

Mario Henrique Villani

**AGROSOFTWARE**

Assis

2011

Mario Henrique Villani

## AGROSOFTWARE

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito parcial e obtenção do Certificado de Conclusão.

Orientado: Mario Henrique Villani

Orientador: Luiz Carlos Begosso

Assis

2011

## DEDICATÓRIA

Dedico este trabalho a minha família,  
que me deu todo apoio e suporte para  
que eu fizesse um curso de  
graduação.

## AGRADECIMENTOS

Agradeço a minha família pelo apoio durante esses três anos de curso.

Ao professor Dr. Luiz Carlos Begosso pela Orientação e paciência.

A todos os professores do curso de Processamento de Dados que me ensinaram muito durante todo o curso.

A todos os meus amigos que me apoiarão.

## RESUMO

O agronegócio é um setor que necessita de softwares de gerenciamento, é muito difícil encontrar softwares de qualidade no mercado. Daí surge a ideia de desenvolver um sistema que ajude os agricultores a ter uma melhor gestão de sua propriedade.

O sistema possibilita ao usuário ter um melhor controle das atividades desenvolvidas em suas propriedades. Para o desenvolvimento do sistema foi utilizado a plataforma ASP.NET com a linguagem de programação C# e Sql Server 2005 como banco de dados.

## LISTA DE FIGURAS

|  |    |
|--|----|
| <b>Figura 1:</b> Diagrama de Caso de Uso Geral.....  | 22 |
| <b>Figura 2:</b> Caso de Uso: UseCase_01 - Cadastro de Funcionário.....                        | 23 |
| <b>Figura 3:</b> Caso de Uso: UseCase_02 - Cadastrar Fornecedor .....                          | 25 |
| <b>Figura 4:</b> Caso de Uso: UseCase_03 - Cadastrar Propriedade .....                         | 27 |
| <b>Figura 5:</b> Caso de Uso: UseCase_04 - Cadastrar Talhão.....                               | 29 |
| <b>Figura 6:</b> Caso de Uso: UseCase_05 - Cadastrar Atividade.....                            | 31 |
| <b>Figura 7:</b> Caso de Uso: UseCase_06 - Cadastrar Maquinário.....                           | 33 |
| <b>Figura 8:</b> Caso de Uso: UseCase_07 - Cadastrar Cultura.....                              | 35 |
| <b>Figura 9:</b> Caso de Uso: UseCase_08 - Cadastrar Compra de Insumo.....                     | 37 |
| <b>Figura 12:</b> Caso de Uso: UseCase_11 - Cadastrar Compra de Peças.....                     | 39 |
| <b>Figura 13:</b> Caso de Uso: UseCase_12 - Cadastrar Venda de Produção.....                   | 43 |
| <b>Figura 14:</b> Caso de Uso: UseCase_13 - Cadastrar Histórico de Produção.....               | 45 |
| <b>Figura 19:</b> Caso de Uso: UseCase_18 - Gerar Relatório de Cultura por Talhão.....         | 47 |
| <b>Figura 21:</b> Caso de Uso: UseCase_20 - Gerar Relatório de Compra de Peças.....            | 48 |
| <b>Figura 22:</b> Caso de Uso: UseCase_21 - Gerar Relatório de Funcionários.....               | 49 |
| <b>Figura 23:</b> Caso de Uso: UseCase_22 - Gerar Relatório de Manutenção dos Maquinários..... | 50 |
| <b>Figura 24:</b> Caso de Uso: UseCase_23 - Gerar Relatório de Produção por Safra.....         | 51 |
| <b>Figura 25:</b> Caso de Uso: UseCase_24 - Gerar Relatório de Vendas.....                     | 52 |
| <b>Figura 26:</b> Caso de Uso: UseCase_25 - Gerar Relatório de Contas a Pagar.....             | 53 |
| <b>Figura 27:</b> Caso de Uso: UseCase_26 - Gerar Relatório de Atividade.....                  | 54 |
| <b>Figura 28:</b> Caso de Uso: UseCase_27 - Gerar Gráfico de Produção por Safra.....           | 55 |

|  |    |
|--|----|
| <b>Figura 29:</b> DiagramSequence_01 - Cadastrar Funcionário.....                            | 57 |
| <b>Figura 30:</b> DiagramSequence_02 – Cadastrar Fornecedor.....                             | 58 |
| <b>Figura 31:</b> DiagramSequence_03 – Cadastrar Propriedade.....                            | 59 |
| <b>Figura 32:</b> DiagramSequence_04 – Cadastrar Talhão.....                                 | 60 |
| <b>Figura 33:</b> DiagramSequence_05 – Cadastrar Atividade.....                              | 61 |
| <b>Figura 34:</b> DiagramSequence_06 – Cadastrar Maquinário.....                             | 62 |
| <b>Figura 35:</b> DiagramSequence_07 - Cadastrar Cultura.....                                | 63 |
| <b>Figura 36:</b> DiagramSequence_08 - Cadastrar Compra de Insumo.....                       | 64 |
| <b>Figura 39:</b> DiagramSequence_11 - Cadastrar Compra de Peças.....                        | 64 |
| <b>Figura 40:</b> DiagramSequence_12 - Cadastrar Venda de Produção.....                      | 66 |
| <b>Figura 41:</b> DiagramSequence_13 - Cadastrar Histórico de Produção.....                  | 67 |
| <b>Figura 46:</b> DiagramSequence_18 - Gerar Relatório de Cultura por Talhão.....            | 68 |
| <b>Figura 47:</b> DiagramSequence_19 - Gerar Relatório de Gastos com Colheita.....           | 69 |
| <b>Figura 48:</b> DiagramSequence_20 - Gerar Relatório de Compra de Peças.....               | 70 |
| <b>Figura 49:</b> DiagramSequence_21 - Gerar Relatório de Funcionários.....                  | 71 |
| <b>Figura 50:</b> DiagramSequence_22 - Gerar Relatório de Manutenção dos<br>Maquinários..... | 72 |
| <b>Figura 51:</b> DiagramSequence_23 - Gerar Relatório de Produção por Safra.....            | 73 |
| <b>Figura 52:</b> DiagramSequence_24 - Gerar Relatório de Vendas.....                        | 74 |
| <b>Figura 53:</b> DiagramSequence_25 - Gerar Relatório de Contas a Pagar.....                | 75 |
| <b>Figura 54:</b> DiagramSequence_26 - Gerar Relatório de Atividade.....                     | 76 |
| <b>Figura 55:</b> DiagramSequence_27 - Gerar Gráfico de Produção por Safra.....              | 77 |
| <b>Figura 56:</b> Diagrama de Classes.....   | 79 |
| <b>Figura 57:</b> Diagrama Entidade Relacionamento.....                                      | 81 |
| <b>Figura 58:</b> Estrutura Analítica do Projeto.....  | 83 |
| <b>Figura 54:</b> Sequenciamento de Atividades.....  | 85 |

|  |     |
|--|-----|
| <b>Figura 55:</b> Camadas.....                 | 86  |
| <b>Figura 56:</b> ORM.....                     | 87  |
| <b>Figura 57:</b> DAL.....                     | 88  |
| <b>Figura 58:</b> executarComando.....         | 89  |
| <b>Figura 59:</b> Consultar.....               | 89  |
| <b>Figura 60:</b> Métodos.....                 | 90  |
| <b>Figura 61:</b> BLL.....                     | 91  |
| <b>Figura 62:</b> AGROSOFTWARE_Service.....    | 92  |
| <b>Figura 63:</b> VIEWAGROSOFTWARE.....        | 92  |
| <b>Figura 64:</b> Menu.....                    | 93  |
| <b>Figura 65:</b> Cadastro de Funcionário..... | 94  |
| <b>Figura 66:</b> Estoque Insumos.....         | 95  |
| <b>Figura 67:</b> Compra.....                  | 96  |
| <b>Figura 68:</b> Atividade.....               | 97  |
| <b>Figura 69:</b> Relatórios.....              | 98  |
| <b>Figura 70:</b> Orçamento.....               | 102 |
| <b>Figura 71:</b> Cronograma.....              | 103 |

## SUMÁRIO

|   |           |
|---|-----------|
| <b>1. INTRODUÇÃO.....</b>                 | <b>11</b> |
| 1.1 OBJETIVOS.....                        | 12        |
| 1.2 PÚBLICO ALVO.....                     | 13        |
| 1.3 JUSTIFICATIVA.....                    | 14        |
| <b>2 MÉTODO DE DESENVOLVIMENTO.....</b>   | <b>15</b> |
| 2.1 METODOLOGIA DE ANÁLISE.....           | 15        |
| 2.2 C# (SHARP), (MICROSOFT.NET).....      | 16        |
| 2.3 BANCO DE DADOS SQL SERVER 2005.....   | 17        |
| 2.4 MODELAGEM DO SISTEMA.....             | 18        |
| <b>3 DESCRIÇÃO GERAL DO SISTEMA.....</b>  | <b>19</b> |
| 3.1 ANÁLISE DE REQUISITOS.....            | 19        |
| 3.2 LISTA DE EVENTOS.....                 | 20        |
| 3.3 DIAGRAMA DE CASO DE USO GERAL.....    | 21        |
| 3.4 CASO DE USO.....                      | 23        |
| 3.5 DIAGRAMA DE SEQUENCIA.....            | 56        |
| 3.6 DIAGRAMA DE CLASSES.....              | 78        |
| 3.7 DIAGRAMA ENTIDADE RELACIONAMENTO..... | 80        |
| 3.8 ESTRUTURA ANALÍTICA DO PROJETO.....   | 82        |
| 3.9 SEQUENCIAMENTO DE ATIVIDADES.....     | 84        |
| <b>4 IMPLEMENTAÇÃO.....</b>               | <b>86</b> |

|  |            |
|--|------------|
| <b>4.1 ORGANIZAÇÃO DO PROJETO.....</b> | <b>86</b>  |
| <b>4.2 INTERFACES DO SISTEMA.....</b>  | <b>93</b>  |
| <b>5 ORÇAMENTO.....</b>                | <b>99</b>  |
| <b>5.1 ESTIMATIVA DE CUSTOS.....</b>   | <b>100</b> |
| <b>5.2 CUSTOS DE MATERIAIS.....</b>    | <b>101</b> |
| <b>5.3 ORÇAMENTO DO PROJETO.....</b>   | <b>102</b> |
| <b>6 CRONOGRAMA.....</b>               | <b>103</b> |
| <b>7 CONCLUSÃO.....</b>                | <b>104</b> |
| <b>8 REFERENCIAS.....</b>              | <b>105</b> |

## 1. INTRODUÇÃO

O agronegócio brasileiro provou em várias situações e contextos que é uma atividade que merece respeito e incentivo, a tecnologia por sua vez pode contribuir e muito para a melhoria de condições para o campo. A área tecnológica não dava tanta importância para o setor, tanto é que no mercado quando se fala em software para o agronegócio fica muito a desejar, pois muitas vezes não se encontra softwares específicos para uma determinada área do ramo e quando encontrado infelizmente estão incompletos. Em algumas cooperativas e associações quando se pergunta sobre a necessidade dos produtores simplesmente eles não sabem responder qual software o produtor precisa para ajudar no processo de tomadas de decisões e no controle de sua atividade. Quando se fala em um software para o agronegócio não se deve pensar somente em um programa que controle o rebanho ou uma lavoura específica, é preciso pensar em um software que ajude não somente a controlar uma determinada atividade, mas também auxiliar no processo de tomada de decisão, um software capaz de ajudar o produtor a tomar a melhor decisão em determinados assuntos ou em num determinado momento do processo de produção.

Nos casos que a tecnologia está envolvida no ambiente de produção já provou que esta pode ser de grande valia para a lida no campo e, neste processo em que a fazenda deixou de ser uma atividade de fim de semana para se tornar uma empresa isso pode fazer a diferença. A figura do computador está cada vez mais presente nos escritórios das fazendas ou sítios seja para consulta ou para busca de informações. Basta olhar para a agricultura de precisão para termos uma noção do que a tecnologia pode fazer para ajudar.

Nesse momento em que o agronegócio procura constantemente melhores índices de produtividade a tecnologia pode ajudar e muito nesse processo.

## 1.1. OBJETIVOS

Este trabalho de Conclusão de Curso tem por objetivo geral: desenvolver um sistema que ajude os produtores rurais nas suas atividades de controle e gerenciamento do agronegócio, contribuir para a melhoria nas condições para o campo e assim proporcionando para o agronegócio brasileiro, mais oportunidades de negócios para chamar a atenção da área da informática para o campo que é carente de softwares.

Como objetivo específico o software será capaz de: controlar os funcionários, maquinários, estoque, compra e venda de produtos, previsão de produção, previsão dos custos, pragas, correção da terra, custos por talhão e controle financeiro da propriedade.

## 1.2. PÚBLICO ALVO

O sistema Agrossoftware, desenvolvido para agricultores de pequeno e médio porte, que tenham como cultivo as culturas de soja e milho.

### 1.3. JUSTIFICATIVA

Os pequenos e médios produtores por muitas vezes desconhecem o uso de softwares no setor do agronegócio. Desenvolver novas ferramentas para o setor pode ser uma ótima idéia para empresas da área.

Os custos de produção na agricultura tem aumentado consideravelmente, nos últimos anos. A partir dessa nova realidade, tornou-se necessária a utilização de eficiente controle desses custos para garantir lucros ao negócio. De forma geral, o agricultor não tem controle eficaz sobre suas propriedades, funcionários, maquinários e custos de produção. O presente trabalho se justifica por apresentar uma alternativa de software objetivando auxiliar o proprietário da terra na tomada de decisão.

## **2. MÉTODO DE DESENVOLVIMENTO**

### **2.1. METODOLOGIA DE ANÁLISE**

Para a análise e desenvolvimento do sistema, como na implementação, será utilizada a orientação a objetos. O termo orientação a objetos significa organizar o mundo real como uma coleção de objetos que incorporam estrutura de dados e um conjunto de operações que manipulam estes dados.

## 2.2. C#

A ferramenta escolhida para o desenvolvimento foi Visual Studio 2010, que utilizará a linguagem C#, a linguagem conta com um Framework incrível que ajuda muito na implementação, além dos métodos *DataTable*, *DataSet* e *DataSource*, esses métodos facilitam muito os métodos de consulta, popular tabelas, listas etc... .

A ferramenta ainda conta com um ótimo sistema de *Debug* que facilita muito para encontrar erros e entender melhor como o aplicativo funciona.

Também escrito como C# ou C Sharp, é uma linguagem de programação orientada a objetos, fortemente tipada, desenvolvida pela Microsoft como parte da plataforma .NET. A sua sintaxe orientada a objetos foi baseada no C++ mas inclui muitas influências de outras linguagens de programação, como *Object Pascal* 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, e pouco das inconsistências e anacronismo, resultando em uma linguagem mais limpa e lógica. (Sharp, 2007, 4)

### 2.3. BANCO DE DADOS SQL SERVER 2005

O Banco de Dados será o Microsoft SQL Server 2005.

Segundo Soukup (1998)

O Microsoft SQL Server é um sistema de gerenciamento de banco de dados relacional cliente/servidor (SGBDR) de alto desempenho. Ele foi projetado para suportar processamento de transação de grande volume (como o de entrada de pedidos online, estoque, contabilidade ou manufatura). (Soukup, 1998, 31)

## 2.4. MODELAGEM DO SISTEMA

Para a modelagem do sistema, será utilizada a ferramenta Astah Professional 6.1 que é um editor UML, leve, integrado com DER (Diagrama entidade relacionamento é um modelo diagramático que descreve o modelo de dados de um sistema com alto nível de abstração), DFD (O diagrama de fluxos de dados é uma ferramenta para a modelagem de sistemas. O mesmo fornece apenas uma visão do sistema, a visão estruturada das funções, ou seja, o fluxo dos dados), CRUD (*Create, Retrieve, Update e Delete* para as quatro operações básicas utilizadas em bancos de dados relacionais (RDBMS) ou em interface para usuários para criação, consulta, atualização e destruição de dados) e outros recursos de mapeamento *Mind* para desenvolvedores de software.

### 3. DESCRIÇÃO GERAL DO SISTEMA

#### 3.1. ANÁLISE DE REQUISITOS

Para o desenvolvimento do sistema não foi realizada entrevista padrão com o usuário por não existir um cliente específico. Porém todas as necessidades do sistema foram levantadas a partir de conversas com o agricultor Antonio Villani.

As necessidades são:

- Dividir a propriedade por talhão para uma melhor visão das atividades.
- Desenvolver um cadastro onde o usuário deverá efetuar o cadastro de todas as atividades a serem desenvolvidas. O cadastro deverá conter informações importantes como quais funcionários trabalham na atividade, quais maquinários foram utilizados, quais os tipos e quantidade de insumos foram gastos, entre outras informações.
- O agricultor Antonio disse sobre a necessidade de criação de um histórico de produção que irá gerar gráficos informativos sobre produção e custos, comparações entre safras de diferentes anos.
- Disponibilizar informações importantes como previsão do tempo e cotação de grãos.

#### 3.2. Lista de Eventos

1. Cadastrar Funcionário
2. Cadastrar Fornecedor
3. Cadastrar Propriedade
4. Cadastrar Talhão
5. Cadastrar Atividade
6. Cadastrar Maquinário
7. Cadastrar Cultura
8. Cadastrar Compra de Insumos
9. Cadastrar Compra de Combustível
10. Cadastrar Compra de Peças
11. Cadastrar Venda de Produção
12. Cadastrar Histórico de Produção
13. Gerar Relatório de Insumos gastos por safra
14. Gerar Relatório de Cultura por Talhão
15. Gerar Relatório de Compra de Peças
16. Gerar Relatório de Funcionários
17. Gerar Relatório de Manutenção dos Maquinários
18. Gerar Relatório de Produção por Safra
19. Gerar Relatório de Vendas
20. Gerar Relatório de Contas a Pagar
21. Gerar Relatório de Atividade
22. Gerar Gráfico de Produção por Safra

### 3.3. DIAGRAMA DE CASO DE USO GERAL

Para a especificação dos eventos será utilizado o Diagrama Use Case (Caso de Uso(Figura 1)).

Segundo C. Lee e Tepfenhart (2001).

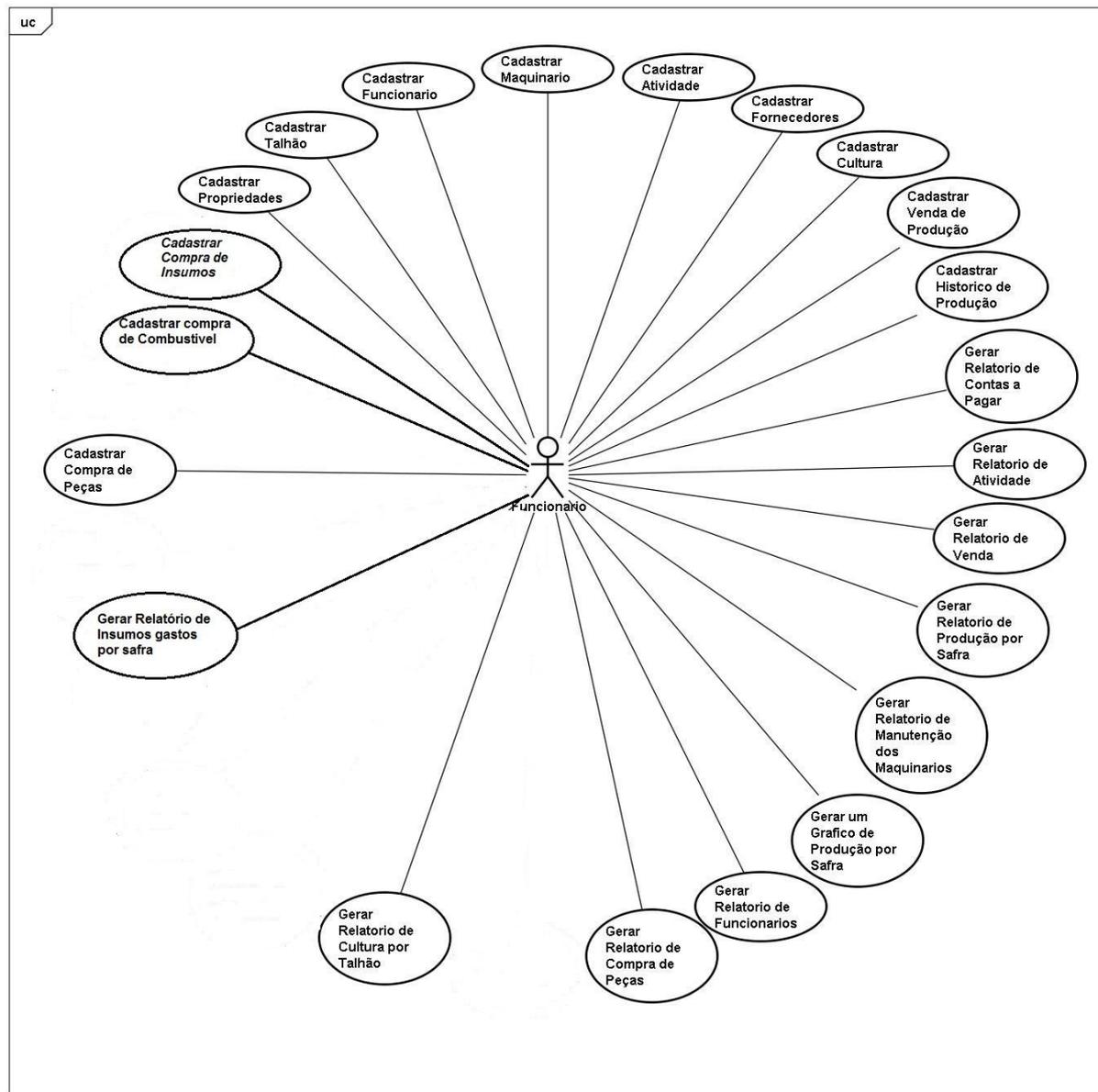
Um ator (*actor*) é uma entidade externa que interage com um sistema.

Um caso de uso (*use case*) é a descrição de uma interação que consegue um objetivo proveitoso para um ator.

Um grupo de casos de uso (*use case bundle*) é uma coleção de casos de uso estreitamente correlacionada com alguma atividade ou elemento organizador de negócio.

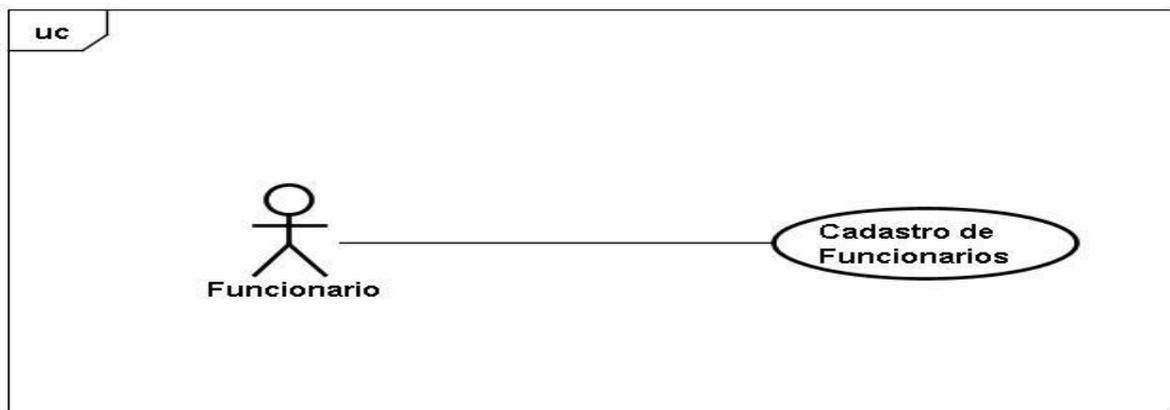
Um grupo de casos de uso nos propicia um modo de organizar nossos casos de uso em coleções que cooperarão conosco para entendermos melhor a funcionalidade do sistema que estamos desenvolvendo. (C. Lee e Tepfenhart, 2001, 42).

A Figura 1 ilustra o caso de uso que pode ser representado por uma elipse contendo, internamente, o nome do caso de uso.



**Figura 1 - Ilustra o Diagrama de Caso de Uso Geral.**

### 3.4. DIAGRAMA DE CASO DE USO



**Figura 2 - Ilustra o Caso de Uso: UseCase\_01 - Cadastrar Funcionário**

### 1. Finalidade/Objetivo

- Permite ao usuário cadastrar todos os funcionários no sistema.

### 2. Ator

- Funcionário

### 3. Precondições

- O usuário deve ter efetuado login como Administrador

### 4. Fluxo principal

- O usuário seleciona a opção cadastrar funcionário no menu principal e em seguida escolhe a opção cadastrar funcionário.[A1].[A2].
- O sistema solicita os dados necessários para o cadastro do funcionário
- O usuário fornece os dados e confirma a operação.[A3].
- O sistema verifica se todos foram fornecidos e em seguida mostra uma mensagem de confirmação.

### 5. Fluxo Alternativo

**A1.** O usuário seleciona a opção Excluir Funcionário.

- O sistema solicita o código do funcionário a ser excluído.

**A2.** O usuário escolhe a opção Alterar Funcionário.

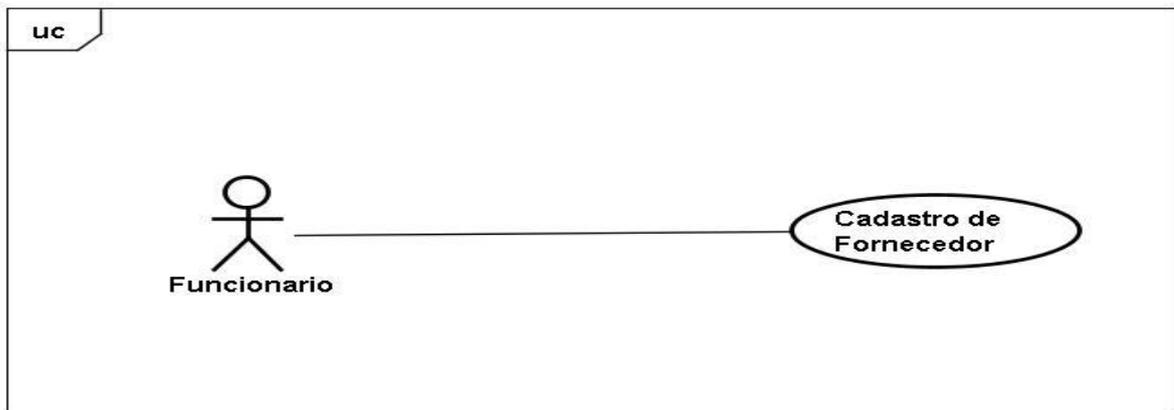
- O sistema solicita o código do funcionário a ser alterado.

**A3.** O usuário escolhe a opção Cancelar.

- O sistema volta ao passo 4.a do Fluxo Principal.

## **6. Casos de Teste**

- Verificar se todos os campos foram preenchidos corretamente (campos Numéricos, data e etc...).



**Figura 3 - Ilustra o Caso de Uso: UseCase\_02 - Cadastro de Fornecedor.**

### 1. Finalidade/Objetivo

- Permite ao usuário cadastrar todos os fornecedores no sistema.

### 2. Ator

- Funcionário

### 3. Precondições

- O usuário deve ter efetuado login como Administrador

### 4. Fluxo principal

- O usuário seleciona a opção cadastrar fornecedor no menu principal e em seguida escolhe a opção cadastrar de fornecedor.[A1].[A2].
- O sistema solicita os dados necessários para o cadastro do fornecedor
- O usuário fornece os dados e confirma a operação.[A3]
- O sistema verifica se todos foram fornecidos e em seguida mostra uma mensagem de confirmação.

### 5. Fluxo Alternativo

**A1.** O usuário seleciona a opção Excluir Fornecedor.

- O sistema solicita o código do fornecedor a ser excluído.

**A2.** O usuário escolhe a opção Alterar fornecedor.

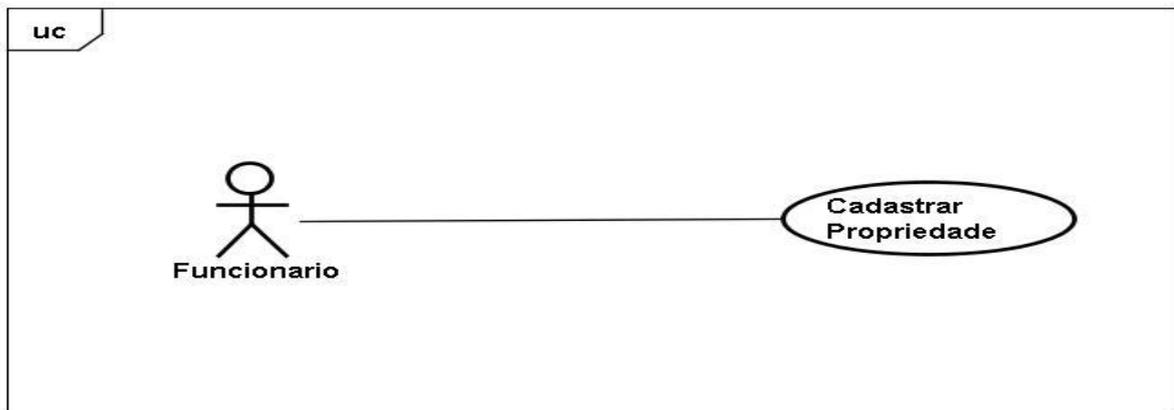
- O sistema solicita o código do fornecedor a ser alterado.

**A3.** O usuário escolhe a opção Cancelar.

- O sistema volta ao passo 4.a do Fluxo Principal.

## **6. Casos de Teste**

Verificar se todos os campos foram preenchidos corretamente (campos Numéricos, data e etc...).



**Figura 4 - Ilustra o Caso de Uso: UseCase\_03 - Cadastro de Propriedade.**

### 1. Finalidade/Objetivo

- Permite ao usuário cadastrar todas as Propriedades no sistema.

### 2. Ator

- Funcionário

### 3. Precondições

- O usuário deve ter efetuado login como Administrador

### 4. Fluxo principal

- O usuário seleciona a opção cadastrar propriedade no menu principal e em seguida escolhe a opção cadastrar propriedade.[A1].[A2].
- O sistema solicita os dados necessários para o cadastro da propriedade
- O usuário fornece os dados e confirma a operação.[A3]
- O sistema verifica se todos foram fornecidos e em seguida mostra uma mensagem de confirmação.

### 5. Fluxo Alternativo

**A1.** O usuário seleciona a opção Excluir Propriedade.

- O sistema solicita o código da propriedade a ser excluído.

**A2.** O usuário escolhe a opção Alterar Propriedade.

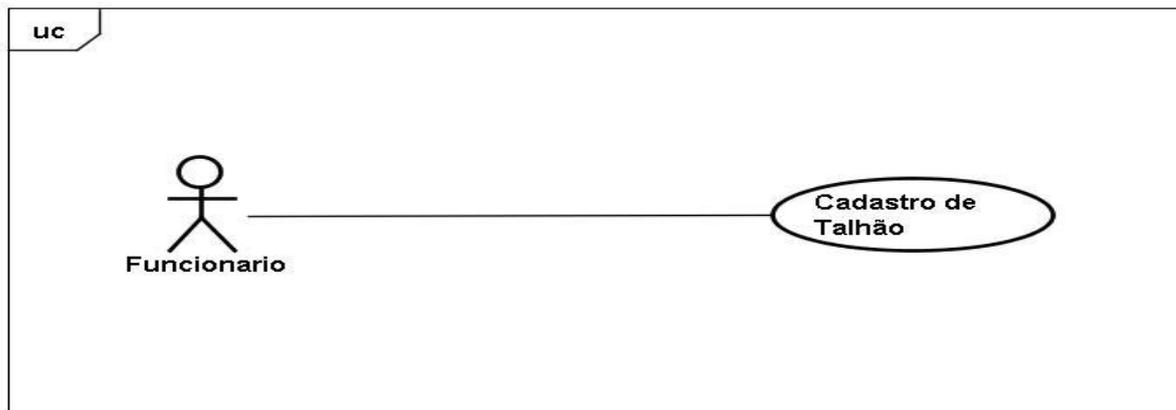
- O sistema solicita o código da propriedade a ser alterado.

**A3.** O usuário escolhe a opção Cancelar.

- O sistema volta ao passo 4.a do Fluxo Principal.

## **6. Casos de Teste**

Verificar se todos os campos foram preenchidos corretamente (campos Numéricos, data e etc...).



**Figura 5 - Ilustra o Caso de Uso: UseCase\_04 - Cadastro de Talhão.**

### 1. Finalidade/Objetivo

- Permite ao usuário cadastrar todos os talhões no sistema.

### 2. Ator

- Funcionário

### 3. Precondições

- O usuário deve ter efetuado login como Administrador

### 4. Fluxo principal

- O usuário seleciona a opção cadastrar talhão no menu principal e em seguida escolhe a opção cadastrar talhão.[A1].[A2].
- O sistema solicita os dados necessários para o cadastro do talhão
- O usuário fornece os dados e confirma a operação.[A3]
- O sistema verifica se todos foram fornecidos e em seguida mostra uma mensagem de confirmação.

### 5. Fluxo Alternativo

**A1.** O usuário seleciona a opção Excluir Talhão.

- O sistema solicita o código do talhão a ser excluído.

**A2.** O usuário escolhe a opção Alterar Talhão.

- O sistema solicita o código do talhão a ser alterado.

**A3.** O usuário escolhe a opção Cancelar.

- O sistema volta ao passo 4.a do Fluxo Principal.

## **6. Casos de Teste**

Verificar se todos os campos foram preenchidos corretamente (campos Numéricos, data e etc...).



**Figura 6 - Ilustra o Caso de Uso: UseCase\_05 - Cadastro de Atividade.**

### 1. Finalidade/Objetivo

- Permite ao usuário cadastrar todas as atividades no sistema.

### 2. Ator

- Funcionário

### 3. Precondições

- O usuário deve ter efetuado login como Administrador

### 4. Fluxo principal

- O usuário seleciona a opção cadastrar atividade no menu principal e em seguida escolhe a opção cadastrar atividade.[A1].[A2].
- O sistema solicita os dados necessários para o cadastro da atividade
- O usuário fornece os dados e confirma a operação.[A3]
- O sistema verifica se todos foram fornecidos e em seguida mostra uma mensagem de confirmação.

### 5. Fluxo Alternativo

**A1.** O usuário seleciona a opção Excluir Atividade.

- O sistema solicita o código da atividade a ser excluído.

**A2.** O usuário escolhe a opção Alterar Atividade.

- O sistema solicita o código da atividade a ser alterado.

**A3.** O usuário escolhe a opção Cancelar.

- O sistema volta ao passo 4.a do Fluxo Principal.

## **6. Casos de Teste**

Verificar se todos os campos foram preenchidos corretamente (campos Numéricos, data e etc...).



**Figura 7 - Ilustra o Caso de Uso: UseCase\_06 - Cadastro de Maquinário.**

### 1. Finalidade/Objetivo

- Permite ao usuário cadastrar todos os maquinários no sistema.

### 2. Ator

- Funcionário

### 3. Precondições

- O usuário deve ter efetuado login como Administrador

### 4. Fluxo principal

- O usuário seleciona a opção cadastrar maquinário no menu principal e em seguida escolhe a opção cadastrar maquinário.[A1].[A2].
- O sistema solicita os dados necessários para o cadastro do maquinário
- O usuário fornece os dados e confirma a operação.[A3].
- O sistema verifica se todos foram fornecidos e em seguida mostra uma mensagem de confirmação.

### 5. Fluxo Alternativo

**A1.** O usuário seleciona a opção Excluir Maquinário.

- O sistema solicita o código do maquinário a ser excluído.

**A2.** O usuário escolhe a opção Alterar Maquinário.

- O sistema solicita o código do maquinário a ser alterado.

**A3.** O usuário escolhe a opção Cancelar.

- O sistema volta ao passo 4.a do Fluxo Principal.

## **6. Casos de Teste**

Verificar se todos os campos foram preenchidos corretamente (campos Numéricos, data e etc...).



**Figura 8 - Ilustra o Caso de Uso: UseCase\_07 - Cadastro de Cultura.**

### **1. Finalidade/Objetivo**

- Permite ao usuário cadastrar todas as culturas no sistema.

### **2. Ator**

- Funcionário

### **3. Precondições**

- O usuário deve ter efetuado login como Administrador

### **4. Fluxo principal**

- a. O usuário seleciona a opção cadastrar cultura no menu principal e em seguida escolhe a opção cadastrar cultura.[A1].[A2].
- b. O sistema solicita os dados necessários para o cadastro da cultura.
- c. O usuário fornece os dados e confirma a operação.[A3].
- d. O sistema verifica se todos foram fornecidos e em seguida mostra uma mensagem de confirmação.

### **5. Fluxo Alternativo**

**A1.** O usuário seleciona a opção Excluir Cultura.

- O sistema solicita o código da cultura a ser excluída.

**A2.** O usuário escolhe a opção Alterar Cultura.

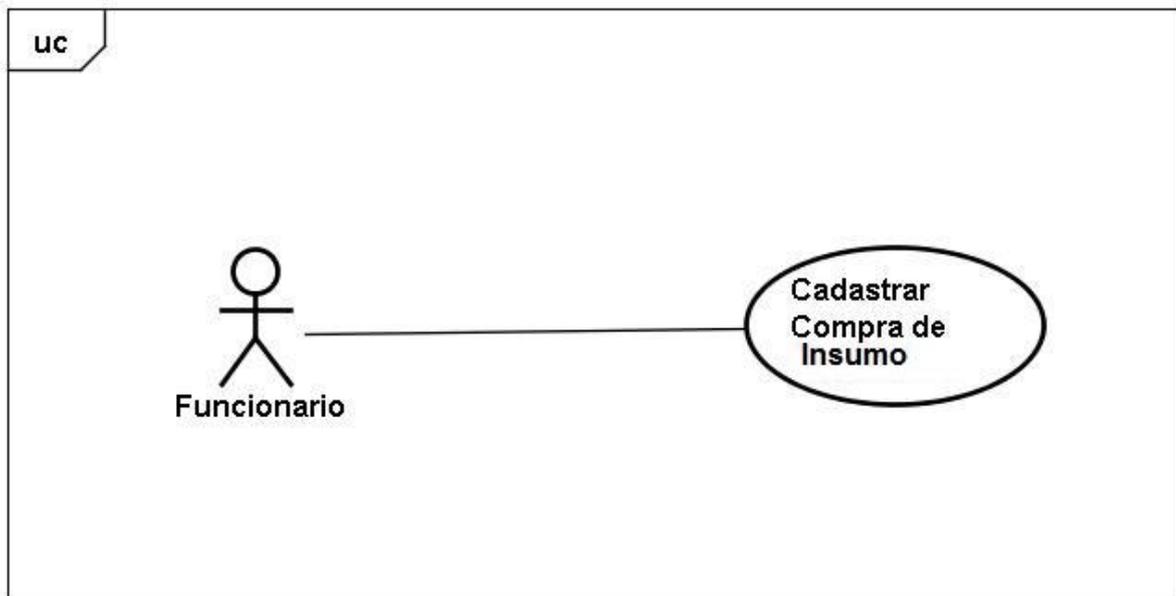
- O sistema solicita o código da cultura a ser alterada.

**A3.** O usuário escolhe a opção Cancelar.

- O sistema volta ao passo 4.a do Fluxo Principal.

## **6. Casos de Teste**

Verificar se todos os campos foram preenchidos corretamente (campos Numéricos, data e etc...).



**Figura 9 - Ilustra o Caso de Uso: UseCase\_08 - Cadastrar Compra de Insumo.**

### 1. Finalidade/Objetivo

- Permite ao usuário cadastrar todas as compras de insumos no sistema.

### 2. Ator

- Funcionário

### 3. Precondições

- O usuário deve ter efetuado login como Administrador

### 4. Fluxo principal

- a. O usuário seleciona a opção cadastrar compras de insumo no menu principal e em seguida escolhe a opção cadastrar compra de insumo.[A1],[A2].
- b. O sistema solicita os dados necessários para o cadastro da compra de insumo.
- c. O usuário fornece os dados e confirma a operação.[A3].
- d. O sistema verifica se todos foram fornecidos e em seguida mostra uma mensagem de confirmação.

### 5. Fluxo Alternativo

- A1. O usuário seleciona a opção Excluir Compra de Insumo.

- O sistema solicita o código da compra de agrotóxicos a ser excluída.

**A2.** O usuário escolhe a opção Alterar Compra de Insumo.

- O sistema solicita o código da compra de insumo a ser alterada.

**A3.** O usuário escolhe a opção Cancelar.

- O sistema volta ao passo 4.a do Fluxo Principal.

## **6. Casos de Teste**

Verificar se todos os campos foram preenchidos corretamente (campos Numéricos, data e etc...).



**Figura 9 - Ilustra o Caso de Uso: UseCase\_11 - Cadastrar Compra de Peças.**

### 1. Finalidade/Objetivo

- Permite ao usuário cadastrar todas as compras de peças no sistema.

### 2. Ator

- Funcionário

### 3. Precondições

- O usuário deve ter efetuado login como Administrador

### 4. Fluxo principal

- O usuário seleciona a opção cadastrar compra de peças no menu principal e em seguida escolhe a opção cadastrar compra de peças.[A1].[A2].
- O sistema solicita os dados necessários para o cadastro da compra de peças.
- O usuário fornece os dados e confirma a operação.[A3].
- O sistema verifica se todos foram fornecidos e em seguida mostra uma mensagem de confirmação.

### 5. Fluxo Alternativo

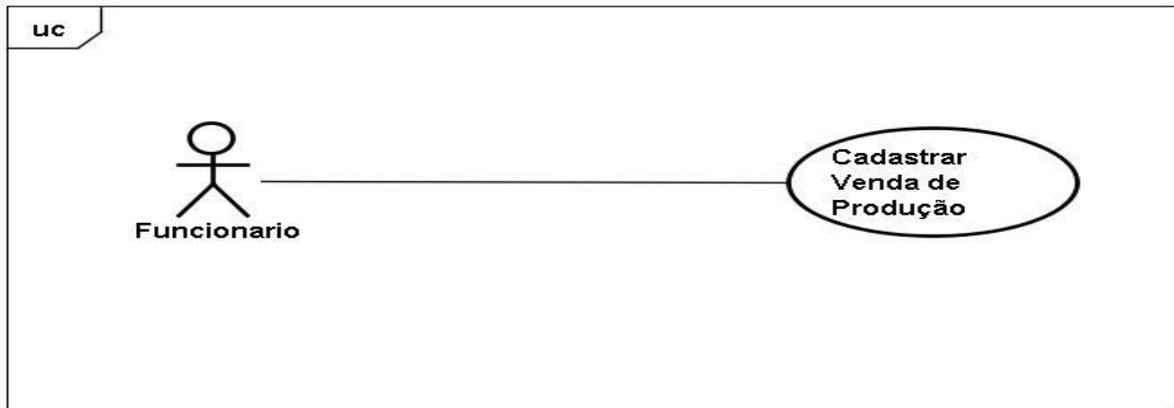
- O usuário seleciona a opção Excluir Compra de Peças.
  - O sistema solicita o código da compra de peças a ser excluída.
- O usuário escolhe a opção Alterar Compra de Peças.
  - O sistema solicita o código da compra de peças a ser alterada.

**A3.** O usuário escolhe a opção Cancelar.

- O sistema volta ao passo 4.a do Fluxo Principal.

## **6. Casos de Teste**

Verificar se todos os campos foram preenchidos corretamente (campos Numéricos, data e etc...).



**Figura 10 - Ilustra o Caso de Uso: UseCase\_12 - Cadastrar Venda de Produção.**

### 1. Finalidade/Objetivo

- Permite ao usuário cadastrar todas as vendas no sistema.

### 2. Ator

- Funcionário

### 3. Precondições

- O usuário deve ter efetuado login como Administrador

### 4. Fluxo principal

- O usuário seleciona a opção cadastrar venda de produção no menu principal e em seguida escolhe a opção cadastrar venda de produção.[A1].[A2].
- O sistema solicita os dados necessários para o cadastro da venda de produção.
- O usuário fornece os dados e confirma a operação.[A3].
- O sistema verifica se todos foram fornecidos e em seguida mostra uma mensagem de confirmação.

### 5. Fluxo Alternativo

- O usuário seleciona a opção Excluir Venda de Produção.
- O sistema solicita o código da venda de produção a ser excluída.
- O usuário escolhe a opção Alterar Venda de Produção.

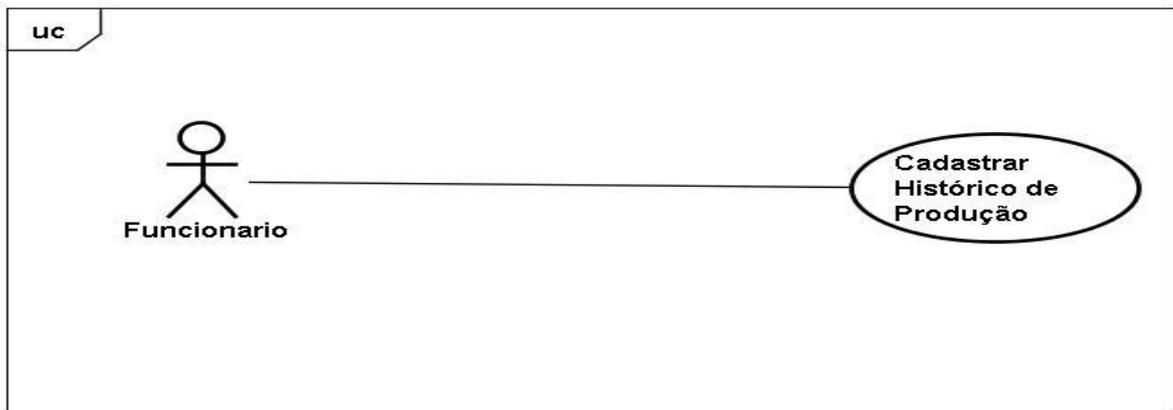
- O sistema solicita o código da venda de produção a ser alterada.

**A3.** O usuário escolhe a opção Cancelar.

- O sistema volta ao passo 4.a do Fluxo Principal.

## **6. Casos de Teste**

Verificar se todos os campos foram preenchidos corretamente (campos Numéricos, data e etc..).



**Figura 11 - Ilustra o Caso de Uso: UseCase\_13 - Cadastrar Histórico de Produção.**

### 1. Finalidade/Objetivo

- Permite ao usuário cadastrar o histórico de produção no sistema.

### 2. Ator

- Funcionário

### 3. Precondições

- O usuário deve ter efetuado login como Administrador

### 4. Fluxo principal

- a. O usuário seleciona a opção cadastrar histórico de produção no menu principal e em seguida escolhe a opção cadastrar histórico de produção.[A1],[A2].
- b. O sistema solicita os dados necessários para o cadastro do histórico de produção.
- c. O usuário fornece os dados e confirma a operação.
- d. O sistema verifica se todos foram fornecidos e em seguida mostra uma mensagem de confirmação.

### 5. Fluxo Alternativo

- A1.** O usuário seleciona a opção Excluir Histórico de Produção.
- O sistema solicita o código do histórico de produção a ser excluído.
- A2.** O usuário escolhe a opção Alterar Histórico de Produção.

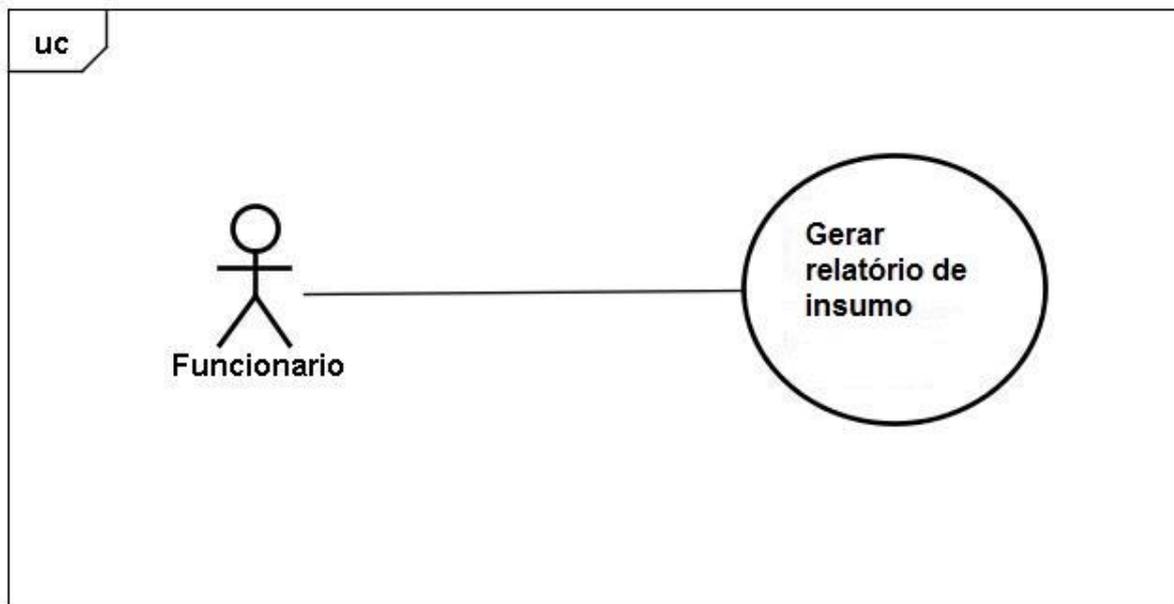
- O sistema solicita o código do histórico de produção a ser alterado.

**A3.** O usuário escolhe a opção Cancelar.

- O sistema volta ao passo 4.a do Fluxo Principal.

## **6. Casos de Teste**

Verificar se todos os campos foram preenchidos corretamente (campos Numéricos, data e etc...).



**Figura 12 - Ilustra o Caso de Uso: UseCase\_14 - Gerar Relatório de Insumo.**

### **1. Finalidade/Objetivo**

- Permite ao usuário gerar relatórios de insumos.

### **2. Ator**

- Funcionário

### **3. Precondições**

- O usuário deve ter efetuado login como Administrador.

### **4. Fluxo principal**

- a. O usuário seleciona a opção gerar relatórios no menu principal e em seguida escolhe a opção relatório de insumos gastos por aplicação.
- b. O sistema solicita os dados da aplicação desejada.
- c. O usuário fornece os dados e confirma a operação.[A1].
- d. O sistema emite o relatório na tela para o usuário.

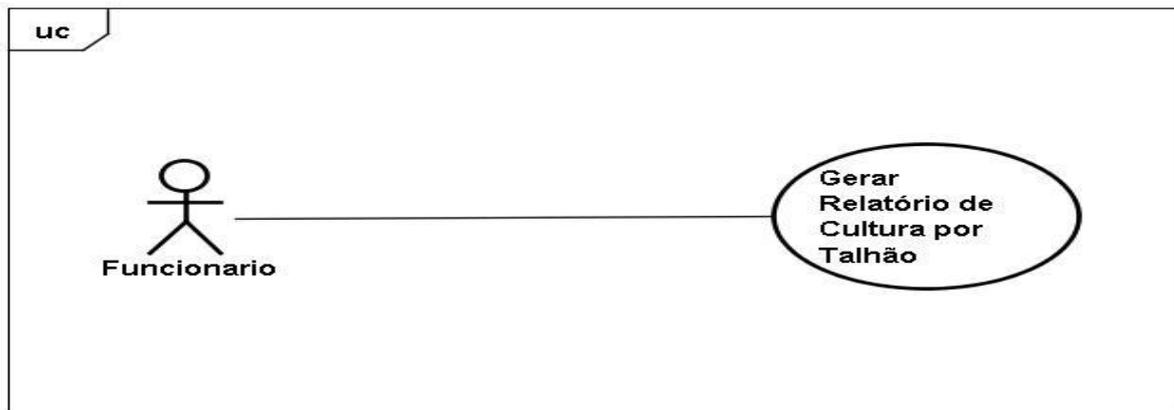
### **5. Fluxo Alternativo**

**A1.** O usuário escolhe a opção Cancelar.

- O sistema volta ao passo 4.a do Fluxo Principal.

## **6. Casos de Teste**

Verificar se os dados fornecidos pelo usuário existem.



**Figura 13 - Ilustra o Caso de Uso: UseCase\_18 - Gerar Relatório de Cultura por talhão.**

### **1. Finalidade/Objetivo**

- Permite ao usuário gerar relatórios de cultura por talhão.

### **2. Ator**

- Funcionário

### **3. Precondições**

- O usuário deve ter efetuado login como Administrador

### **4. Fluxo principal**

- a. O usuário seleciona a opção gerar relatórios no menu principal e em seguida escolhe a opção relatório de cultura por talhão.
- b. O sistema solicita os dados da aplicação desejada.
- c. O usuário fornece os dados e confirma a operação.[A1].
- d. O sistema emite o relatório na tela para o usuário.

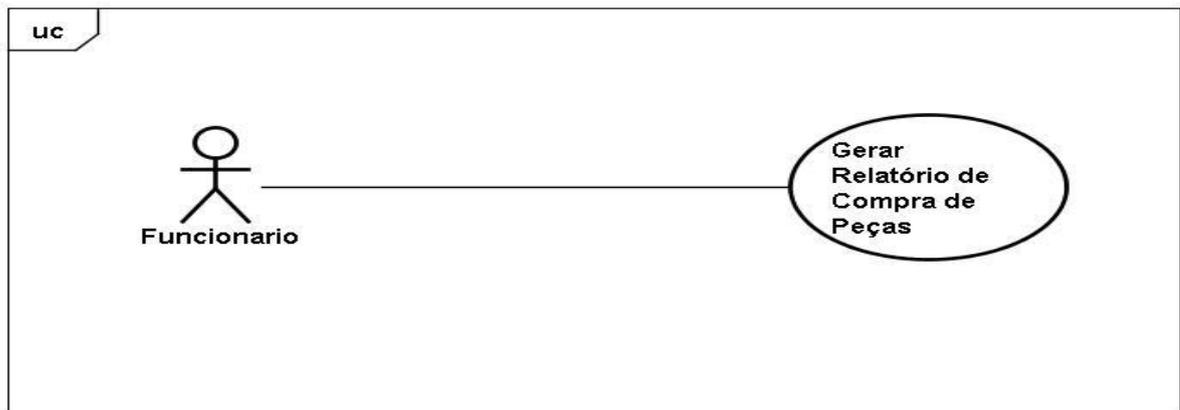
### **5. Fluxo Alternativo**

**A1.** O usuário escolhe a opção Cancelar.

- O sistema volta ao passo 4.a do Fluxo Principal.

### **6. Casos de Teste**

Verificar se os dados fornecidos pelo usuário existem.



**Figura 14 - Ilustra o Caso de Uso: UseCase\_20 - Gerar Relatório de Compra de Peças.**

### 1. Finalidade/Objetivo

- Permite ao usuário gerar relatórios de compra de peças.

### 2. Ator

- Funcionário

### 3. Precondições

- O usuário deve ter efetuado login como Administrador

### 4. Fluxo principal

- a. O usuário seleciona a opção gerar relatórios no menu principal e em seguida escolhe a opção relatório de compra de peças.
- b. O sistema solicita os dados da aplicação desejada.
- c. O usuário fornece os dados e confirma a operação.[A1].
- d. O sistema emite o relatório na tela para o usuário.

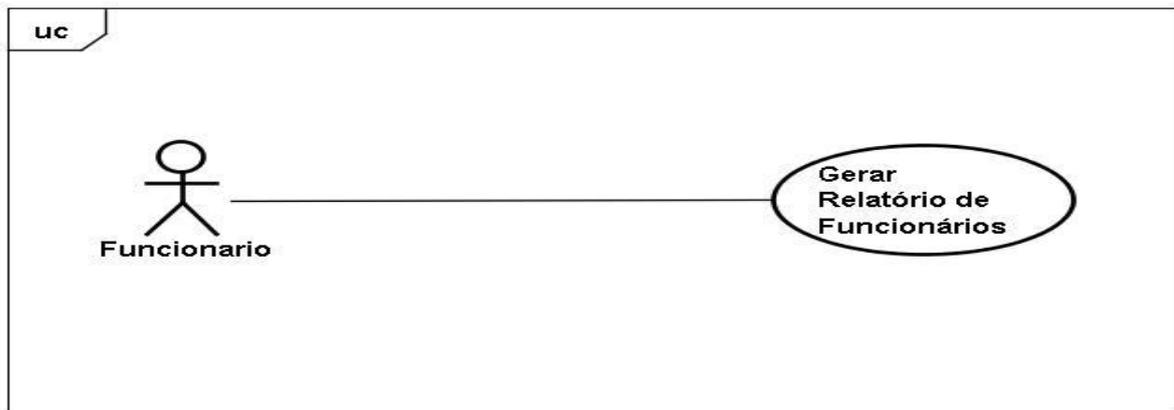
### 5. Fluxo Alternativo

**A1.** O usuário escolhe a opção Cancelar.

- O sistema volta ao passo 4.a do Fluxo Principal.

### 6. Casos de Teste

Verificar se os dados fornecidos pelo usuário existem.



**Figura 16 - Ilustra o Caso de Uso: UseCase\_21 - Gerar Relatório de Funcionários.**

### **1. Finalidade/Objetivo**

- Permite ao usuário gerar relatórios de funcionarios.

### **2. Ator**

- Funcionário

### **3. Precondições**

- O usuário deve ter efetuado login como Administrador

### **4. Fluxo principal**

- a. O usuário seleciona a opção gerar relatórios no menu principal e em seguida escolhe a opção relatório de funcionarios.
- b. O sistema solicita os dados da aplicação desejada.
- c. O usuário fornece os dados e confirma a operação.[A1].
- d. O sistema emite o relatório na tela para o usuário.

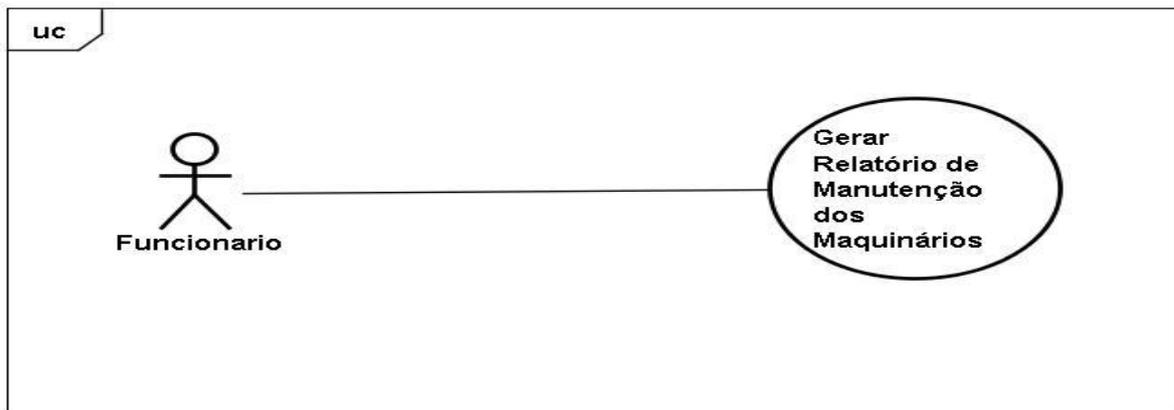
### **5. Fluxo Alternativo**

**A1.** O usuário escolhe a opção Cancelar.

- O sistema volta ao passo 4.a do Fluxo Principal.

### **6. Casos de Teste**

Verificar se os dados fornecidos pelo usuário existem.



**Figura 17 - Ilustra o Caso de Uso: UseCase\_22 - Gerar Relatório de Manutenção dos Maquinários**

### 1. Finalidade/Objetivo

- Permite ao usuário gerar relatórios de manutenção dos maquinários.

### 2. Ator

- Funcionário

### 3. Precondições

- O usuário deve ter efetuado login como Administrador

### 4. Fluxo principal

- a. O usuário seleciona a opção gerar relatórios no menu principal e em seguida escolhe a opção relatório de manutenção dos maquinários.
- b. O sistema solicita os dados da aplicação desejada.
- c. O usuário fornece os dados e confirma a operação.[A1].
- d. O sistema emite o relatório na tela para o usuário.

### 5. Fluxo Alternativo

**A1.** O usuário escolhe a opção Cancelar.

- O sistema volta ao passo 4.a do Fluxo Principal.

### 6. Casos de Teste

Verificar se os dados fornecidos pelo usuário existem.



**Figura 19 - Ilustra o Caso de Uso: UseCase\_23 - Gerar Relatório de Produção por Safra.**

### 1. Finalidade/Objetivo

- Permite ao usuário gerar relatórios de produção por safra.

### 2. Ator

- Funcionário

### 3. Precondições

- O usuário deve ter efetuado login como Administrador

### 4. Fluxo principal

- a. O usuário seleciona a opção gerar relatórios no menu principal e em seguida escolhe a opção relatório de produção por safra.
- b. O sistema solicita os dados da aplicação desejada.
- c. O usuário fornece os dados e confirma a operação.[A1]
- d. O sistema emite o relatório na tela para o usuário.

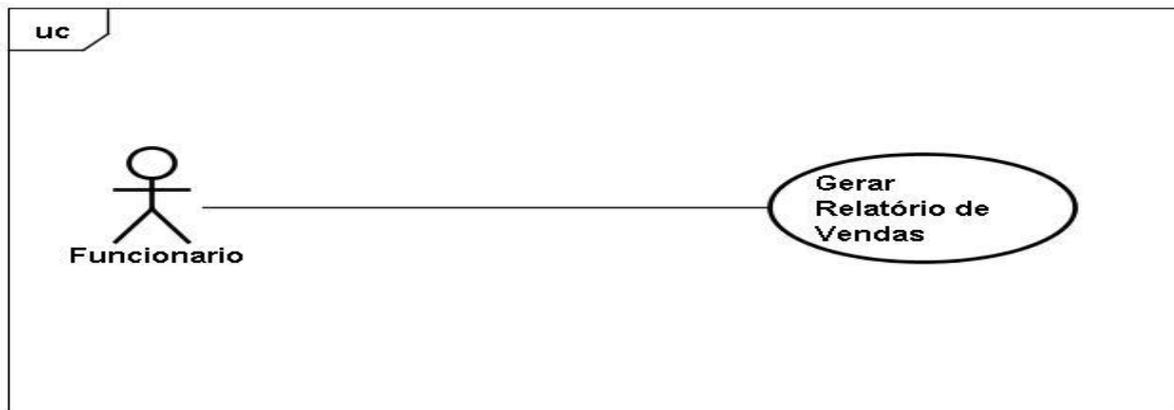
### 5. Fluxo Alternativo

**A1.** O usuário escolhe a opção Cancelar.

- O sistema volta ao passo 4.a do Fluxo Principal.

### 6. Casos de Teste

Verificar se os dados fornecidos pelo usuário existem.



**Figura 20 - Ilustra o Caso de Uso: UseCase\_24 - Gerar Relatório de Vendas.**

### 1. Finalidade/Objetivo

- Permite ao usuário gerar relatórios de vendas.

### 2. Ator

- Funcionário

### 3. Precondições

- O usuário deve ter efetuado login como Administrador

### 4. Fluxo principal

- O usuário seleciona a opção gerar relatórios no menu principal e em seguida escolhe a opção relatório de vendas.
- O sistema solicita os dados da aplicação desejada.
- O usuário fornece os dados e confirma a operação.[A1].
- O sistema emite o relatório na tela para o usuário.

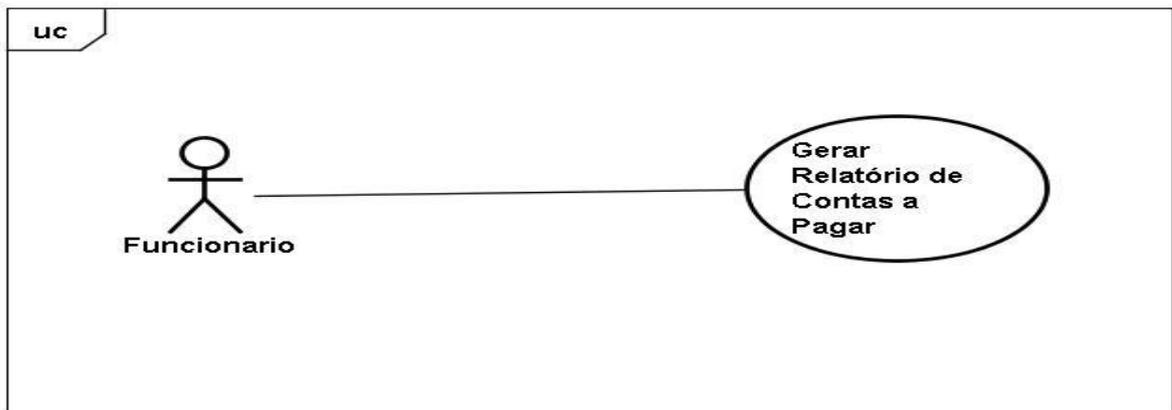
### 5. Fluxo Alternativo

**A1.** O usuário escolhe a opção Cancelar.

- O sistema volta ao passo 4.a do Fluxo Principal.

### 6. Casos de Teste

Verificar se os dados fornecidos pelo usuário existem.



**Figura 21 - Ilustra o Caso de Uso: UseCase\_25 - Gerar Relatório de Contas a Pagar.**

### 1. Finalidade/Objetivo

- Permite ao usuário gerar relatórios de contas a pagar.

### 2. Ator

- Funcionário

### 3. Precondições

- O usuário deve ter efetuado login como Administrador

### 4. Fluxo principal

- O usuário seleciona a opção gerar relatórios no menu principal e em seguida escolhe a opção relatório de contas a pagar.
- O sistema solicita os dados da aplicação desejada.
- O usuário fornece os dados e confirma a operação.[A1].
- O sistema emite o relatório na tela para o usuário.

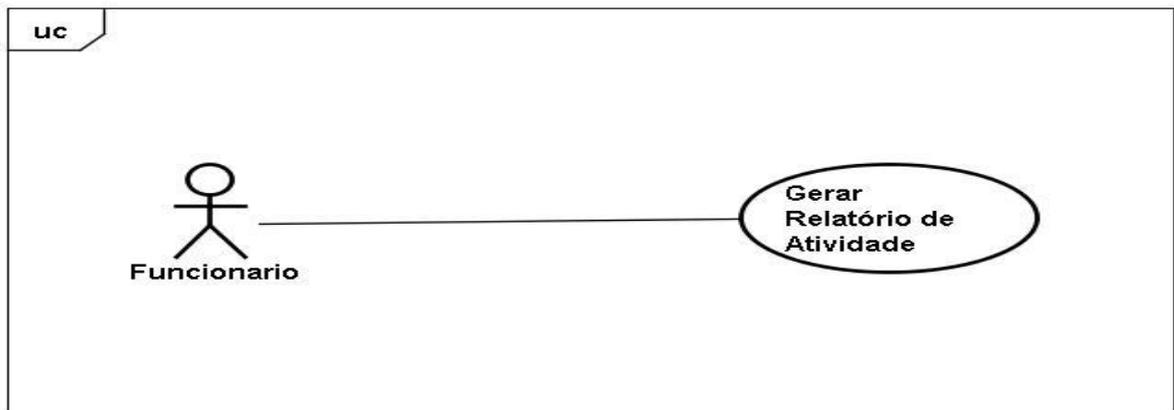
### 5. Fluxo Alternativo

**A1.** O usuário escolhe a opção Cancelar.

- O sistema volta ao passo 4.a do Fluxo Principal.

### 6. Casos de Teste

Verificar se os dados fornecidos pelo usuário existem.



**Figura 28 - Ilustra o Caso de Uso: UseCase\_26 - Gerar Relatório de Atividades.**

### 1. Finalidade/Objetivo

- Permite ao usuário gerar relatórios de atividades.

### 2. Ator

- Funcionário

### 3. Precondições

- O usuário deve ter efetuado login como Administrador

### 4. Fluxo principal

- O usuário seleciona a opção gerar relatórios no menu principal e em seguida escolhe a opção relatório de atividades.
- O sistema solicita os dados da aplicação desejada.
- O usuário fornece os dados e confirma a operação.[A1]
- O sistema emite o relatório na tela para o usuário.

### 5. Fluxo Alternativo

**A1.** O usuário escolhe a opção Cancelar.

- O sistema volta ao passo 4.a do Fluxo Principal.

### 6. Casos de Teste

Verificar se os dados fornecidos pelo usuário existem.



**Figura 29 - Ilustra o Caso de Uso: UseCase\_27 - Gerar Gráfico de Produção por Safra.**

### 1. Finalidade/Objetivo

- Permite ao usuário gerar gráfico de produção por safra.

### 2. Ator

- Funcionário

### 3. Precondições

- O usuário deve ter efetuado login como Administrador

### 4. Fluxo principal

- O usuário seleciona a opção gerar relatórios no menu principal e em seguida escolhe a opção gráfico de produção por safra.
- O sistema solicita a safra desejada.
- O usuário fornece os dados e confirma a operação.[A1].
- O sistema gera o gráfico na tela para o usuário.

### 5. Fluxo Alternativo

**A1.** O usuário escolhe a opção Cancelar.

- O sistema volta ao menu principal.

### 6. Casos de Teste

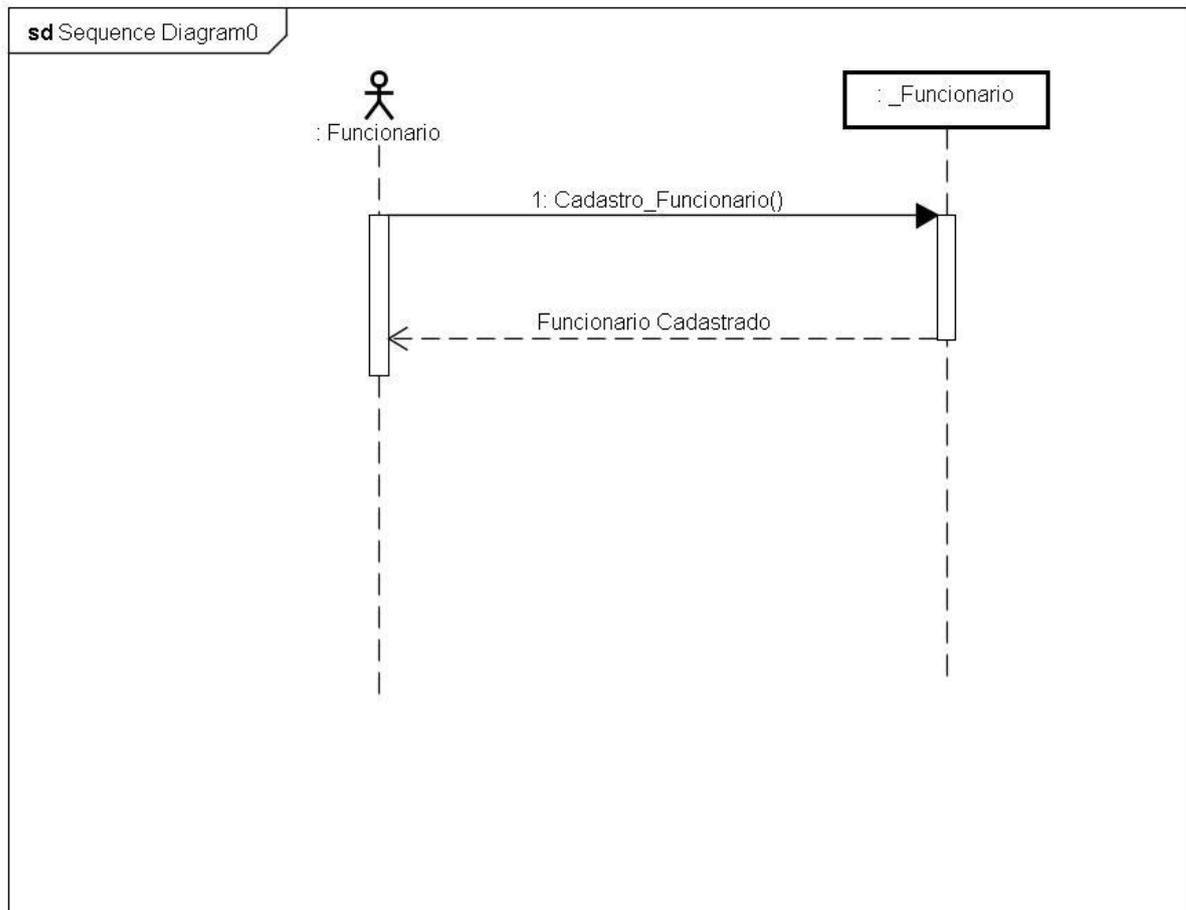
Verificar se os dados fornecidos pelo usuário existem.

### 3.5. DIAGRAMA DE SEQUÊNCIA

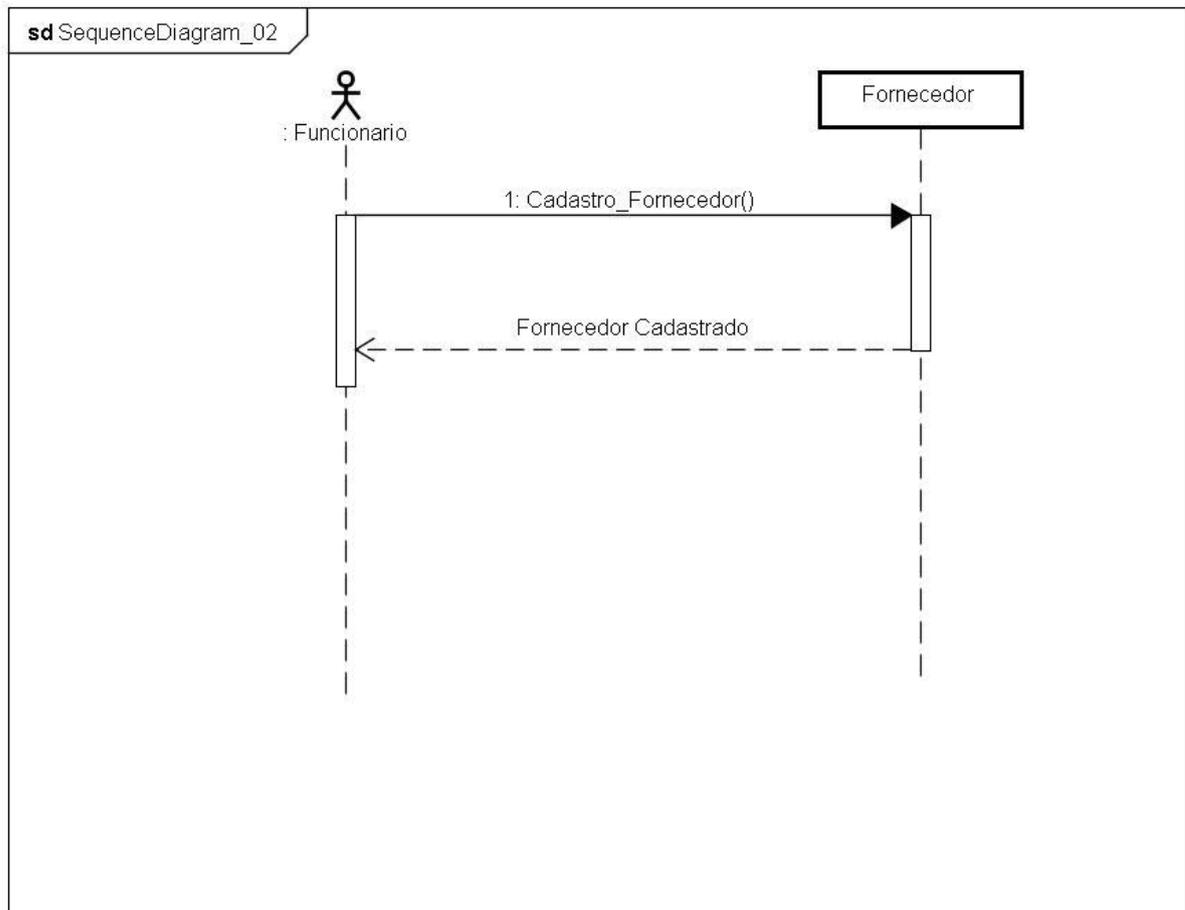
Os detalhes de um Caso de Uso podem ser mostrados em um diagrama de sequência.

Segundo C. Lee e Tepfenhart (2001).

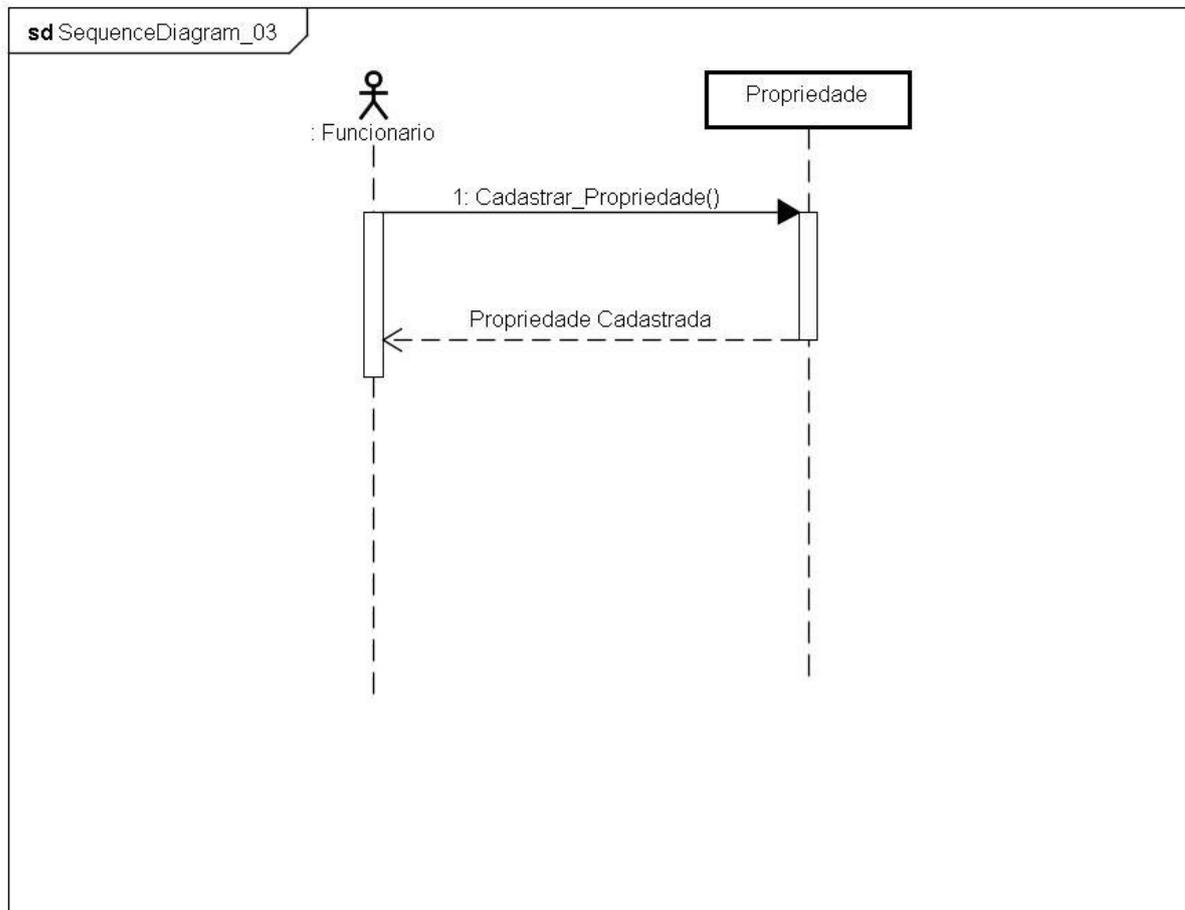
Um diagrama de sequência mostra a ordem na qual mensagens são trocadas entre os atores e o sistema. O diagrama de sequência tem os participantes representados por caixas retangulares. Partindo dessas caixas retangulares, há uma linha estendida tracejada e vertical. As trocas de mensagens entre participantes são ilustradas como setas dirigidas e são identificadas pela mensagem que está sendo comunicada. A sequência de é lida de cima para baixo. Assim, o tempo transcorre de cima para baixo. (C. Lee e Tepfenhart, 2001, 52).



**Figura 30 – Diagrama de Sequência: DiagramSequence\_01 – Cadastrar Funcionário.**



**Figura 31 – Diagrama de Sequência: DiagramSequence\_02 – Cadastrar Fornecedor.**



**Figura 32 – Diagrama de Sequência: DiagramSequence\_03 – Cadastrar Propriedade.**

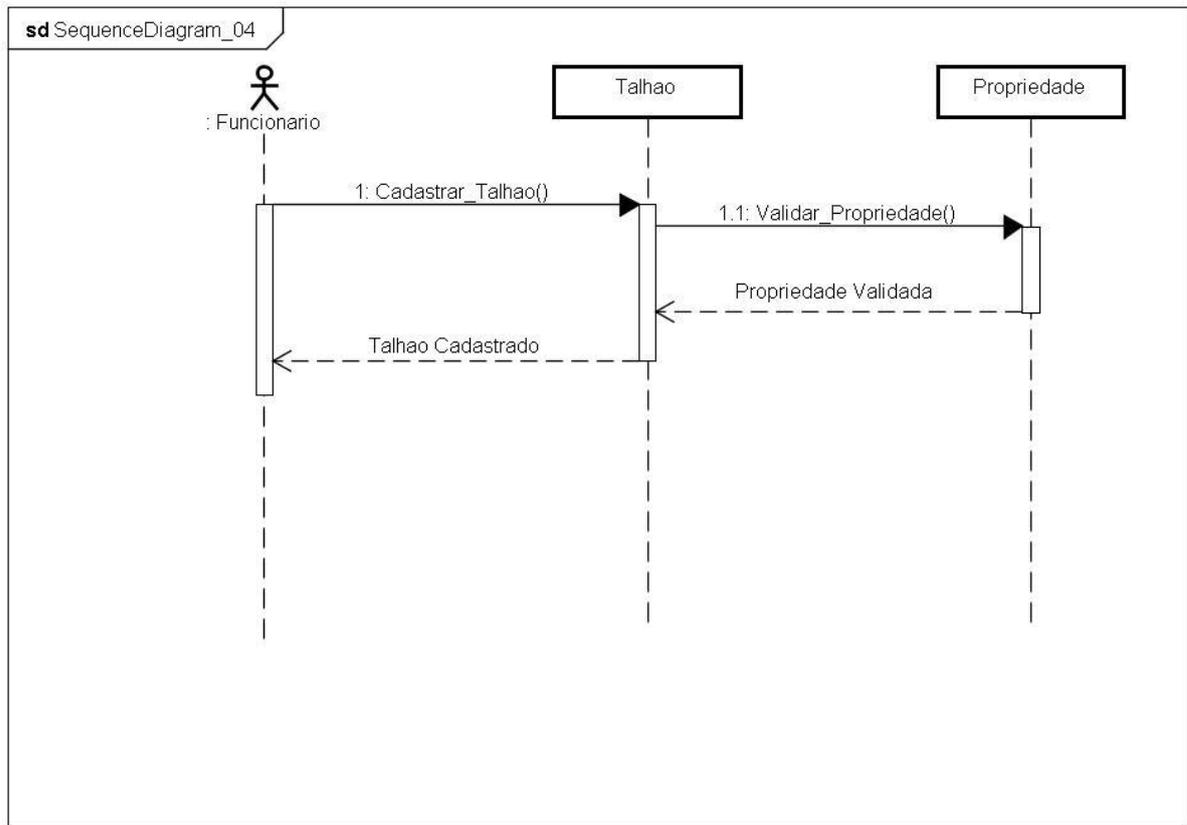
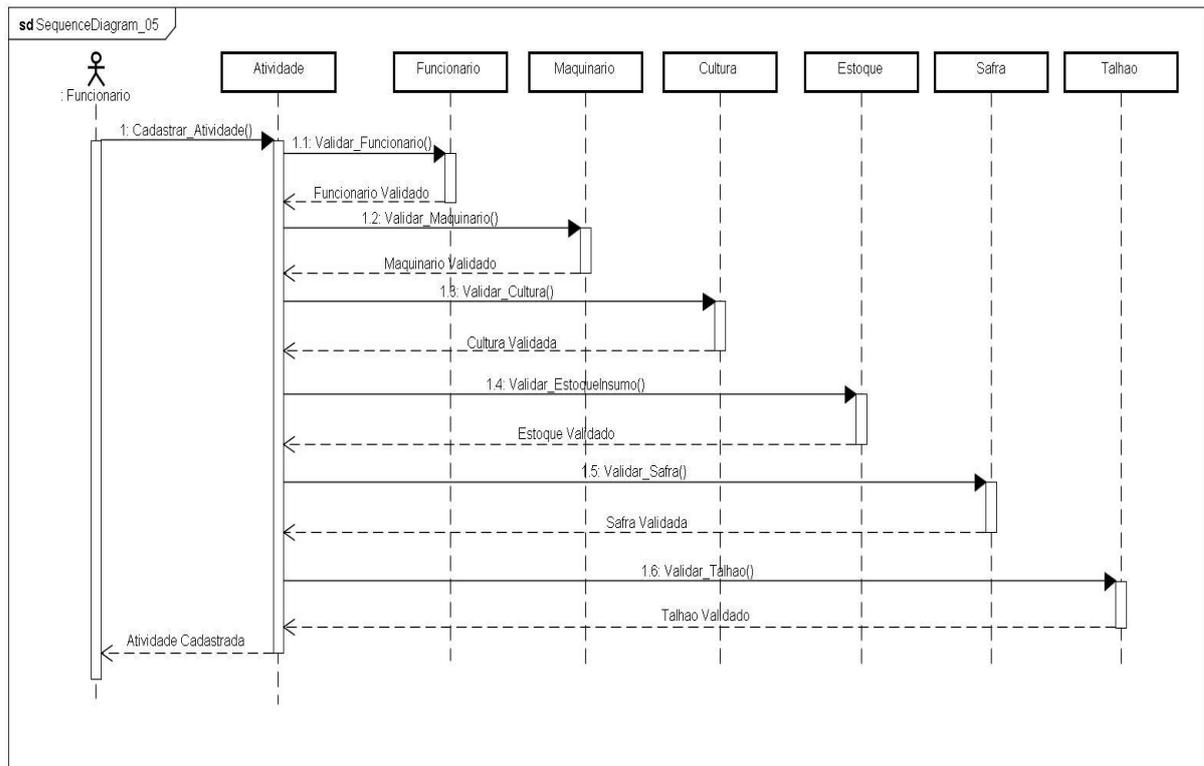
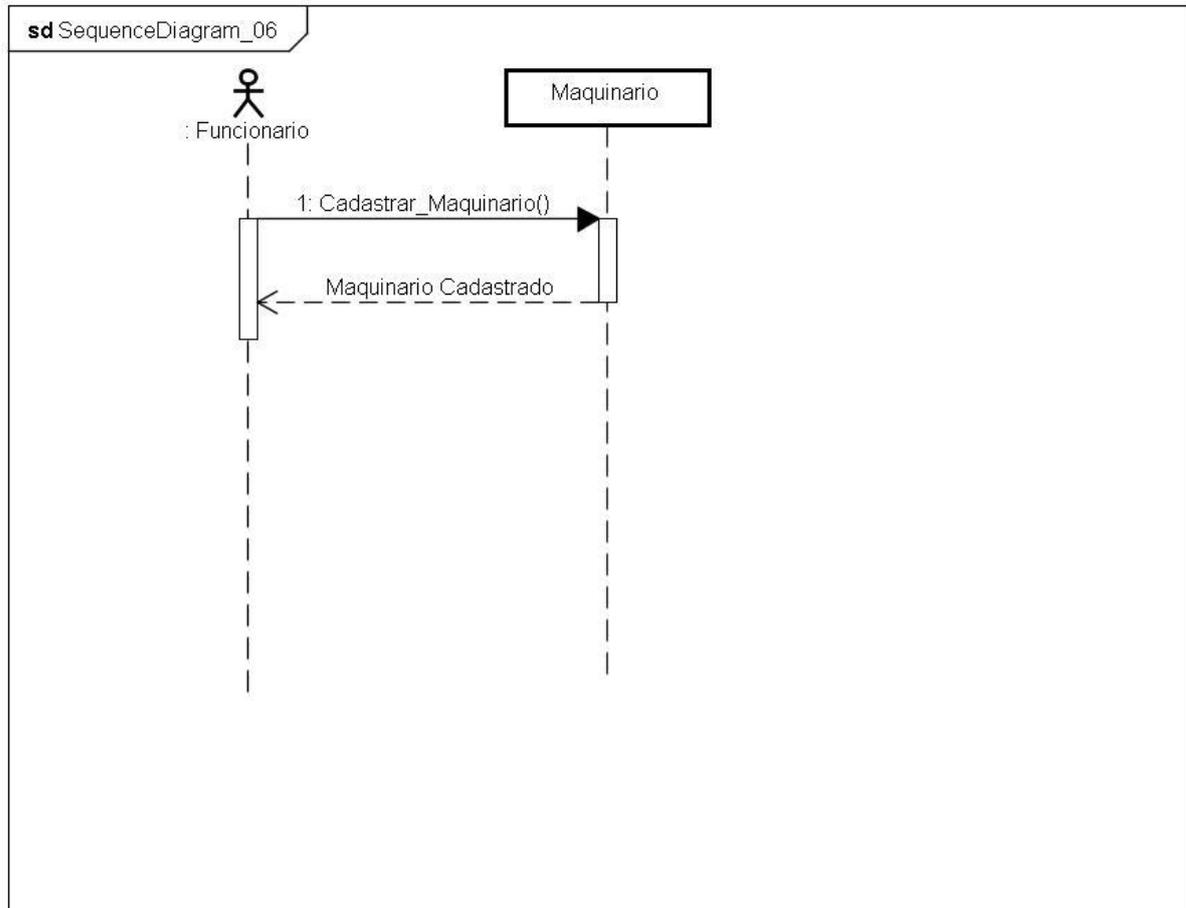


Figura 33 – Diagrama de Sequência: DiagramSequence\_04 – Cadastrar Talhão.



**Figura 34 – Diagrama de Sequência: DiagramSequence\_05 – Cadastrar Atividade.**



**Figura 35 – Diagrama de Sequência: DiagramSequence\_06 – Cadastrar Maquinário.**

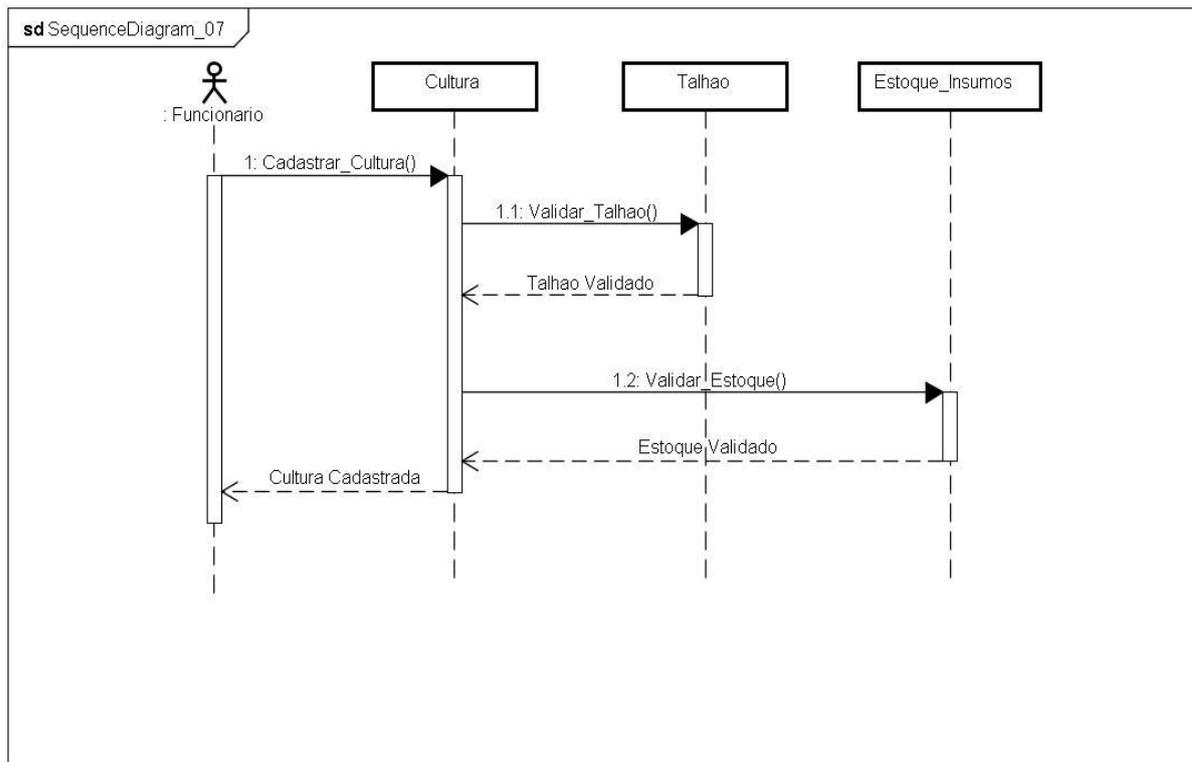
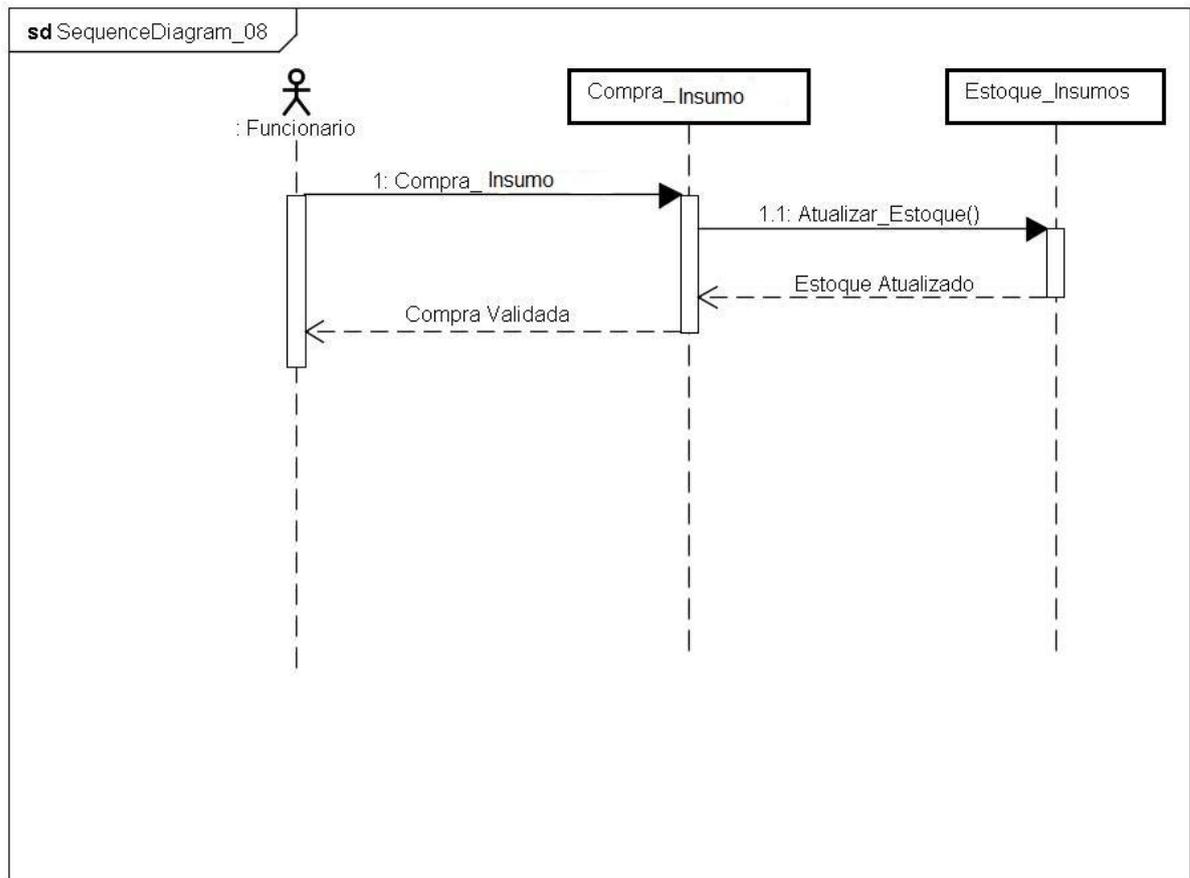


Figura 36 – Diagrama de Sequência: DiagramSequence\_07 – Cadastrar Cultura.



**Figura 37 – Diagrama de Sequência: DiagramSequence\_08 – Cadastrar Insumos.**

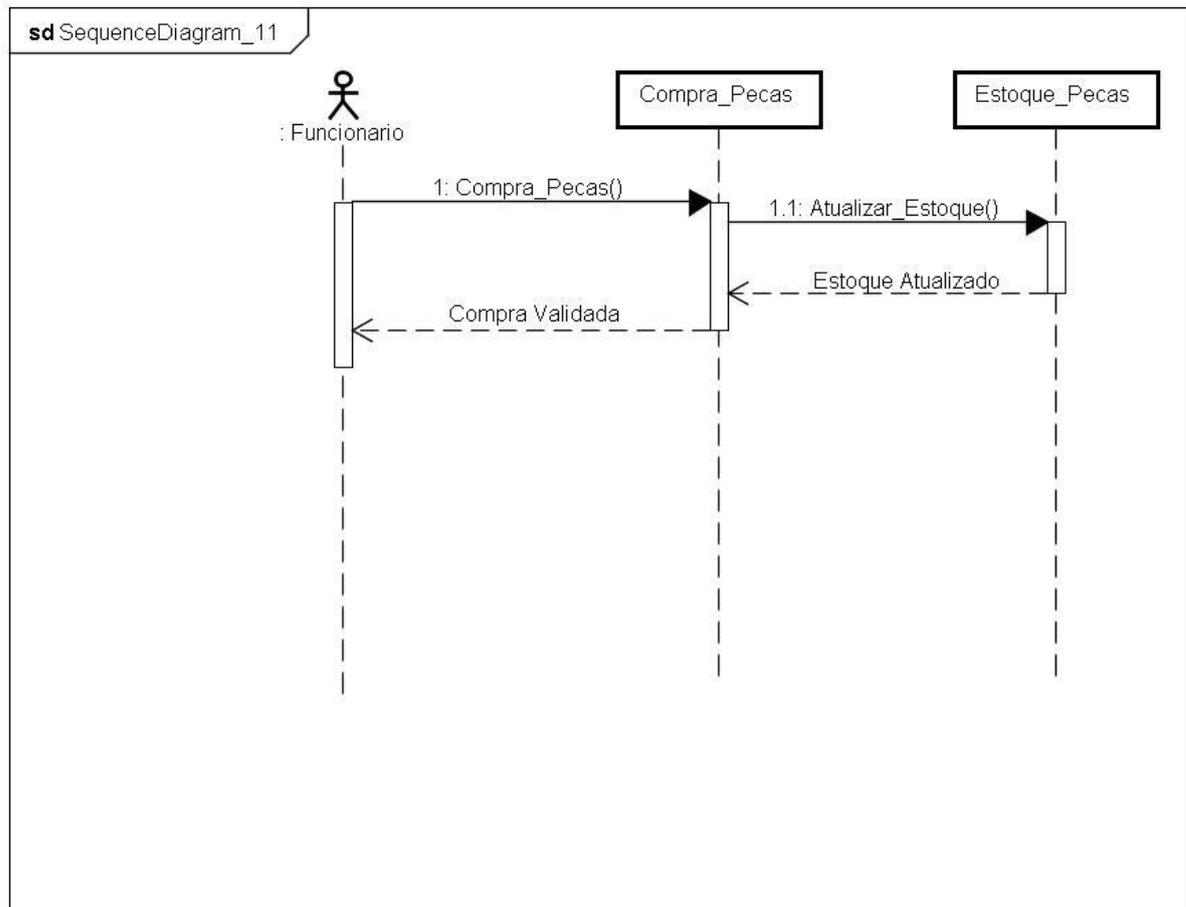


Figura 38 – Diagrama de Sequência: DiagramSequence\_11 – Compra Peças.

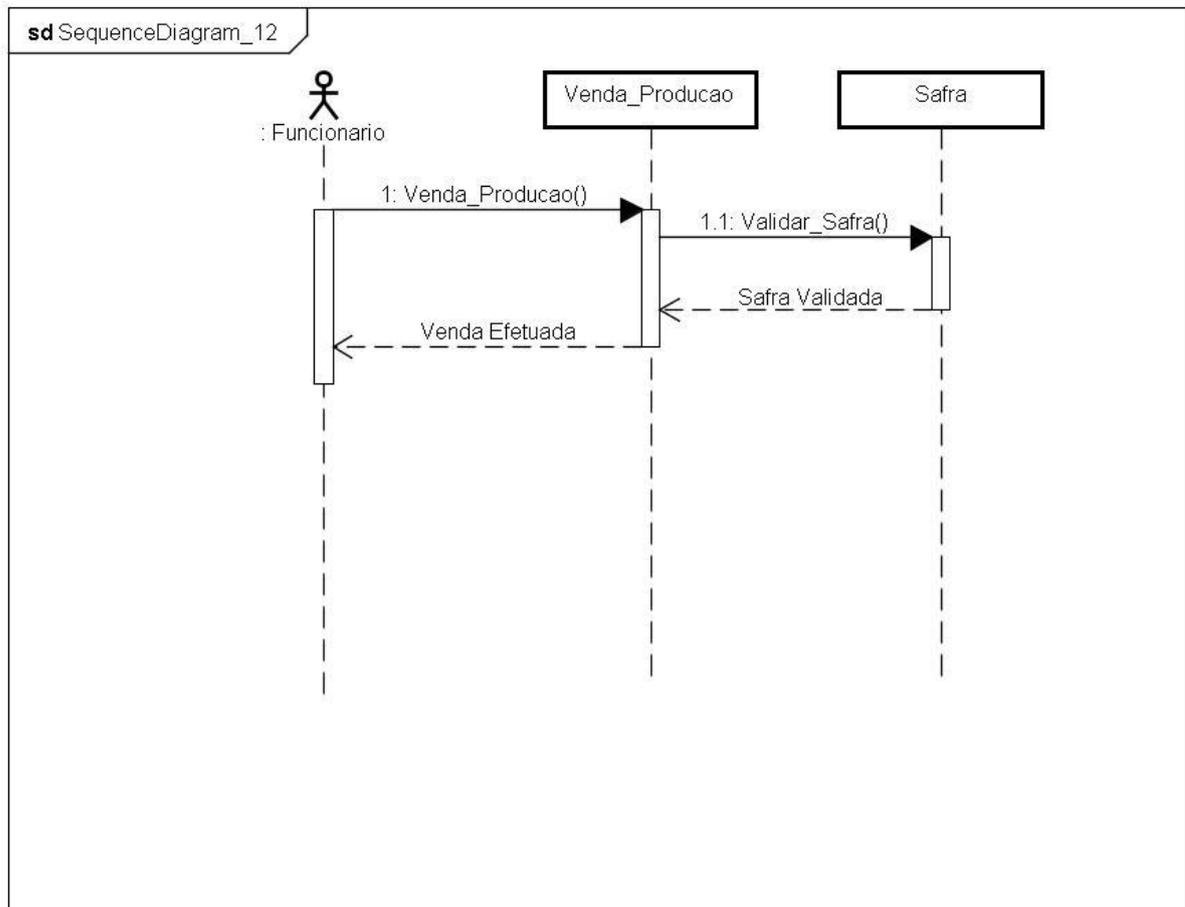


Figura 39 – Diagrama de Sequência: DiagramSequence\_12 – Venda Produção.

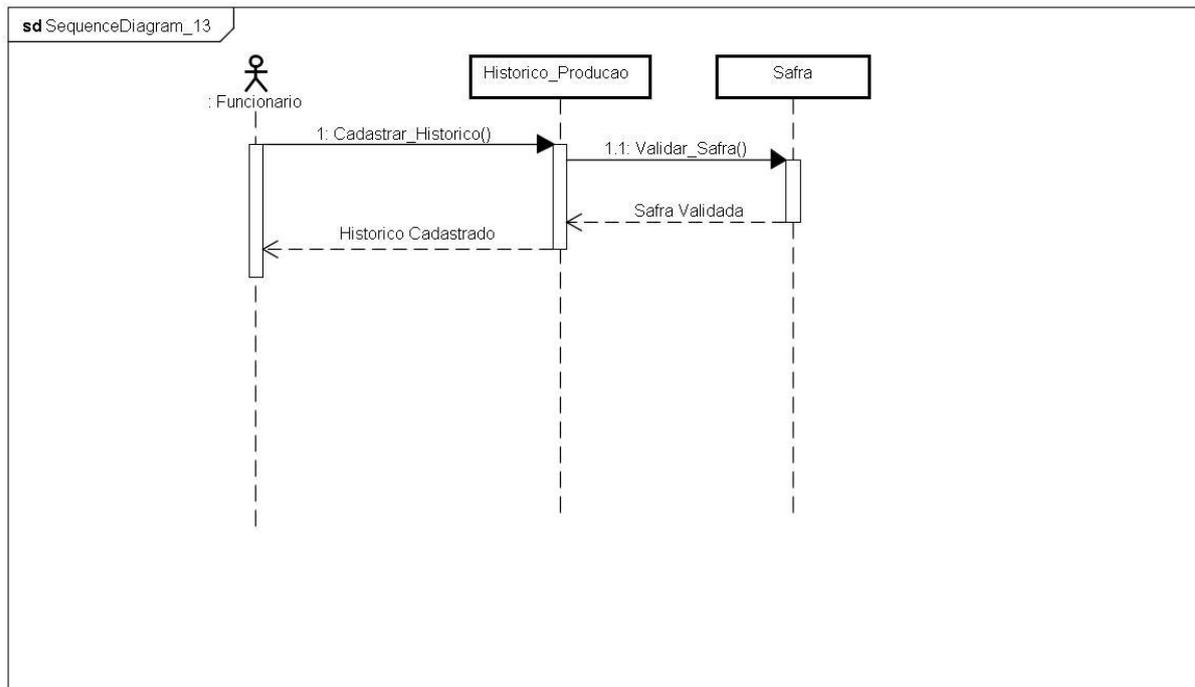
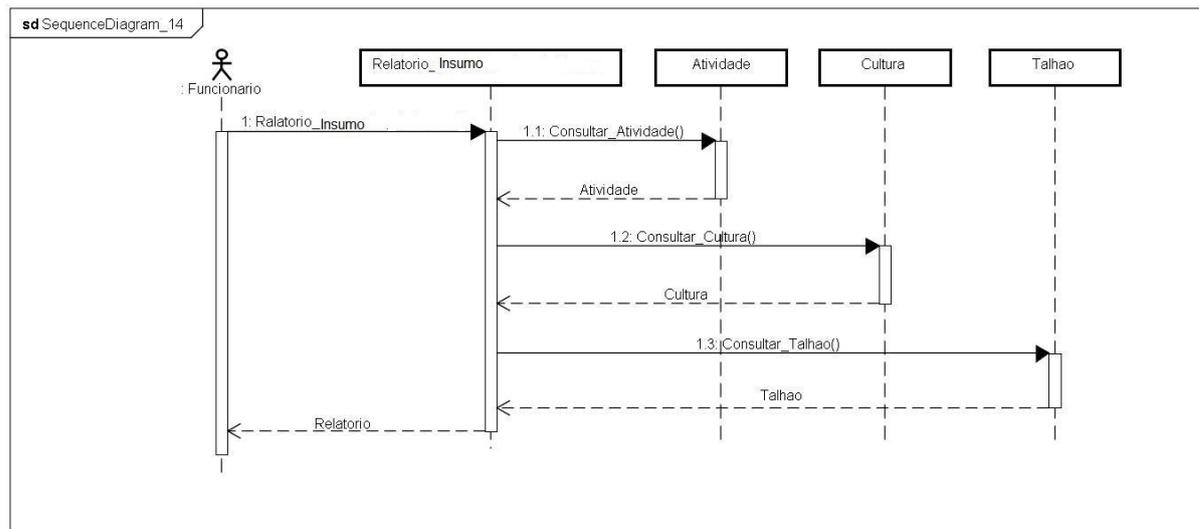
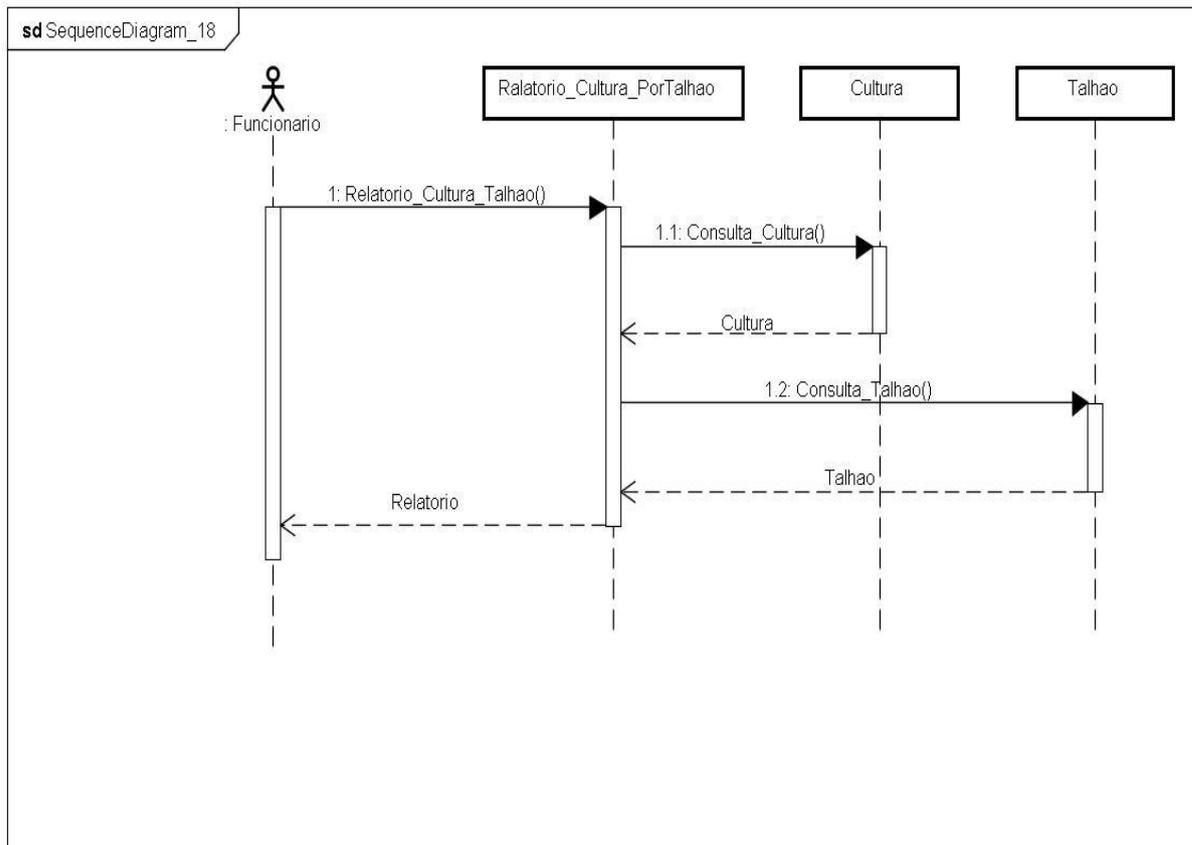


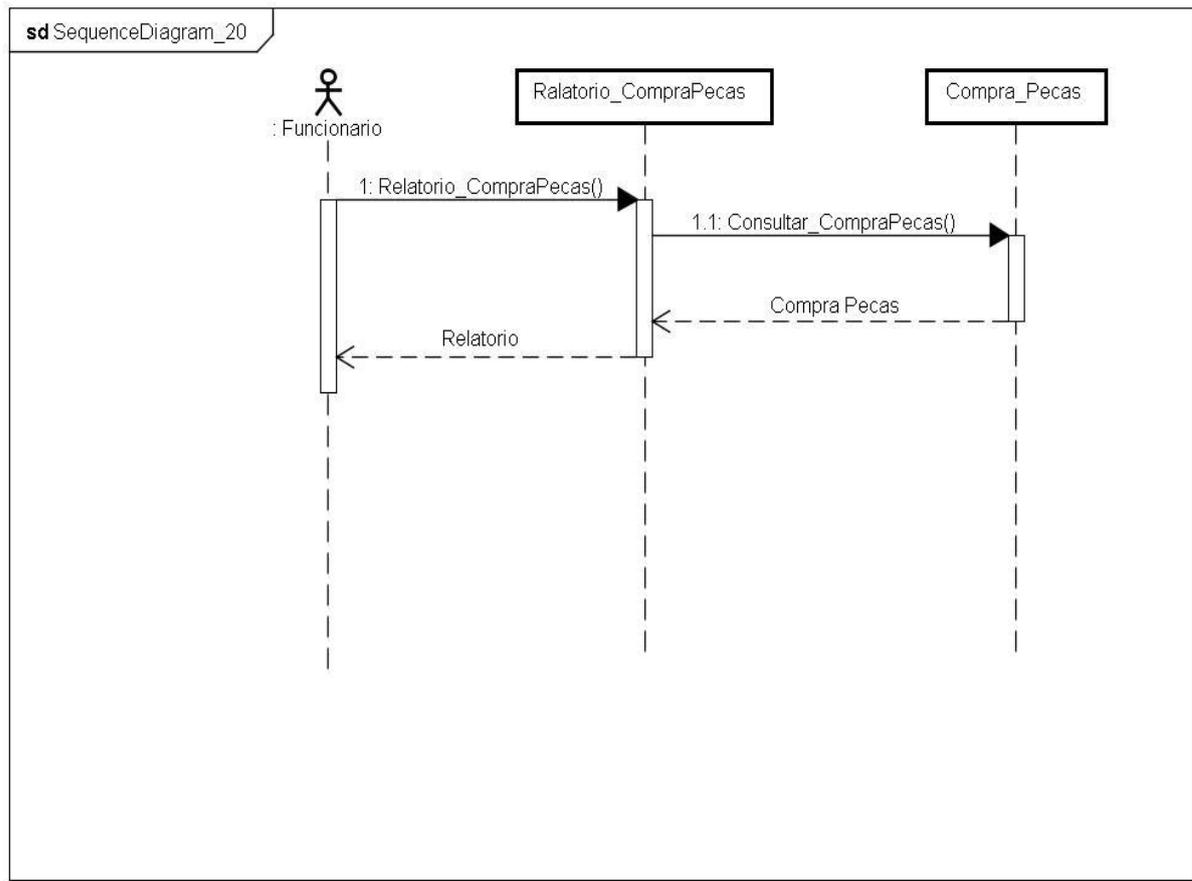
Figura 40 – Diagrama de Sequência: DiagramSequence\_13 – Venda Histórico.



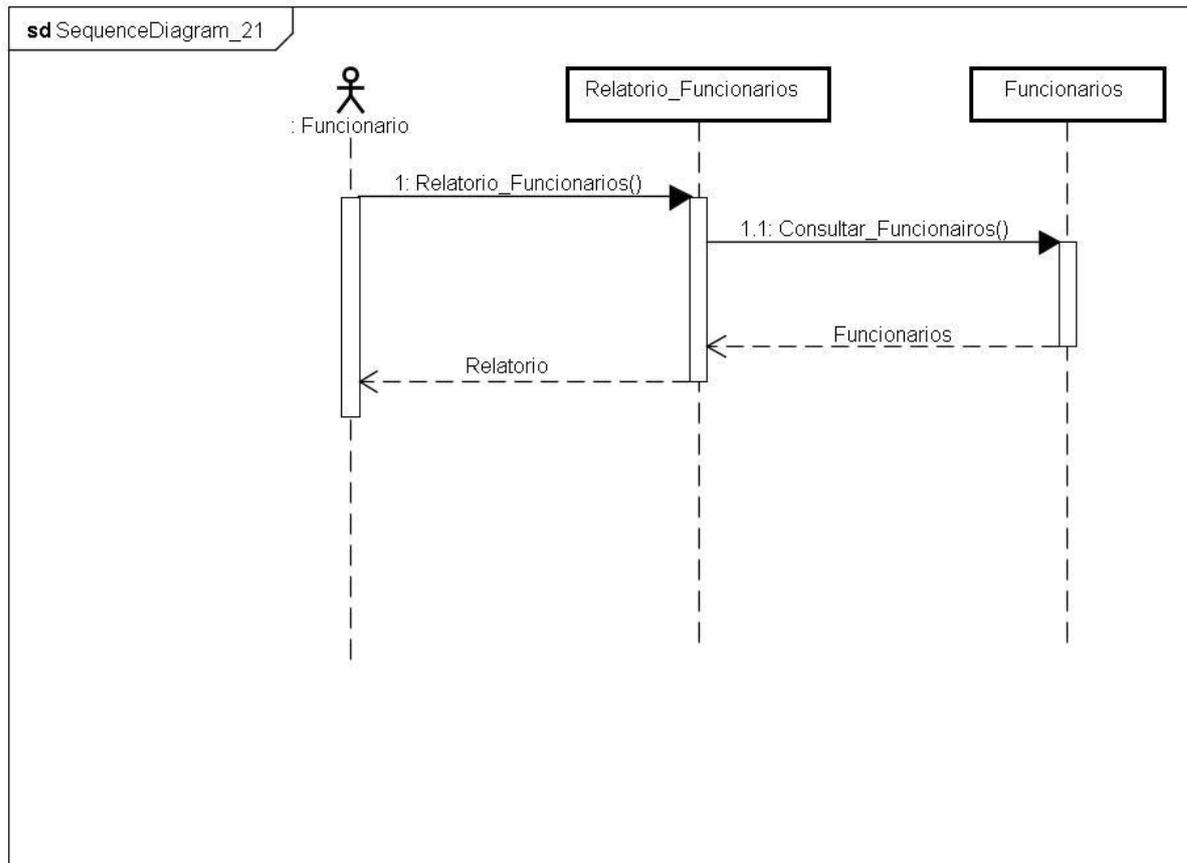
**Figura 41 – Diagrama de Sequência: DiagramSequence\_14 – Relatório de insumos.**



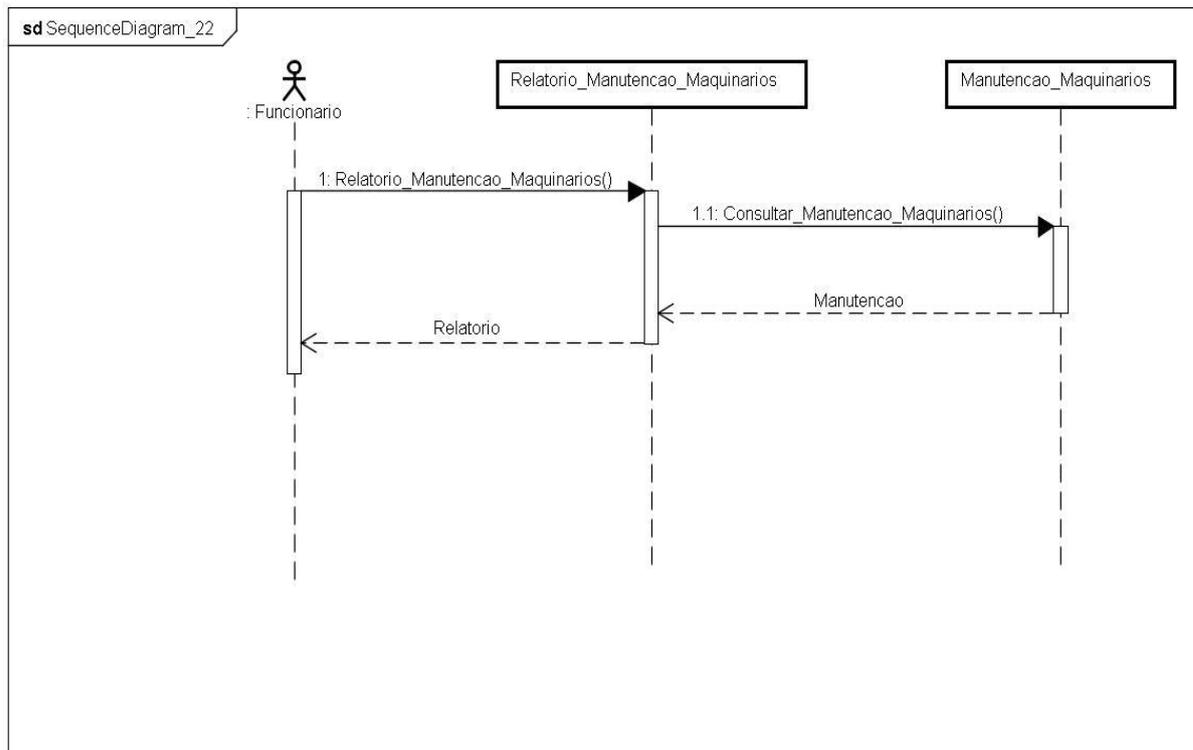
**Figura 42 – Diagrama de Sequência: DiagramSequence\_18 – Relatório de cultura por talhão.**



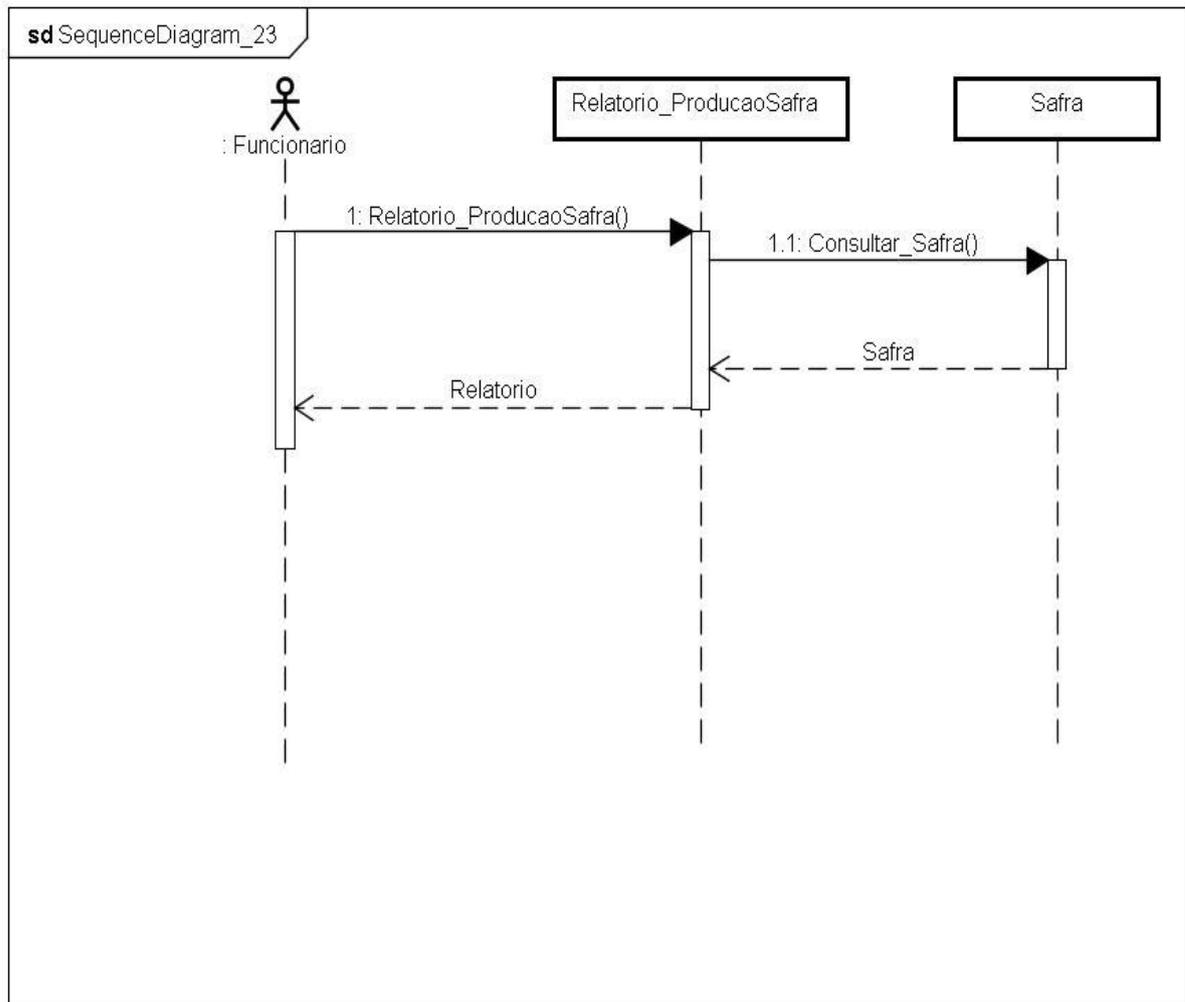
**Figura 43 – Diagrama de Sequência: DiagramSequence\_20 – Relatório de Compra de Peças.**



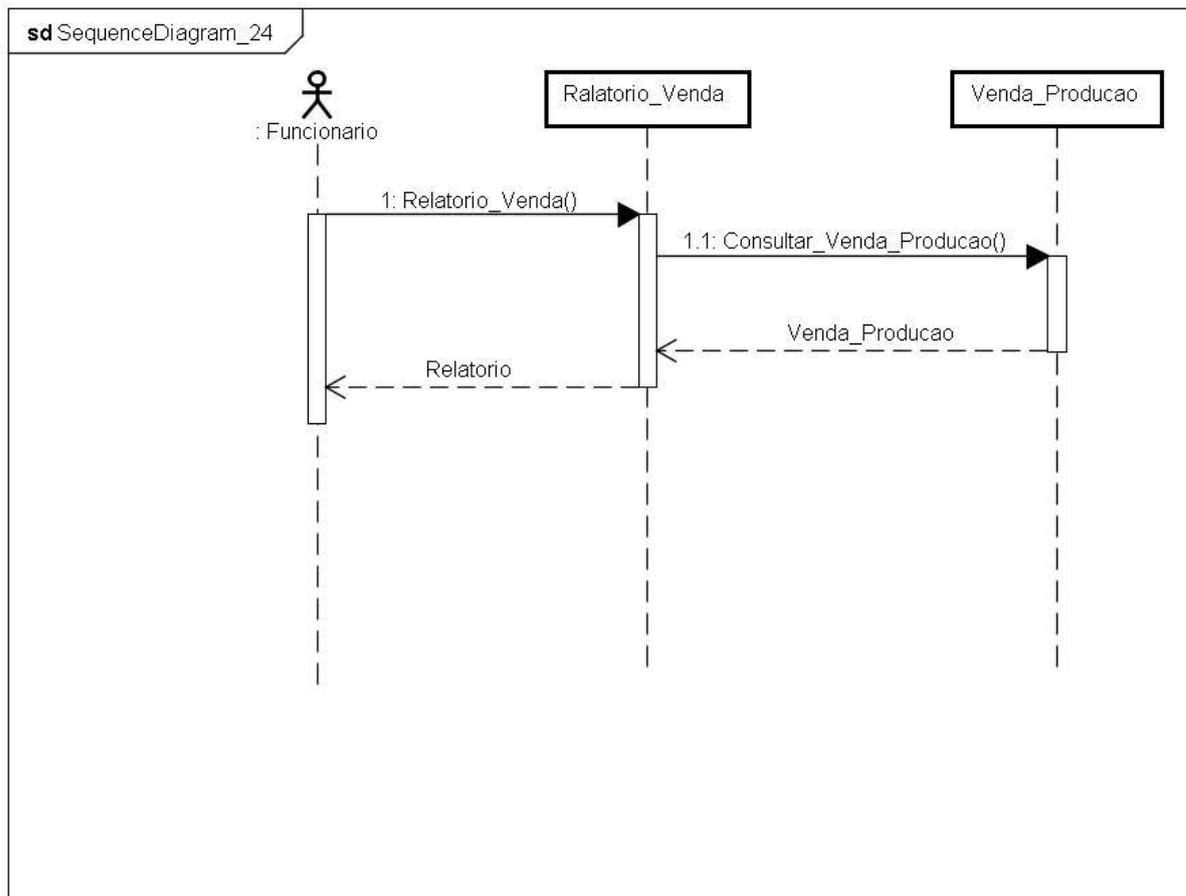
**Figura 44 – Diagrama de Sequência: DiagramSequence\_21 – Relatório de Funcionarios.**



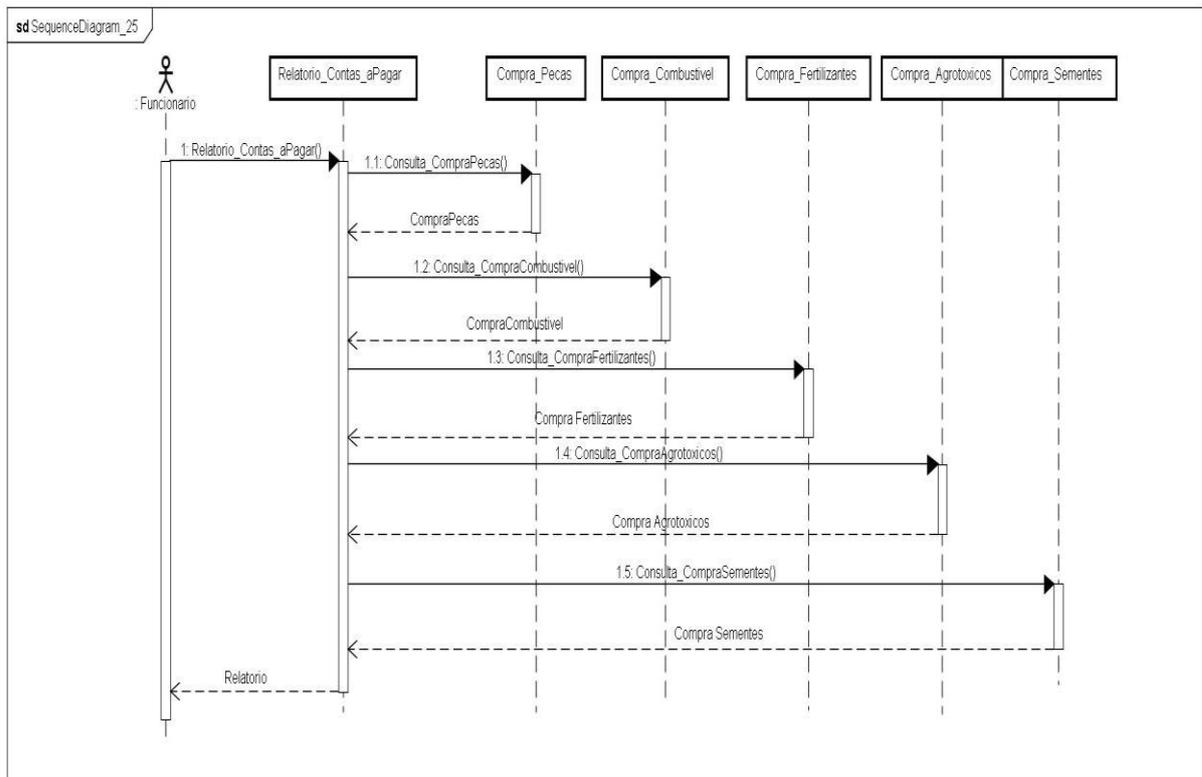
**Figura 45 – Diagrama de Sequência: DiagramSequence\_22 – Relatório de Manutenção dos Maquinários.**



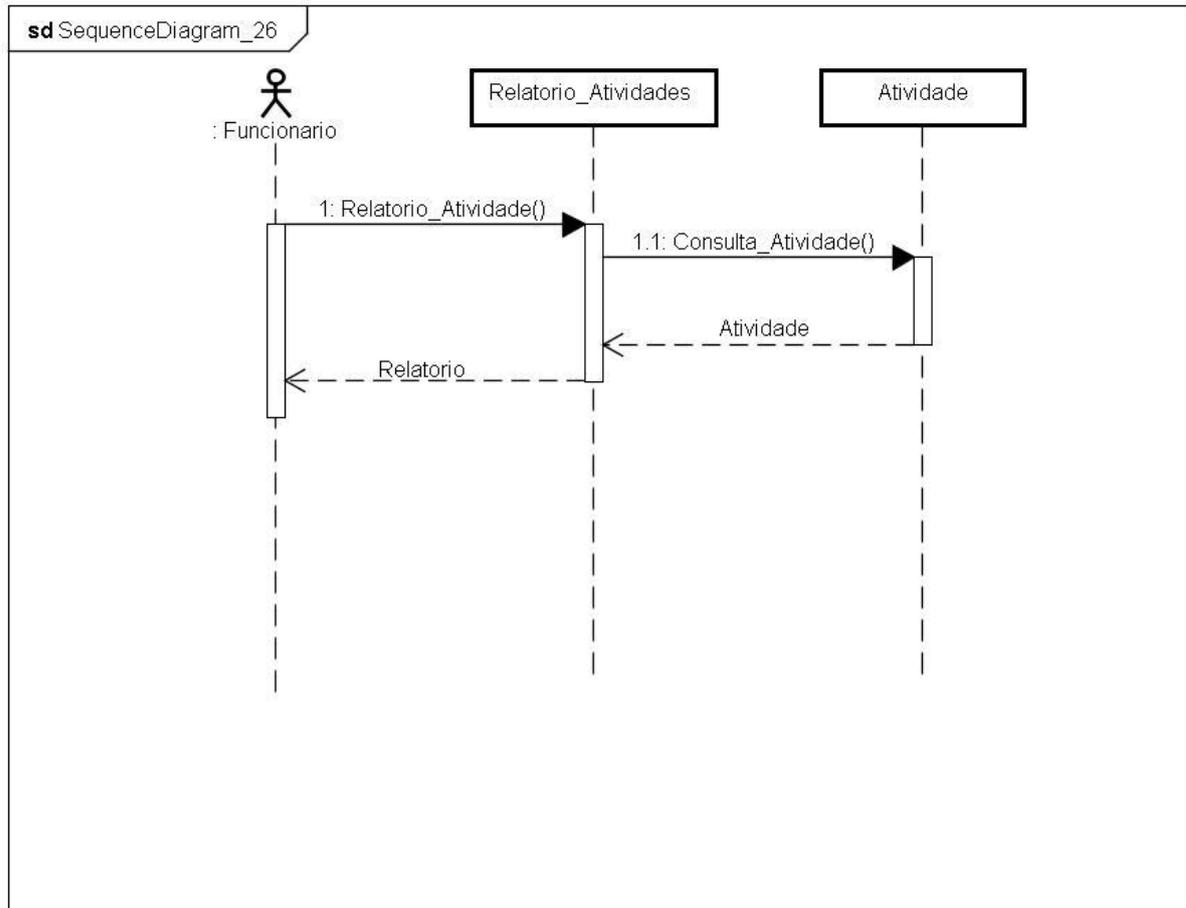
**Figura 46 – Diagrama de Sequência: DiagramSequence\_23 – Relatório de Produção por Safra.**



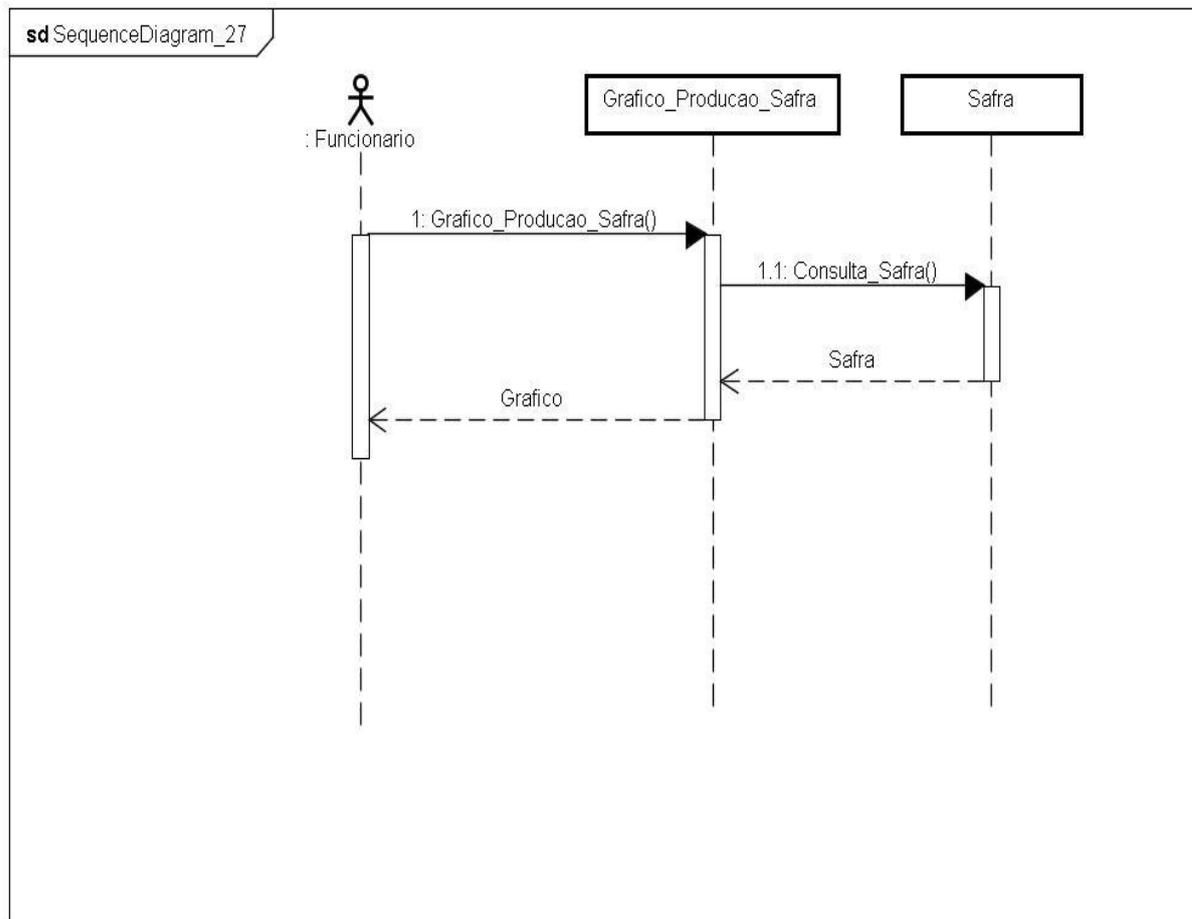
**Figura 47 – Diagrama de Sequência: DiagramSequence\_24 – Relatório de Venda.**



**Figura 48 – Diagrama de Sequência: DiagramSequence\_25 – Relatório de Contas a Pagar.**



**Figura 49 – Diagrama de Sequência: DiagramSequence\_26 – Relatório de Atividade.**



**Figura 50 – Diagrama de Sequência: DiagramSequence\_27 – Gráfico de Produção por Safra.**

### 3.6. DIAGRAMA DE CLASSES

O diagrama de classes(Figura 56) descreve como o sistema é estruturado.

Segundo C. Lee e Tepfenhart (2001).

Descreve “o que são as coisas” e seus relacionamentos estáticos com outras coisas. Descrever todos os objetos em um sistemas seria um tanto tedioso, pois muitos deles teriam características similares e, em qualquer caso, alguns deles seriam criados e destruídos à medida que o programa prosseguisse. Digrama de classe são portanto utilizados com mais freqüência do que diagramas de objetos para mostrar essa perspectiva. (C. Lee e Tepfenhart, 2001, 510).

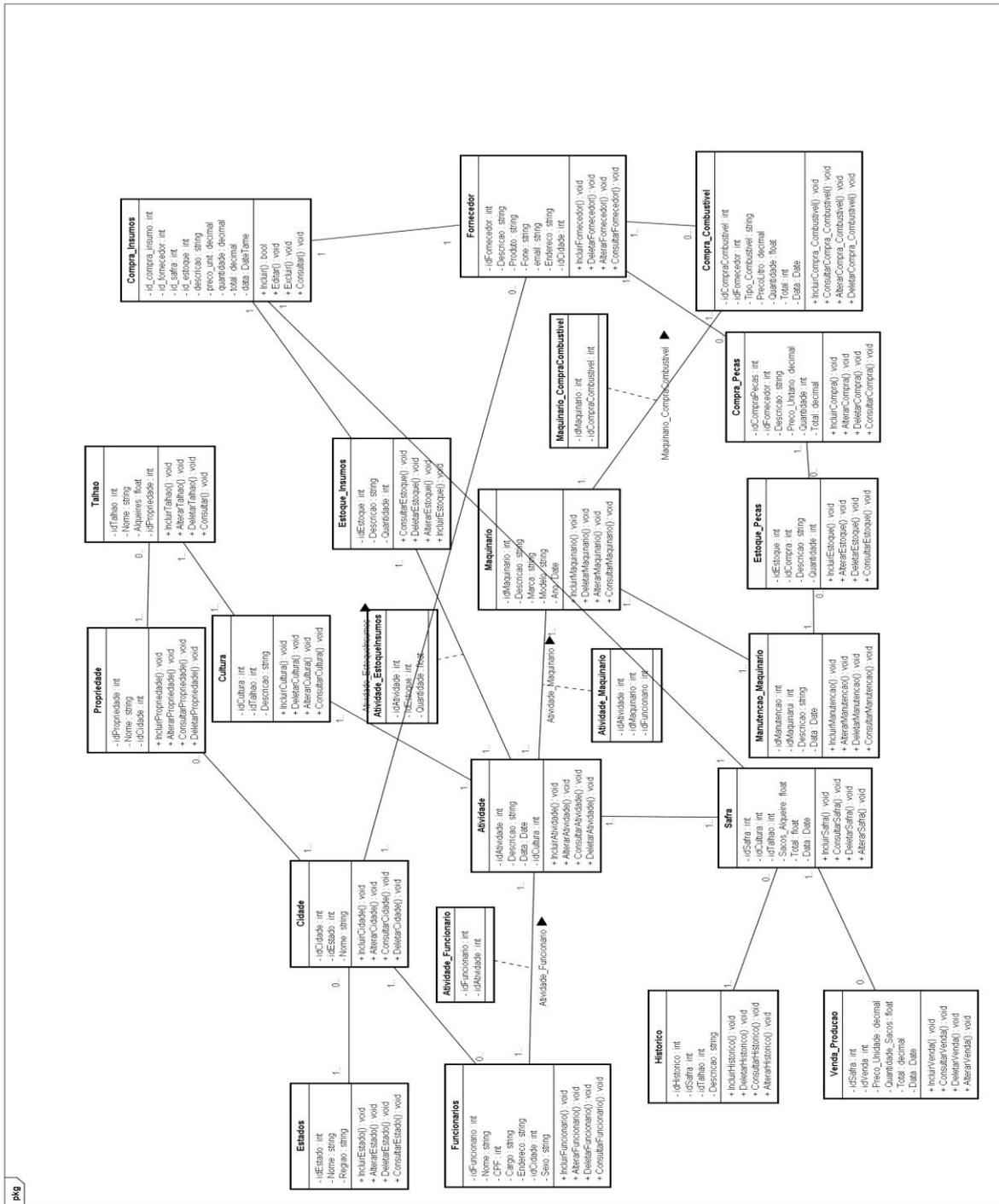


Figura 51 – Diagrama de Classes.

### 3.7. DIAGRAMA ENTIDADE RELACIONAMENTO(Figura 57)

Segundo Martin e Odell (1996)

Para os diagramas ER, as associações são expressas ou dentro ou fora de tipos de entidade. Aquelas que estão dentro são denominadas tipos de atributos. Aquelas que estão fora são representadas em termos de tipos de relação com outros tipos de entidade. Tecnicamente, então, os tipos de objetos e associações de um esquema de 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).

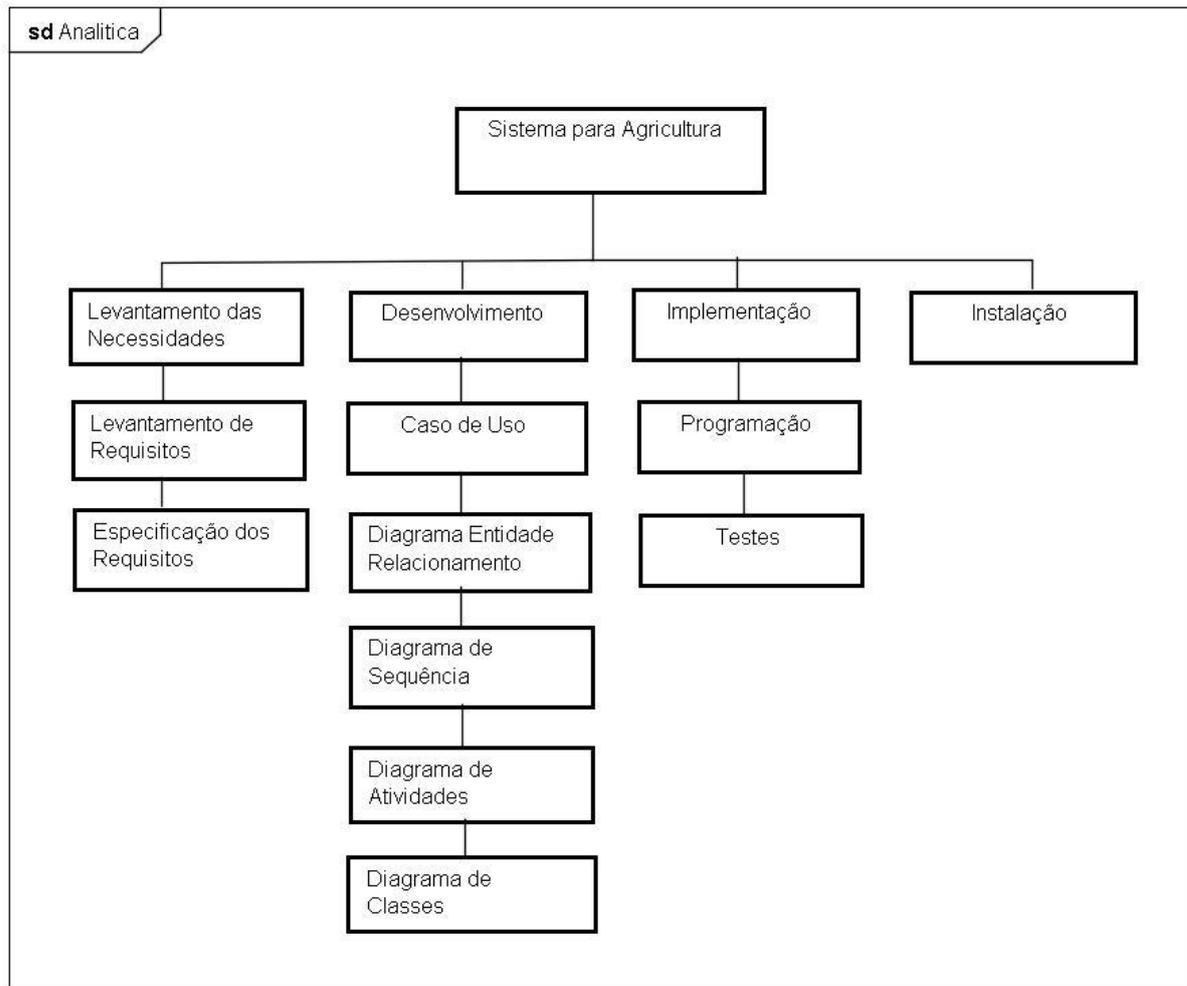


### 3.8. ESTRUTURA ANALÍTICA DO PROJETO

Estrutura Analítica(Figura 58) de Projetos (EAP), do Inglês, *Work breakdown structure* (WBS) é uma ferramenta de decomposição do trabalho do projeto em partes manejáveis. É estruturada em árvore exaustiva, hierárquica (de mais geral para mais específica) orientada às entregas (*deliverables*) que precisam ser feitas para completar um projeto.

Segundo o PMBOK (2011)

Uma Estrutura Analítica de Projetos (EAP), do Inglês, *Work Breakdown Structure* (WBS) é uma ferramenta de decomposição do trabalho do projeto em partes manejáveis. É a estrutura hierárquica (de mais geral para mais específica) orientada às entregas (*deliverables*) que precisam ser feitas para completar um projeto. Dentre os itens listados identifique o que não corresponde a um mandamento da Estrutura Analítica do Projeto.



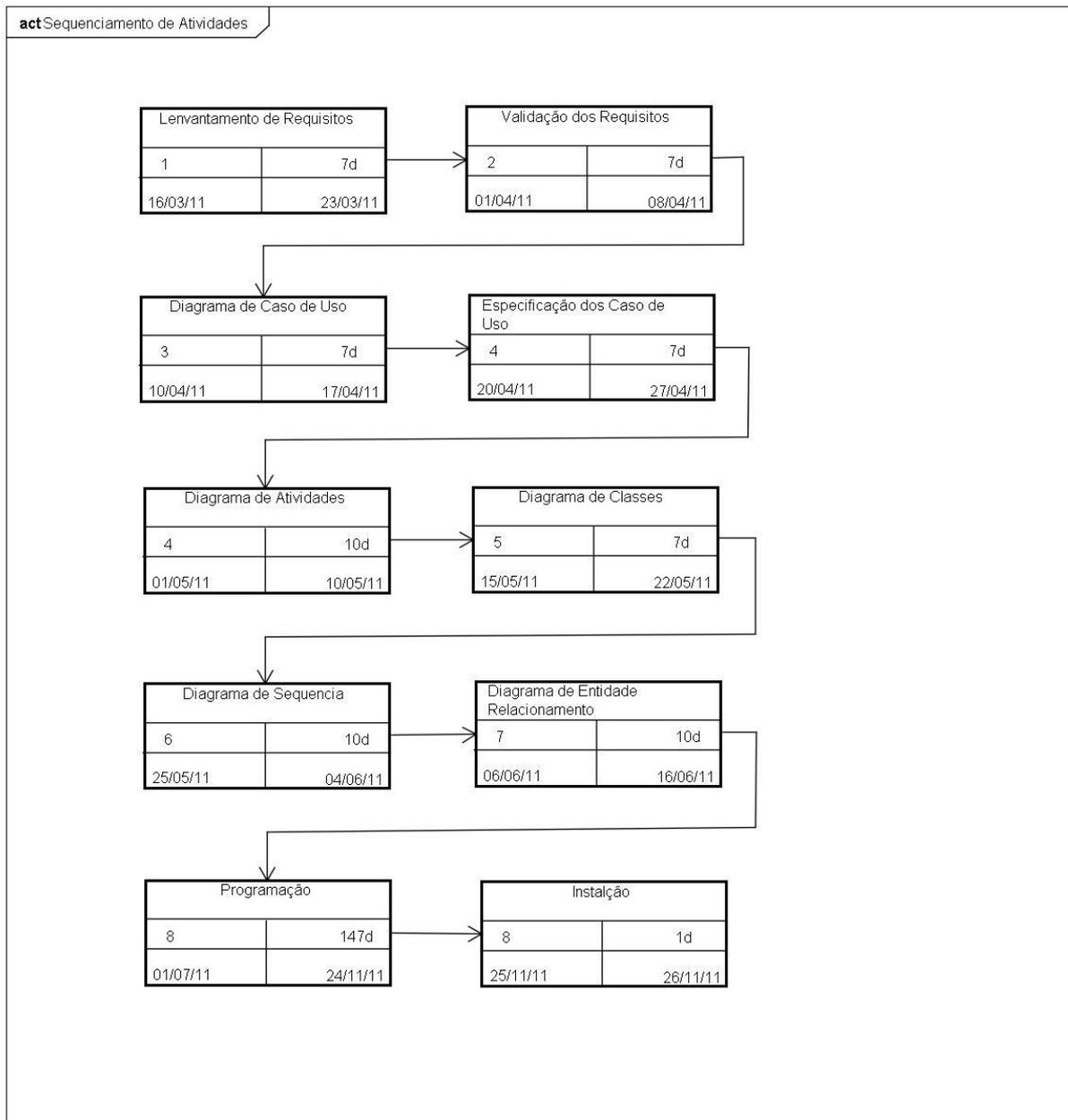
**Figura 53 - Estrutura Analítica do Projeto**

### 3.9. SEQUENCIAMENTO DE ATIVIDADES

O sequenciamento de atividades(Figura 59) envolve a identificação e documentação dos relacionamentos lógicos entre as atividades do cronograma.

De acordo com o PMBOK (2011)

O sequenciamento da atividade envolve identificar e documentar os relacionamentos lógicos entre as atividades. As atividades devem ser seqüenciadas corretamente para suportar o desenvolvimento de um cronograma realístico e alcançável. O sequenciamento pode ser feito com o auxílio de um computador (por exemplo, utilizando softwares de gerência de projeto) ou com técnicas manuais. As técnicas manuais são, geralmente, mais efetivas em projetos menores e em fases iniciais de projetos maiores quando poucos detalhes estão disponíveis. As técnicas manuais e automatizadas podem, também, ser utilizadas em conjunto.



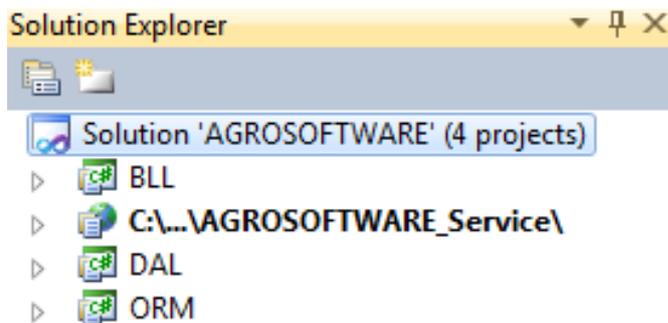
**Figura 54 – Sequenciamento de Atividades**

## 4. IMPLEMENTAÇÃO

Para a implementação do sistema foi utilizada a plataforma .net com a linguagem c#.

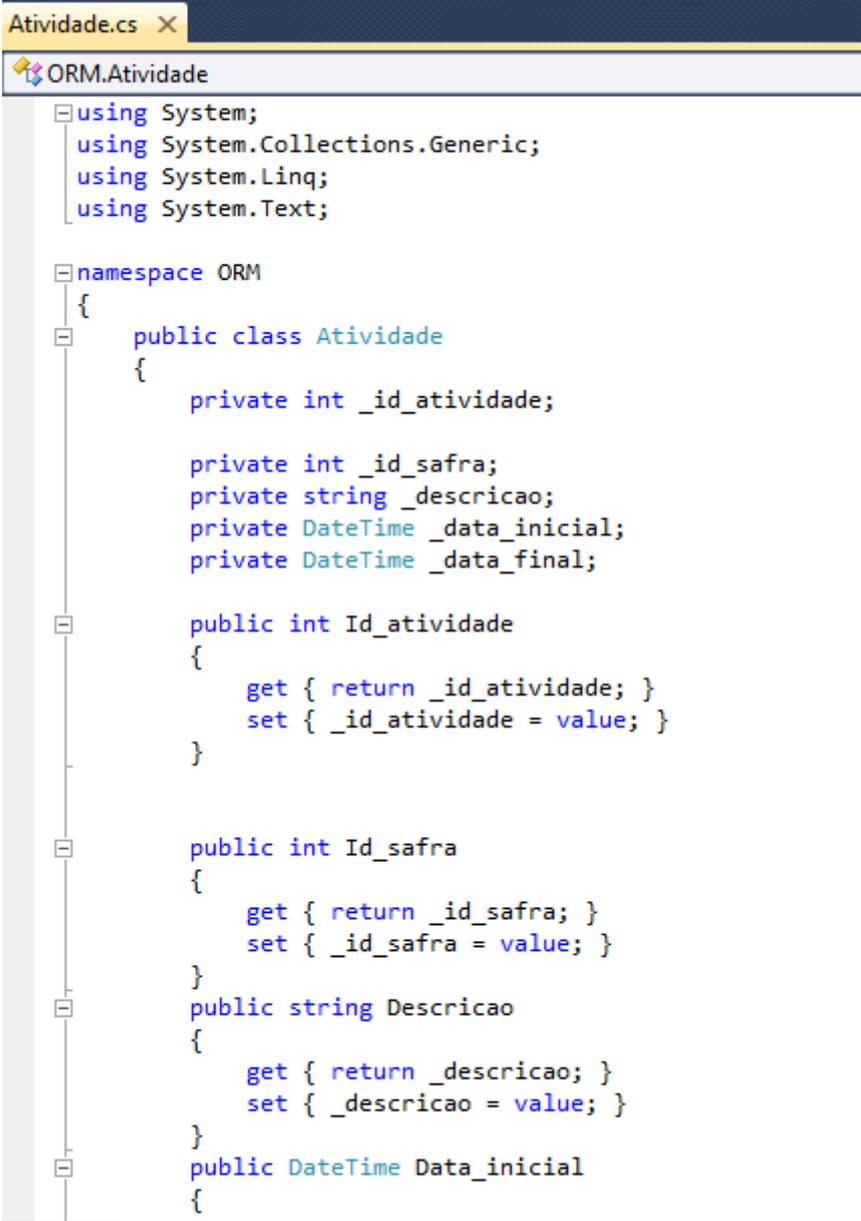
### 4.1. ORGANIZAÇÃO DO PROJETO

Para a organização do projeto, o sistema foi desenvolvido em três camadas(Figura 55), com utilização de um Webservice, camada AGROSOFTWARE ORM, camada AGROSOFTWARE.DAL, AGROSOFTWARE.BLL.



**Figura 55 – Camadas.**

Camada AGROSOFTWARE ORM(Figura 56): é a camada onde são criadas as entidades que modelam o sistema.



```
Atividade.cs X
ORM.Atividade
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ORM
{
    public class Atividade
    {
        private int _id_atividade;

        private int _id_safra;
        private string _descricao;
        private DateTime _data_inicial;
        private DateTime _data_final;

        public int Id_atividade
        {
            get { return _id_atividade; }
            set { _id_atividade = value; }
        }

        public int Id_safra
        {
            get { return _id_safra; }
            set { _id_safra = value; }
        }

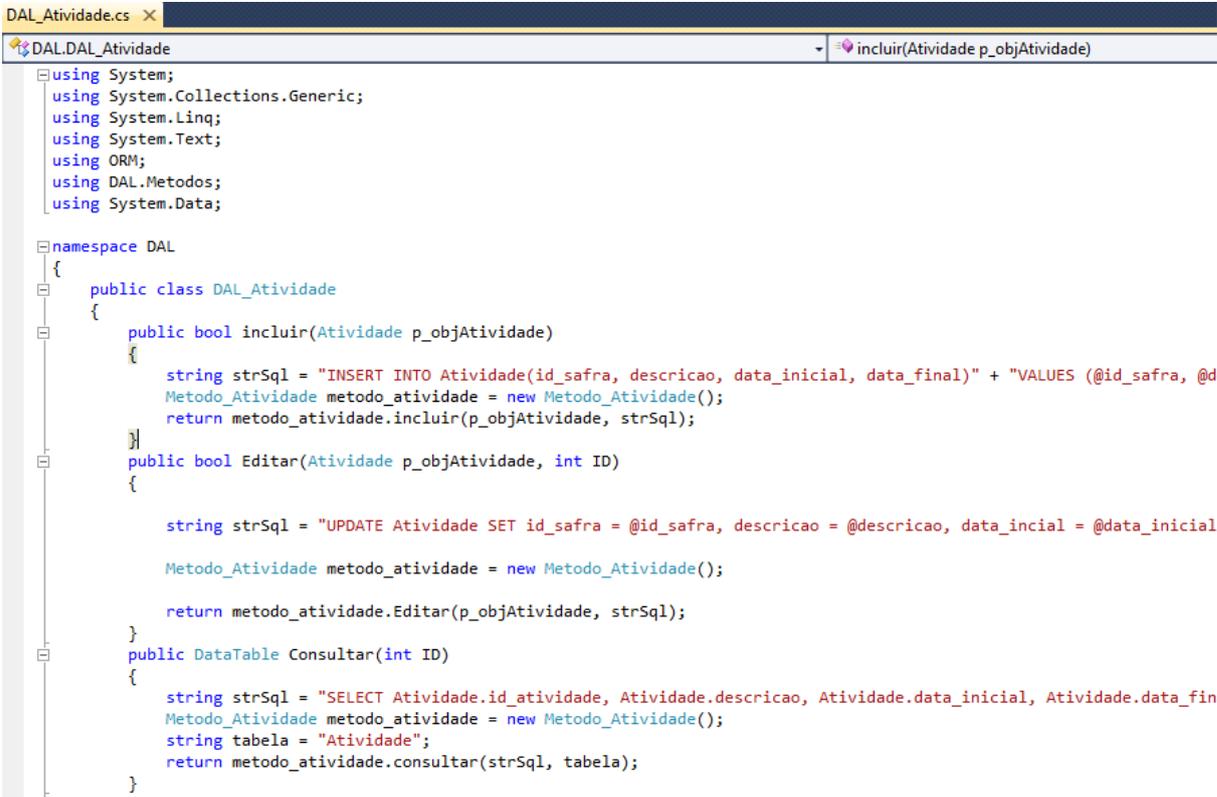
        public string Descricao
        {
            get { return _descricao; }
            set { _descricao = value; }
        }

        public DateTime Data_inicial
        {
```

Figura 56 – ORM.

Camada AGROSOFTWARE.DAL(Figura 57): é a camada onde são criados os métodos que fazem relacionamento com o banco de dados, na camada DAL foi criada a ConnectionManeger(Figura 58), onde foram criados alguns métodos que podem ser aproveitados por todos os métodos da camada DAL como por exemplo, os métodos executarComando(Figura 58), que executa comandos como *insert* e *update* retornando falso se o comando falhar ou verdadeiro se for executado corretamente e o método Consultar(Figura 59) que executa todos os comandos *select*, retornando um *DataTable*, o mesmo pode ser usado por todos os métodos de consulta, independentemente da tabela a ser consultada.

O folder Métodos(Figura 60), contem os métodos incluir , excluir, editar, consultar, que executam os comandos passados pelo parâmetro string strSql, assim garantindo maior reaproveitamento de códigos.



```

DAL_Atividade.cs x
DAL.DAL_Atividade
incluir(Atividade p_objAtividade)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using ORM;
using DAL.Metodos;
using System.Data;

namespace DAL
{
    public class DAL_Atividade
    {
        public bool incluir(Atividade p_objAtividade)
        {
            string strSql = "INSERT INTO Atividade(id_safra, descricao, data_inicial, data_final)" + "VALUES (@id_safra, @d
            Metodo_Atividade metodo_atividade = new Metodo_Atividade();
            return metodo_atividade.incluir(p_objAtividade, strSql);
        }

        public bool Editar(Atividade p_objAtividade, int ID)
        {
            string strSql = "UPDATE Atividade SET id_safra = @id_safra, descricao = @descricao, data_inicial = @data_inicial
            Metodo_Atividade metodo_atividade = new Metodo_Atividade();

            return metodo_atividade.Editar(p_objAtividade, strSql);
        }

        public DataTable Consultar(int ID)
        {
            string strSql = "SELECT Atividade.id_atividade, Atividade.descricao, Atividade.data_inicial, Atividade.data_fin
            Metodo_Atividade metodo_atividade = new Metodo_Atividade();
            string tabela = "Atividade";
            return metodo_atividade.consultar(strSql, tabela);
        }
    }
}

```

Figura 57 – DAL.

```

public bool executarComando(string p_strSql, List<SqlParameter> p_obParams)
{
    bool blnResult = false;
    if (!this.conectar())
    {
        return false;
    }

    SqlCommand objCmd = new SqlCommand(p_strSql, objConn);

    foreach (SqlParameter param in p_obParams)
    {
        objCmd.Parameters.Add(param);
    }

    try
    {
        blnResult = (objCmd.ExecuteNonQuery() > 0 ? true : false);
    }
    catch
    {
        blnResult = false;
    }

    this.desconectar();

    return blnResult;
}

```

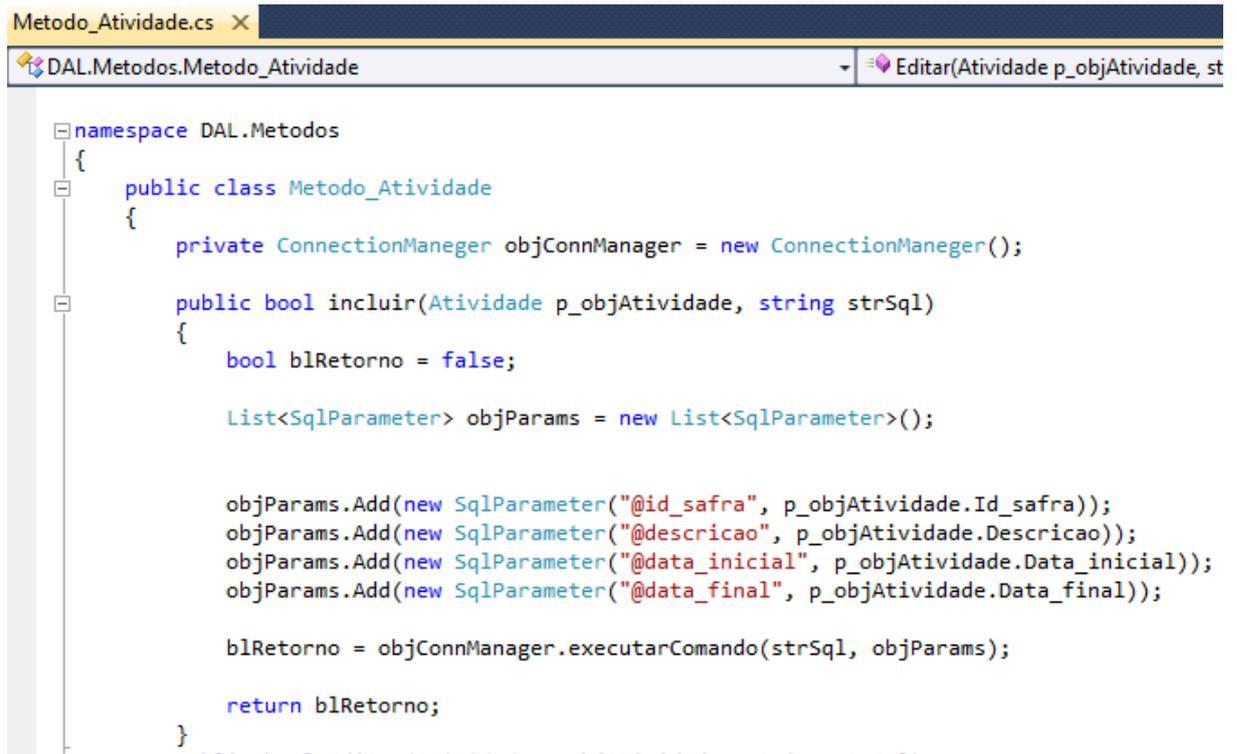
**Figura 58 – executarComando.**

```

public DataTable Consultar(string p_strSql, string p_strNmTabelaRetorno)
{
    conectar();
    SqlCommand objCmd = new SqlCommand(p_strSql, objConn);
    SqlDataAdapter objAdp = new SqlDataAdapter(objCmd);
    DataSet ds = new DataSet();
    objAdp.Fill(ds, p_strNmTabelaRetorno);
    desconectar();
    return ds.Tables[p_strNmTabelaRetorno];
}

```

**Figura 59 - Consultar**



```
Metodo_Atividade.cs x
DAL.Metodos.Metodo_Atividade Editar(Atividade p_objAtividade, st

namespace DAL.Metodos
{
    public class Metodo_Atividade
    {
        private ConnectionManeger objConnManager = new ConnectionManeger();

        public bool incluir(Atividade p_objAtividade, string strSql)
        {
            bool blRetorno = false;

            List<SqlParameter> objParams = new List<SqlParameter>();

            objParams.Add(new SqlParameter("@id_safra", p_objAtividade.Id_safra));
            objParams.Add(new SqlParameter("@descricao", p_objAtividade.Descricao));
            objParams.Add(new SqlParameter("@data_inicial", p_objAtividade.Data_inicial));
            objParams.Add(new SqlParameter("@data_final", p_objAtividade.Data_final));

            blRetorno = objConnManager.executarComando(strSql, objParams);

            return blRetorno;
        }
    }
}
```

Figura 60 – Métodos.

Camada AGROSOFTWARE.BLL(Figura 61): é a camada de negócios e também a camada que deve ser referenciada pelo WebService(AGROSOFTWARE\_Service(Figura 62)), o WebService disponibiliza serviços como os métodos criados na camada DAL, os serviços podem ser acessados por qualquer aplicativo criado em qualquer plataforma que suporte XML, os serviços são acessados pelo projeto VIEWAGROSOFTWARE(Figura 63), onde estão as interfaces, com essa estrutura separando as camadas que fazem relacionamento com o banco de dados, da interface facilita a instalação, por exemplo: o AGROSOFTWARE é instalado em um servidor e seus serviços AGROSOFTWARE\_Service estão online, agora é só instalar o VIEWAGROSOFTWARE em algum computador em qualquer local do mundo onde esteja disponível conexão com a internet.

```
namespace BLL
{
    public class BLL_Atividade
    {
        public bool Incluir(Atividade p_objAtividade)
        {
            DAL_Atividade dal = new DAL_Atividade();
            return dal.incluir(p_objAtividade);
        }
        public bool Editar(Atividade p_objAtividade, int ID)
        {
            DAL_Atividade dal = new DAL_Atividade();
            return dal.Editar(p_objAtividade, ID);
        }
        public DataTable Consultar(int id)
        {
            DAL_Atividade dal = new DAL_Atividade();
            return dal.Consultar(id);
        }
        public DataTable InnerJoinFunc_Maq_imp(int id_atividade)
        {
            DAL_Atividade dal = new DAL_Atividade();
            return dal.InnerJoinFunc_Maq_imp(id_atividade);
        }
        public DataTable ConsultarTodos()
        {
            DAL_Atividade dal = new DAL_Atividade();
            return dal.ConsultarTodos();
        }
    }
}
```

**Figura 61 – BLL.**

```

Service_Atividade Service_Atividade()
[WebMethod]
public DataTable InnerJoinFunc_Maq_imp(int id_atividade)
{
    BLL_Atividade bll = new BLL_Atividade();
    return bll.InnerJoinFunc_Maq_imp(id_atividade);
}
[WebMethod]
public bool Incluir(int id_safra, string descricao, DateTime data_inicial, DateTime data_final)
{
    try
    {
        Atividade atividade = new Atividade();

        atividade.Id_safra = id_safra;
        atividade.Descricao = descricao;
        atividade.Data_inicial = data_inicial;
        atividade.Data_final = data_final;
        BLL_Atividade bll = new BLL_Atividade();
        return bll.Incluir(atividade);
    }
    catch (Exception ex)
    {
        throw new Exception(ex.ToString());
    }
}
[WebMethod]
public DataTable Consultar(int id_atividade)
{
    BLL_Atividade bll = new BLL_Atividade();
    return bll.Consultar(id_atividade);
}

```

Figura 62 – AGROSOFTWARE\_Service.

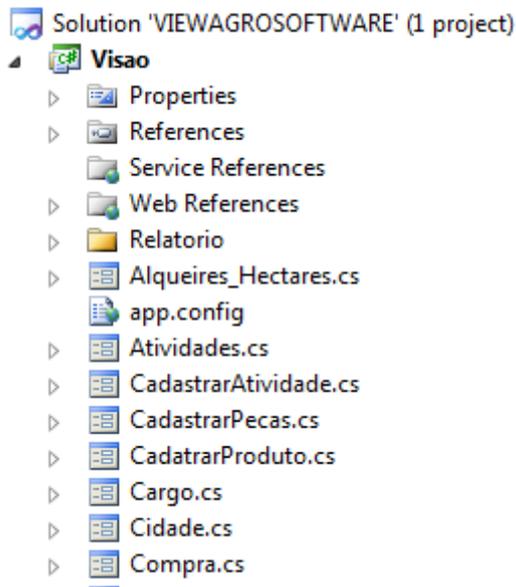


Figura 63 – VIEWAGROSOFTWARE.

## 4.1. INTERFACES DO SISTEMA

Ao acessar o sistema abre o formulário Menu(Figura 69) com cinco opções, a previsão do tempo está disponível no formulário menu.

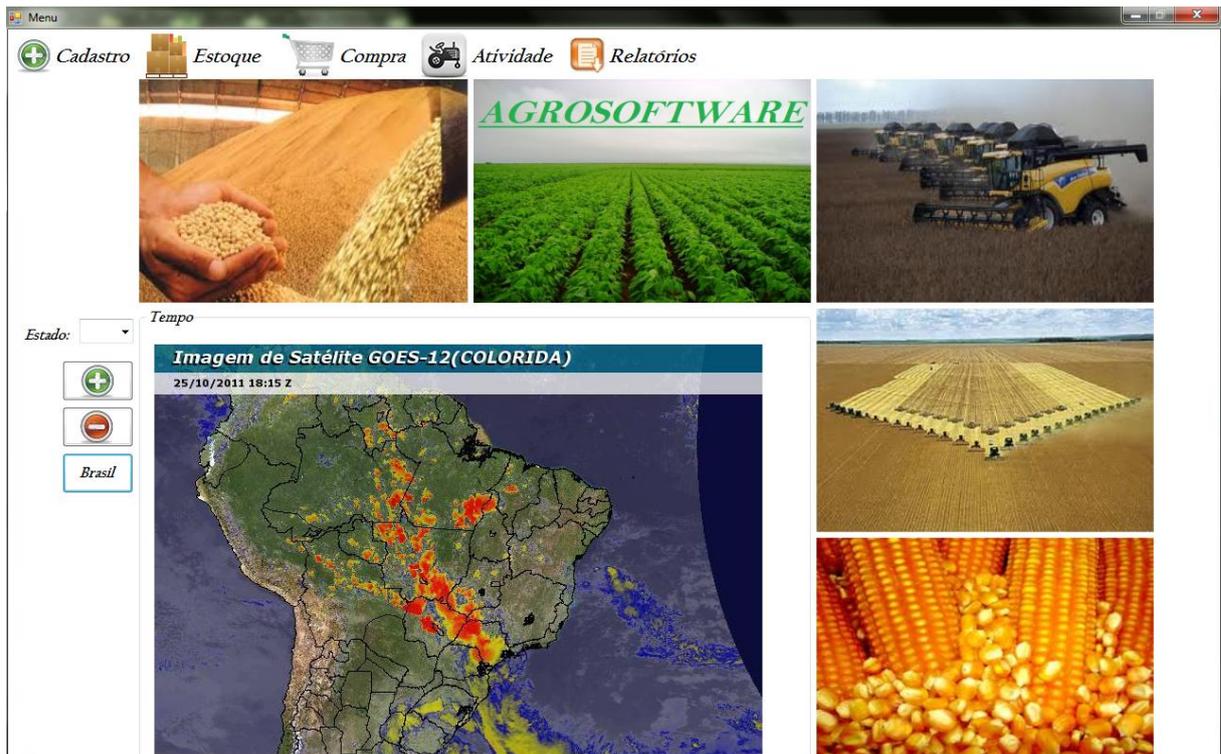


Figura 64 - Menu.

Ao clicar em cadastro, surgira as opções dos cadastros por exemplo: cadastro de funcionário(Figura 75).

| ID Funcionário | Nome     | CPF      | Fone     | Sexo | Salário | Cargo         | Endereço |
|----------------|----------|----------|----------|------|---------|---------------|----------|
| 14             | Fernando | 63636363 | 62626262 | M    | 1000,00 | Administrador | sítio    |
| 13             | Antonio  | 83838383 | 22333444 | M    | 1000,00 | Tratorista    | sítio    |
| 12             | Mario    | 837373   | 737383   | M    | 1000,00 | Tratorista    | sítio    |

**Incluir**

Nome:

CPF:

Fone:

Sexo: M

Salário:

Cargo: Administrador

Cidade: AC

Endereço:

**Editar**

Código:

Nome:

CPF:

Fone:

Sexo: M

Salário:

Cargo: Administrador

Cidade: AC

Endereço:

Figura 65 – Cadastro de Funcionário.

Ao clicar em Estoque surgirão as opções de estoque, exemplo: insumos(Figura 66).

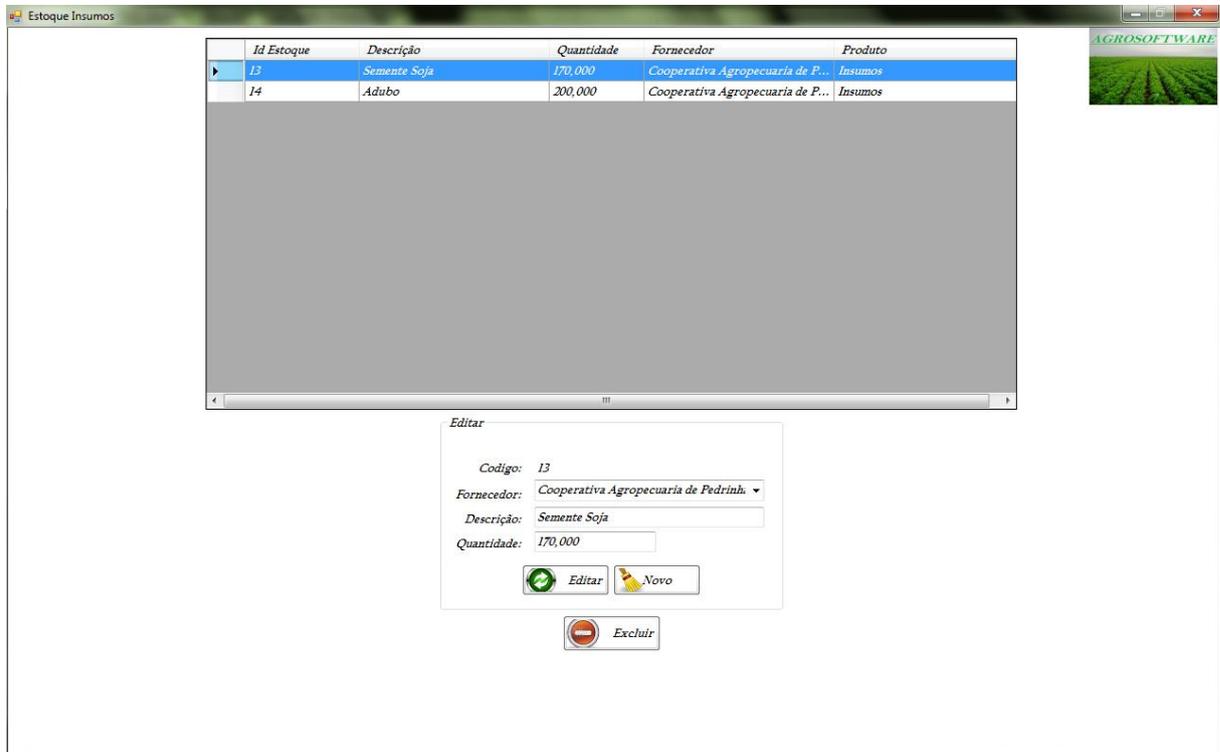


Figura 66 – Estoque Insumos.

Ao clicar na opção compra surgirá o formulário de compra(Figura 77), com três opções de compra, peças, combustível e insumos.

Compra

Escolha uma opção

Compra Insumos  Compra Combustível  Compra Peças

Comprar

Fornecedor: Coopdrinhas Seção de Peças

Produto: ..

Endereço: ..

Fone: ..

Safrá: Soja

Data Inicial: ..

Data Final: ..

Tipo Combustível: ..

Preço Unitário: ..

Quantidade: ..

Total: ..

Data e Hora: ..:..:..

Incluir

Novo

Figura 67 – Compra.

Ao clicar na opção atividade abre o formulário atividade(Figura 68), onde é feito o relacionamento, funcionário, maquinários, propriedades, talhões e insumos.

Atividade

**Relação para a Atividade**

*Relação Propriedade e Insumos*

Seleção a Atividade: Aplicação de Veneno   
 Data Atividade: 01/10/2011 00:00:00 - 01/10/2011 00:00:00  
 Safra: Trigo  
 Data Safra: 11/11/2011 - 10/10/2012   
 Seleção a Propriedade: São Geraldo   
 Hectares: 25,000  
 Cidade: Pedrinhas Paulista   
 Talhões: T1   
 Hectares: 10,000  
 Insumos: Semente Soja   
 Fornecedor: Cooperativa Agropecuaria de Pedrinhas Pa  
 Quantidade em Estoque: 200,000  
 Preço Unitário: 40,00  
 Quantidade: 20  
 Total: 800,00  
 Incluir

| Atividade           | Talhão | Hectares |
|---------------------|--------|----------|
| Aplicação de Veneno | T1     | 10,000   |
| Aplicação de Veneno | T1     | 10,000   |
| Aplicação de Veneno | T2     | 15,000   |

 Incluir

**Gerar Relatório da Relação**

*Cadastrar apenas Funcionários Sem Relacionamento com Maquinários*

Funcionario: Antonio   
 Cargo: Antonio  
 Sexo: M  
 Incluir

| Atividade           | Funcionário |
|---------------------|-------------|
| Aplicação de Veneno | Mario       |
| Aplicação de Veneno | Antonio     |

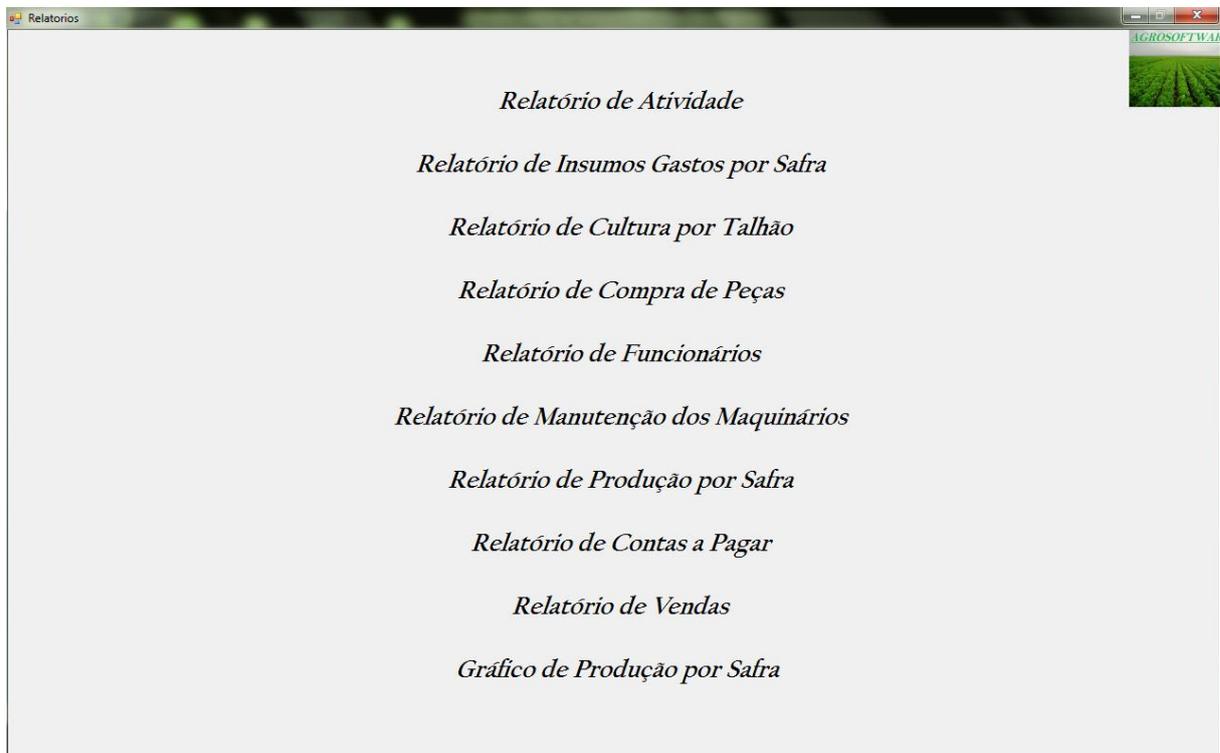
*Relação Funcionário Maquinário*

Funcionario: Mario   
 Cargo: Tratorista  
 Sexo: M  
 Maquinário: Trator 290   
 Marca: MarchelFerguson  
 Potencia: 100  
 Sem Implemento  
 Implemento: plantadeira   Incluir

| Atividade           | Funcionário | Maquinário       |
|---------------------|-------------|------------------|
| Aplicação de Veneno | Mario       | Colheitadeira TC |
| Aplicação de Veneno | Mario       | Colheitadeira TC |
| Aplicação de Veneno | Mario       | Trator 290       |
| Aplicação de Veneno | Mario       | Trator 290       |

Figura 68 – Atividade.

Ao clicar na opção Relatórios abre o formulário Relatórios(Figura 69) onde o usuário deverá escolher qual relatório deseja gerar.



**Figura 69 – Relatórios.**

## 5. ORÇAMENTO

- 01 Analista – Programador.
- 01 Notebook Centrium, Dual - Core, 4 Gigas de memória RAM, HD de 320 Giga.
- 01 Multifuncional HP.
- Visual Studio 2010.
- SQL Server 2005 Express Edition: Gratuito.

## 5.1. ESTIMATIVA DE CUSTOS

Custo Analista – Programador

Custo Diário: R\$ 45,00 (Quarenta e cinco Reais).

Total de Dias: 167.

Custo Total:  $(167 * 45,00) = R\$ 6.680,00$  (Seis mil e seiscentos e oitenta reais).

## 5.2 CUSTOS DE MATERIAIS

Notebook: R\$ 1.700,00 (Mil e setecentos reais).

Depreciação 3 anos:  $R\$ 1.700,00 / 36$  (meses) = R\$ 47,22 (Quarenta e sete centavos e vinte e dois centavos) por mês.

Custo de um dia:  $R\$ 47,22 / 30$  (dias) = R\$ 1,57 (um real e cinquenta e sete centavos) por dia.

Custo de 167 dias:  $R\$ 1,57 * 167 = R\$ 262,19$  (duzentos e sessenta e dois reais e dezenove centavos).

Multifuncional: R\$ 265,00 (duzentos e setenta reais).

Depreciação 3 anos:  $R\$ 265,00 / 36$  (meses) = R\$ 7,36 (sete reais e trinta e seis centavos) por mês.

Custo de um dia:  $R\$ 7,36 / 30$  (dias) = R\$ 0,24 (vinte e quatro centavos) por dia.

Custo de 167 dias:  $R\$ 0,24 * 167 = R\$ 40,97$ (quarenta reais e noventa e sete centavos).

Sistema Operacional Microsoft Windows 7 Ultimate: R\$ 669,00 (seiscentos e sessenta e nove reais).

Depreciação 2 anos:  $R\$ 669,00 / 24$  (meses) = R\$ 27,87 (vinte e sete reais e oitenta e sete centavos) por mês.

Custo de um dia:  $R\$ 27,87 / 30$  (dias) = R\$ 0,92 (noventa e dois centavos) por dia.

Custo de 167 dias:  $R\$ 0,92 * 167 = R\$ 153,64$  (cento e cinquenta e três reais e sessenta e quatro centavos).

Visual Studio 2010: R\$ 890,00 (oitocentos e noventa reais).

Depreciação 3 anos:  $R\$ 890,00 / 36$  (meses) = R\$ 24,72 (vinte e quatro reais e setenta e dois centavos) por mês.

Custo de um dia:  $R\$ 24,72 / 30$  (dias) = R\$ 0,82 (oitenta e dois centavos) por dia.

Custo de 167 dias:  $R\$ 0,82 * 167 = R\$ 137,62$  (cento e trinta e sete reais e sessenta e dois centavos).

### 5.3. ORÇAMENTO DO PROJETO

| <b>Recursos Utilizados</b> | <b>Valor</b>         |
|----------------------------|----------------------|
| Analista - Programador     | R\$ 6.680,00         |
| Notbook                    | R\$ 1.700,00         |
| Multifuncional             | R\$ 265,00           |
| Visual Studio 2010         | R\$ 890,00           |
| Windows 7 Ultimate         | R\$ 669,00           |
| <b>Total</b>               | <b>R\$ 10.204,00</b> |

Figura 70 - Orçamento.

## 6. CRONOGRAMA

O cronograma é um instrumento de planejamento e controle semelhante a um diagrama, em que são definidas e detalhadas minuciosamente as atividades a serem executadas durante um período estimado.

| Atividades/Periodo           | FEV | MAR | ABR | MAI | JUN | JUL | AGO | SET | OUT | NOV | DEZ |
|------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Pré-Projeto                  | ■   | ■   | ■   |     |     |     |     |     |     |     |     |
| Levantamento de Requisitos   |     | ■   | ■   |     |     |     |     |     |     |     |     |
| Validação dos Requisitos     |     | ■   | ■   |     |     |     |     |     |     |     |     |
| Estrutura Analítica          |     | ■   | ■   |     |     |     |     |     |     |     |     |
| Caso de Uso                  |     | ■   | ■   |     |     |     |     |     |     |     |     |
| Diagrama de Sequencia        |     | ■   | ■   |     |     |     |     |     |     |     |     |
| Diagrama de Classes          |     | ■   | ■   |     |     |     |     |     |     |     |     |
| Diagrama de ER               |     | ■   | ■   |     |     |     |     |     |     |     |     |
| Sequenciamento de Atividades |     | ■   | ■   |     |     |     |     |     |     |     |     |
| Orçamento                    |     | ■   | ■   |     |     |     |     |     |     |     |     |
| Escrita da Qualificação      |     |     | ■   | ■   | ■   | ■   |     |     |     |     |     |
| Implementação                |     |     |     |     |     | ■   | ■   | ■   | ■   |     |     |
| Documento Final do TCC       |     |     |     |     |     | ■   | ■   | ■   | ■   | ■   |     |
| Testes                       |     |     |     |     |     |     |     |     | ■   | ■   |     |
| Apresentação do TCC          |     |     |     |     |     |     |     |     |     | ■   |     |

**Figura 71 – Cronograma**

## 7. CONCLUSÃO

As funcionalidades desenvolvidas dão um diferencial comparado aos softwares disponíveis no mercado. O gerenciamento das atividades permite ao usuário uma visão ampla de tudo que está sendo desenvolvido em sua propriedade.

A etapa de análise facilitou o entendimento do sistema. Por meio da lista de eventos e os casos de uso foi possível identificar com mais facilidade as funcionalidades do sistema. O diagrama de sequência facilitou a compreensão do funcionamento do software, enquanto que o diagrama de classes e o diagrama entidade relacionamento orientou no desenvolvimento do projeto.

No desenvolvimento deste projeto, existiram algumas dificuldades pelo fato de estar trabalhando com uma integração de dois projetos por meio de um Webservice.

## 8 REFERÊNCIAS

SHARP, John. MICROSOFT VISUAL C# 2005 Passo a passo. Tradução de Altair Dias Caldas de Moraes. Porto Alegre, Editora Bookman, 2007.

Soukup, Ron. Desvendando o Microsoft SQL Server 6.5. Tradução de Vitor Hugo da Paixão Alves, João E. N. Tortello, Daniel Vieira, Rio de Janeiro, Editora Campus, 1998.

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.

Martin, James e Odell, James J. Análise e Projeto Orientados a Objeto. Tradução de José Carlos Barbosa dos Santos. São Paulo, Editora Makron books Ltda, 1996.

<<http://www.pmkb.com.br/artigos-mainmenu-25/3229-questionario-sobre-gestao-de-projetos-02.html>>. Acesso em: 19. Jun. 2011.

<[http://www.cin.ufpe.br/~if717/Pmbok2000/pmbok\\_v2p/wsp\\_6.2.html](http://www.cin.ufpe.br/~if717/Pmbok2000/pmbok_v2p/wsp_6.2.html)>. Acesso em: 3. Abr. 2011.

<<http://www.adobe.com/br/products/flex/overview/>>. Acesso em: 20. Jun. 2011.