos dalFornecedorProdutos = new
             DAL.FornecedorProdutos();
              dalFornecedorProdutos.Insert(oFornPro);
          }
      }

      public void Update(Model.FornecedorProdutos oFornPro)
      {
          if (oFornPro.email != string.Empty && oFornPro.nome != string.Empty)
          {
              DAL.FornecedorProdutos dalFornecedorProdutos = new
             DAL.FornecedorProdutos();
              dalFornecedorProdutos.Update(oFornPro);
          }
      }

public void Update(Model.FornecedorProdutos oFornPro)
      {
          if (oFornPro.email != string.Empty && oFornPro.nome != string.Empty)
          {
              DAL.FornecedorProdutos dalFornecedorProdutos = new
```

```
            DAL.FornecedorProdutos();
            dalFornecedorProdutos.Update(oFornPro);
        }
    }

    public bool Delete(Model.FornecedorProdutos oFornPro)
    {

            DAL.FornecedorProdutos dalFornecedorProdutos = new
            DAL.FornecedorProdutos();
            DAL.CompraCombustivel dalCompra = new DAL.CompraCombustivel();
            Model.CompraCombustivel oCom = new Model.CompraCombustivel();
            BLL.CompraCombustivel bllCompraCombustivel = new CompraCombustivel();
            oCom = bllCompraCombustivel.SelectCodCombustivel(oFornPro.codigo);
            oCom = dalCompra.SelectCodFornecedor(oFornPro.codigo);
            DAL.Insumos dalInsumo = new DAL.Insumos();
            Model.Insumo oIns = new Model.Insumo();
            BLL.Insumos bllInsumo = new Insumos(); //Vincula FornecedorProdutos com
Insumo
            oIns = bllInsumo.SelecCodInsumo(oFornPro.codigo); //Camadas.DAL.Insumos
            oIns =
dalInsumo.SelectCodFornecedor(oFornPro.codigo);//Camadas.DAL.Insumos

            if(oCom.codFornecedor != oFornPro.codigo && oIns.codFornecedor !=
oFornPro.codigo)
            {
              dalFornecedorProdutos.Delete(oFornPro);
              return true;
            }
            else
            {
             MessageBox.Show("Exitem Insumos vinculados à este Fornecedor com Codígo
"
                                + oFornPro.codigo, "Remoção Cancelada",
                                MessageBoxButtons.OK, MessageBoxIcon.Error);
                return false;
            }
    }
    public bool Delete(Model.FornecedorProdutos oFornPro)
    {

            DAL.FornecedorProdutos dalFornecedorProdutos = new
            DAL.FornecedorProdutos();
            DAL.CompraCombustivel dalCompra = new DAL.CompraCombustivel();
            Model.CompraCombustivel oCom = new Model.CompraCombustivel();
            BLL.CompraCombustivel bllCompraCombustivel = new CompraCombustivel();
            oCom = bllCompraCombustivel.SelectCodCombustivel(oFornPro.codigo);
            oCom = dalCompra.SelectCodFornecedor(oFornPro.codigo);
            DAL.Insumos dalInsumo = new DAL.Insumos();
            Model.Insumo oIns = new Model.Insumo();
            BLL.Insumos bllInsumo = new Insumos(); //Vincula FornecedorProdutos com
Insumo
            oIns = bllInsumo.SelecCodInsumo(oFornPro.codigo); //Camadas.DAL.Insumos
            oIns = dalInsumo.SelectCodFornecedor(oFornPro.codigo);

            if(oCom.codFornecedor != oFornPro.codigo && oIns.codFornecedor !=
oFornPro.codigo)
            {
              dalFornecedorProdutos.Delete(oFornPro);
              return true;
```

```
            }
            else
            {
             MessageBox.Show("Exitem Insumos vinculados à este Fornecedor com Codígo
                                 " + oFornPro.codigo, "Remoção Cancelada",
                                MessageBoxButtons.OK, MessageBoxIcon.Error);
           return false;

            }
      }
```

Figura 40 - BLL de Fornecedor Produtos.

A classe ValidaCPF (**Figura - 41**) trata os dados inseridos pelo usuário, primeiro verifica quantidade em seguida realiza a multiplicação com os valores informados encontra o primeiro e segundo digito verificador, fazendo isto verifica se o mesmo é valido ou não, se não for não prossegue a operação.

```
  namespace Lacticinio
{
  public class ValidaCPF
    {
      public static bool IsCpf(string cpf)
      {
          //declaracao das variaveis
          int[] multiplicador1 = new int[9] { 10, 9, 8, 7, 6, 5, 4, 3, 2};
          int[] multiplicador2 = new int[10] { 11, 10, 9, 8, 7, 6, 5, 4, 3, 2};
          string veriCpf; //tempCpf
          string digito;
          int soma;
          int resto;

          cpf = cpf.Trim();
          cpf = cpf.Replace(".", "").Replace("-","");

          if (cpf.Length != 11) //verifica quantdade de digitos
              return false;
          veriCpf = cpf.Substring(0,9);
          soma = 0;

          for (int i = 0; i < 9; i++) //encontra o primeiro digito verifacador
              soma += int.Parse(veriCpf[i].ToString()) * multiplicador1[i];

          resto = soma % 11;
          if (resto < 2)
              resto = 0;
          else
              resto = 11 - resto;
          digito = resto.ToString();

          veriCpf = veriCpf + digito;
```

```
            soma = 0;
            for (int i = 0; i < 10; i++) //encontra segundo digito verificador
                soma += int.Parse(veriCpf[i].ToString()) * multiplicador2[i];

            resto = soma % 11;
            if (resto < 2)
                resto = 0;
            else
                resto = 11 - resto;
            digito = digito + resto.ToString();
            return cpf.EndsWith(digito);

        }
    }
}
```

Figura 41 - Validação de CPF.

Abaixo na (**Figura 42**) está descrito evento clique de o botão validar, o código está no formulário, funciona juntamente com a classe (Figura 37), descrita anteriormente retornando uma mensagem para o usuário, se o CPF ou CNPJ informados são validos.

```
private void Validar()
 {
            string mensagem = string.Empty;
            string valor = mskCnpj.Text;
            if (rdbCnpj.Checked)
            {
                if (Lacticinio.ValidaCNPJ.IsCnpj(valor))
                {
                    mensagem = "O número de CNPJ é Válido:...";
                    verifica = true;
                }
                else
                {

                    mensagem = "O número de CNPJ é Invalido:.. ";
                    verifica = false;
                }

            }
            else if (rdbCpf.Checked)
            {
                if (Lacticinio.ValidaCPF.IsCpf(valor))
                {
```

```
                mensagem = "O número de CPF Válido !";
                verifica = true;
            }
            else
            {
                mensagem = "O número de CPF é Inválido !";
                verifica = false;
            }

        }

    }
```

Figura 42 – Código do evento validação do botão validar.

5. CONSIDERAÇÕES FINAIS

Com desenvolvimento de qualquer trabalho nós sempre aprendemos algo novo comigo não foi diferente pude aprender coisas novas e relembrar alguns conceitos, já foram estudados anteriormente. Atraves desta documentação, em conjunto com os diagramas de caso de uso, classe, sequencia, diagrama EAP e modelo de Banco de Dados somados a parte escrita pude ter uma visão mais compreensiva do sistema e seu funcionamento. O software em si possibilitara um apronfundamento nos conceitos para mim, e ao usuário terá uma compreensão mais ampla uma vez que o sistema auxiliara no gerenciamento das atividades.

Sistema trazerá diversas melhorias no local onde será implantado, pelo motivo de informatizar diversas atividades que outrora eram, relizadas manualmente com caneta e papeis, muitas vezes eram perdidos, portanto anotações eram realizadas mas não eram de fato resgistradas.

6. REFERENCIAS

C# Como Programar . AUTOR: DEITEL.

http://pt.wikipedia.org/wiki/Microsoft_SQL_Server. Acesso em 01 de Março de 2014.

http://astah.net/editions/community. Acesso em 01 de Março de 2014.

Freeze, Wayne S. SQL - Guia de referencia do programador. Tradução Tereza Queiroz.

http://www.fabforce.net/dbdesigner4/. Disponível no link: http://translate.google.com.br/translate?hl=pt-BR&sl=en&u=http://www.fabforce.net/dbdesigner4/&prev=/search%3Fq%3Ddbdesigner%26biw%3D1366%26bih%3D664 . Acesso em 19 de fevereiro de 2014.]

Herbert Moroni Cavallari da Costa Gois - Curso Banco de dados com C# e o Visual Studio .NET 2005.

Lee, Richard C e Tepfenhart, William M. UML e C++ Guia Prático de Desenvolvimento Orientado a Objeto. Tradução de Celso Roberto Paschoa. São Paulo, Editora Morkron books Ltda., 2001.

Carlos Olavo de Azevedo Camacho Júnior - Guia prático para o desenvolvimento de Aplicações C# em Camadas**.**

Andrew Stellman e Jennifer Greene – Livro: Use a Cabeça! C#.

http://engineer-scraps.blogspot.com.br/2012/02/o-que-e-visual-studio.html. Acesso em 17 de fevereiro de 2014.