

**Carlos Alberto dos Santos**

Sistema para a venda de materiais de informática

**Assis  
2009**

# Sistema para a venda de materiais de informática

**Carlos Alberto dos Santos**

Trabalho de Conclusão de Curso apresentado ao  
Instituto Municipal de Ensino Superior de Assis,  
como requisito do Curso de Graduação, analisado  
pela seguinte comissão examinadora:

Orientador: \_\_\_\_\_

Analisador (1): \_\_\_\_\_

Analisador (2): \_\_\_\_\_

Carlos Alberto dos Santos

## SISTEMA PARA VENDA DE MATERIAIS DE INFORMÁTICA

Trabalho de Conclusão de Curso apresentado ao  
Instituto Municipal de Ensino Superior de Assis,  
como requisito do Curso de Graduação, analisado  
pela seguinte comissão examinadora:

Orientador: \_\_\_\_\_

Área de concentração: \_\_\_\_\_

\_\_\_\_\_

Assis  
2009

**Carlos Alberto dos Santos**

Sistema para a venda de materiais de informática

**Assis  
2009**

## Sistema para a venda de materiais de informática

**Carlos Alberto dos Santos**

Trabalho de Conclusão de Curso apresentado ao  
Instituto Municipal de Ensino Superior de Assis,  
como requisito do Curso de Graduação, analisado  
pela seguinte comissão examinadora:

Orientador: \_\_\_\_\_

Analisador (1): \_\_\_\_\_

Analisador (2): \_\_\_\_\_

**Carlos Alberto dos Santos**

Sistema para a venda de materiais de informática

Trabalho de Conclusão de Curso apresentado ao  
Instituto Municipal de Ensino Superior de Assis,  
como requisito do Curso de Graduação, analisado  
pela seguinte comissão examinadora:

Orientador: \_\_\_\_\_

Área de Concentração: \_\_\_\_\_

\_\_\_\_\_

Assis  
2008

## AGRADECIMENTOS

Ao nosso bom DEUS, pai maravilhoso que me iluminou o caminho me abençoando com o discernimento para distinguir qual o caminho a ser seguido, pela inteligência que me destes.

Aos familiares, que me deram estímulo para não desistir no início do curso, que nesse período encontrei muitas dificuldades em aprender as matérias, mas consegui, graças a DEUS.

Ao professor, DOMINGOS DE CARVALHO VILLELA JUNIOR, pela orientação e pelo constante estímulo transmitido durante o trabalho.

Aos amigos, da faculdade que me ajudaram muito, do serviço sempre me dando uma voz de incentivo, e a todos que colaboraram direta ou indiretamente, na execução deste trabalho.

## RESUMO

O objetivo principal do trabalho é oferecer um sistema de fácil entendimento para o usuário utilizando a Orientação a Objeto junto às técnicas UML (Linguagem de Modelagem Unificada), para o auxílio na modelagem do sistema.

O Sistema tem a funcionalidade para Gerar Cadastros, Movimentar entradas e saídas de produtos, Movimentar Pagamentos e Emitir Relatórios diversos, para que a empresa possa aumentar sua posição no mercado concorrente na cidade de Quatá, já que a empresa trabalha de forma manual.

**Palavras-chave:** Venda – informática – e-commerce

## ABSTRACT

The objective principal of the work is to offer a system of easy understanding for the user using the Orientation to I Object the techniques close to UML (Language of Unified Modelling), for I aid him in the modelling of the system.

The System has the functionality to Generate Cadasters, to Move entrances and exits of products, to Move Payments and to Emit several Reports, so that the company can increase your position in the competitive market in the city of Quatá, since the company works in a manual way.

**Works-key:** sale – computer science – e-commerce

## LISTA DE ILUSTRAÇÕES

Figura 01 – exemplo de caso de uso .....	23
Figura 02 – Exemplo de Diagrama de classe .....	24
Figura 03 – Exemplo de Diagrama de Objetos .....	24
Figura 04 – Exemplo de Diagrama de Estado.....	25
Figura 05 – Exemplo de Diagrama de Seqüência .....	26
Figura 06 – Exemplo de Diagrama de Colaboração .....	27
Figura 07 – Exemplo de Diagrama de Atividades.....	28
Figura 08 – Exemplo de Diagrama de Componentes .....	29
Figura 09 – Exemplo de Diagrama de Implantação.....	30
Figura 10 – Área de Trabalho do Visual Studio .....	31
Figura 11 – Área de trabalho do SQL Server 2005 express.....	32
Figura 12 – Netbeans .....	33
Figura 13 – EA (Enterprise Architect) .....	34
Figura 14 – Cronograma de atividades .....	35
Figura 15 – WBF .....	36
Figura 16 – Seqüência de Atividades.....	37
Figura 17 – diagrama Use case – cadastro de pais.....	42
Figura 18 – diagrama Use case – cadastro de estado .....	43
Figura 19 – diagrama Use case – cadastro de cidade .....	44
Figura 20 – diagrama Use case – cadastro de bairro .....	45
Figura 21 – diagrama Use case – cadastro de rua.....	46
Figura 22 – diagrama Use case – cadastro de Funcionário .....	47
Figura 23 – diagrama Use case – cadastro de cliente .....	48
Figura 24– diagrama Use case – cadastro de produto.....	49
Figura 25 – diagrama Use case – cadastro de fornecedor .....	50
Figura 26 – diagrama Use case – cadastro de cargo .....	51

Figura 27 – diagrama Use case – Carrinho de Compra .....	52
Figura 28– Diagrama de use case – venda de produto .....	53
Figura 29 – diagrama Use case – Compra de Produtos.....	54
Figura 30 – diagrama Use case – relatório de compra por produto .....	55
Figura 31 – diagrama Use case – produtos vendidos no dia.....	56
Figura 32 – diagrama Use case – produtos do estoque .....	57
Figura 33 – diagrama Use case – relação de vendas por clientes.....	58
Figura 34 – diagrama Use case – relação de vendas mensais por cliente .....	59
Figura 35 – diagrama Use case – relação de produtos comprados por mês .....	60
Figura 36 – diagrama Use case – relação de produtos comprados por fornecedor .....	61
Figura 37 – diagrama classe .....	62
Figura 38 – diagrama de Seqüência – Funcionário .....	63
Figura 39 – diagrama de Seqüência - Cliente .....	63
Figura 40 – Tela Inicial do Web Site .....	64
Figura 41 – Tela para Cadastro de Produtos .....	65
Figura 42 – Tela de Relatório do fluxo de caixa .....	66
Figura 43 – Tela para Cadastro de Clientes.....	67
Figura 44 – Tela para Relatório de todos os produtos do Estoque.....	68

## TABELAS

Tabela 01 – Recursos necessarios .....	38
Tabela 02 – Orçamento do projeto .....	38
Tabela 03 – Lista de eventos .....	39
Tabela 04 – Tabela de caso de uso (use case) continuação .....	41
Tabela 05 – Cadastro de pais .....	42
Tabela 06 – cadastro de estado .....	43
Tabela 07 – Cadastro de cidade .....	44
Tabela 08 – cadastro de bairro.....	45
Tabela 09 – cadastro de rua .....	46
Tabela 10 – cadastro de funcionário.....	47
Tabela 11 – Cadastro de cliente.....	48
Tabela 12 – cadastro de produto .....	49
Tabela 13 – Cadastro de fornecedor.....	50
Tabela 14 – cadastro de cargo .....	51
Tabela 15 – Carrinho de Compra.....	52
Figura 16 – Venda de Produtos .....	53
Tabela 17 – Compra de Produtos.....	54
Tabela 18 – relatório de compra por produto .....	55
Tabela 19 – produtos vendidos no dia .....	56
Tabela 20 – produtos do estoque .....	57
Tabela 21 – relação de vendas por clientes .....	58
Tabela 22 – relação de vendas mensais por cliente.....	59
Tabela 23 – relação de produtos comprados por mês.....	60
Tabela 24 – relação de produtos comprados por fornecedor .....	61

## SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>15</b>
1.1. OBJETIVO .....	16
<b>2. A ORIENTAÇÃO A OBJETOS .....</b>	<b>17</b>
2.1. ABSTRAÇÃO .....	17
2.2. HERANÇA .....	18
2.3. POLIMORFISMO .....	18
2.4. OBJETOS .....	19
2.5. CLASSES .....	19
2.5.1. SERVIÇOS .....	19
2.5.2. ATRIBUTOS .....	20
2.5.3. MÉTODOS .....	20
2.6. POLIMORFISMO .....	21
2.7. ENCAPSULAMENTO .....	21
<b>3. UNIFIED MODELING LANGUAGE (UML) .....</b>	<b>22</b>
3.1. OBJETIVOS DA UML .....	22
3.2. DIAGRAMAS .....	23
3.2.1. DIAGRAMAS DE USE CASE .....	23
3.2.2. DIAGRAMAS DE CLASSE .....	24
3.2.3. DIAGRAMAS DE OBJETOS .....	24
3.2.4. DIAGRAMAS DE ESTADO .....	25
3.2.5. DIAGRAMAS DE SEQÜÊNCIA .....	26
3.2.6. DIAGRAMAS DE COLABORAÇÃO .....	27
3.2.7. DIAGRAMA DE ATIVIDADES .....	28
3.2.8. DIAGRAMA DE COMPONENTES .....	29
3.2.9. DIAGRAMAS DE IMPLANTAÇÃO .....	30
<b>4. FERRAMENTAS PARA O DESENVOLVIMENTO DO SISTEMA .....</b>	<b>31</b>
4.1. LINGUAGEM DE PROGRAMAÇÃO .....	31
4.1.1. VISUAL STUDIO .....	31
4.2. BANCO DE DADOS .....	32
4.2.1. SQL SERVER 2005 EXPRESS .....	32
4.3. FERRAMENTA DA ANÁLISE .....	33
4.3.1. NETBEANS .....	33
4.3.2. EA (ENTERPRISE ARCHITECT) .....	34
<b>5. ANÁLISE E ESPECIFICAÇÃO DO SISTEMA .....</b>	<b>35</b>
5.1. LEVANTAMENTO DE REQUISITOS .....	35
5.2. DECLARAÇÃO DE OBJETIVOS .....	35
5.3. CRONOGRAMA DE ATIVIDADES .....	35
5.4. ESPECIFICAÇÃO DE PRAZOS .....	36
5.4.1. ESTRUTURA ANALÍTICA DO PROJETO .....	36
5.4.2. SEQUENCIAMENTO DAS ATIVIDADES .....	37
5.4.3. RECURSOS NECESSÁRIOS PARA DESENVOLVIMENTO DO SISTEMA .....	38
5.4.4. ESTIMATIVA DE CUSTOS .....	38
5.5. LISTA DE EVENTOS .....	39
5.6. ESCOPO DO SISTEMA .....	40
5.7. TABELA DE CASO DE USO (USE CASE) .....	41
<b>6. ANÁLISE ORIENTADA A OBJETOS .....</b>	<b>42</b>
6.1. DIAGRAMA DE CASOS DE USO (USE-CASE) .....	42
6.2. DIAGRAMA DE CLASSE .....	62
6.3. DIAGRAMA DE SEQÜÊNCIA .....	63

<b>7. INTERFACES DO SISTEMA .....</b>	<b>64</b>
<b>8. CONCLUSÃO .....</b>	<b>69</b>
<b>9. BIBLIOGRAFIA .....</b>	<b>70</b>

# 1. INTRODUÇÃO

Está comprovado que as empresas que já informatizou seu sistema constataram que informatização só aumenta a produtividade e a rentabilidade dos negócios. Atualmente não é questionada se informatiza ou não uma empresa, e sim a forma como se deve fazer o mesmo, investir de forma ordenada só aumentará os lucros.

Atualmente toda empresa tem que levar em consideração a questão da informatização da empresa, pois aquela que não se informatizar certamente terá vida curta, essa informatização deve ser feita para ajudar a melhorar os seus negócios, agilizar processos, atingir seus objetivos e ter um diferencial no mercado.

Esse investimento que muitas empresas vêm fazendo em informática deve ser de forma organizada e ordenada, sempre observando a viabilidade econômica da mesma, deve agregar sempre, benefícios aos produtos e serviços, onde o retorno econômico será maior e principalmente oferecer um melhor atendimento aos seus clientes. Há alguns anos na cidade de Quatá, surgiu uma pequena empresa, empresa que desenvolve sistemas, Manutenção e venda de computadores.

Atualmente essa empresa trabalha no ramo de informática, na venda e manutenção, onde o sistema terá de ter uma atuação maior no ramo de vendas de materiais para a informática. Sempre levando em consideração as rotinas de trabalho da empresa que envolve venda de matérias para a informática observou-se sua maior deficiência, o sistema fará o controle das vendas da empresa. Levando em consideração os problemas no departamento de vendas foi constatado que, era a hora de informatizar o setor.

O projeto a ser implantado na empresa CAS info atuante no ramo da informática, localizada na Rua João Padovan, numero 20, no Município de Quatá, onde atua desde 2007, que visa atender os clientes sempre, e cada vez mais, com muita qualidade, educação e agilidade. Além de venda de matérias para a informática a empresa trabalha também com manutenção de computadores, mas esse assunto não será considerado, pois o mesmo já está informatizado. Nesse sentido o sistema deverá controlar também, além da venda o estoque dos produtos da empresa.

## **1.1. Objetivo**

O trabalho tem como objetivo desenvolver um sistema que forneça apoio a tomada de decisão e mostre confiança ao usuário, fazendo uso da Orientação a objetos e paralelamente com a famosa técnica UML (Linguagem de Modelagem Unificada), para ajudar a fazer a modelagem de todo o sistema. O sistema, esse terá funcionalidades de gerar cadastros, movimentar produtos com entrada e saída de produto, o mesmo criará relatórios de vários aspectos, visando sempre maximizar as vendas e otimizar a produção da empresa.

O Sistema terá como funcionalidade a geração de Cadastros, a movimentação de caixa, ele terá uma ampla opção de relatórios, com o objetivo de assim fazer projeções de lucros e despesas, o mesmo irá ajudar a controlar as vendas da loja, controlando o fluxo financeiro da loja, visando sempre máxima qualidade no controle de vendas.

O sistema visa facilitar, tanto o cliente quanto os funcionários da empresa. A informatização do setor será muito importante, por que assim os gerentes terão maior conhecimento dos lucros da empresa, em relação à venda de produtos da loja. O trabalho tem como objetivo principal, oferecer um sistema que aperfeiçoe as vendas gerando assim mais lucro para a empresa, de fácil entendimento para usuários.

## 2. A ORIENTAÇÃO A OBJETOS

O termo Programação Orientada a Objetos foi criado por Alan Kay, autor da linguagem de programação Smalltalk. Mas mesmo antes da criação do Smalltalk, algumas das idéias da POO já eram aplicadas, sendo que a primeira linguagem a realmente utilizar estas idéias foi a linguagem Simuda 67, por Ole Johan Dahl e Kristen Nygaard em 1967. Note que este paradigma de programação já é bastante antigo, mas só agora vem sendo aceito realmente nas grandes empresas de desenvolvimento de Software. Alguns exemplos de linguagens modernas utilizadas por grandes empresas em todo o mundo que adotaram essas idéias: Java, C#, C++, Object Pascal (Delphi), Ruby, Python, Lisp, etc.

A maioria delas adota as idéias parcialmente, dando espaço para o antigo modelo procedural de programação, como acontece no C++ por exemplo, onde temos a possibilidade de usar POO, mas a linguagem não força o programador a adotar este paradigma de programação, sendo ainda possível programar da forma procedural tradicional. Este tipo de linguagem segue a idéia de utilizar uma linguagem previamente existente como base e adicionar novas funcionalidades a ela. Outras são mais "puras", sendo construídas do zero focando-se sempre nas idéias por trás da orientação a objetos como é o caso das linguagens Smalltalk, Self, IO e famosa e ótima linguagem, o Java, onde TUDO é orientado a objetos.

### 2.1. Abstração

A abstração consiste na valorização do que é mais importante para o sistema e ignorando aspectos sem importância, deixando de lado os seus detalhes. Existem dois tipos de abstração:

Abstração de Dados seria uma forma de definir um tipo de dado por seus atributos. A manipulação dos dados é feita através de seus métodos.

Abstração de Procedimentos Consiste em operar procedimento definido como algo único, mesmo utilizando outros procedimentos internos. Usa-se a abstração de procedimentos no momento em que uma função se divide em sub-funções, que por sua vez destrói quando acaba a função do mesmo.

## **2.2. Herança**

Fazendo uma comparação da Herança com a Biologia, o ser Humano é um animal. E tem suas características que são mais conhecidos por atributos, e pode realizar todas as ações são os famosos métodos de um animal. No entanto, ele tem algumas características e ações que só ele pode realizar. Em momentos como este, se utilizado da herança. Uma classe pode ser estendida para outras classes, estendo assim suas características para a outra classe e adicionar algumas coisas a mais. Desta forma, a classe Ser Humano será uma especialização, ou seja, a subclasse da classe Animal.

A classe Animal seria a classe pai da serHumano, e logicamente, a classe SerHumano seria a classe filha da Animal. Uma classe pode sempre ter vários filhos, mas normalmente as linguagens de programação orientadas a objetos exigem que cada classe filha tenha apenas uma classe pai. A linguagem C++ permite que uma classe herde as características de varias classes (herança múltipla), mas C++ não é um bom exemplo quando se está falando sobre conceitos de POO.

## **2.3. Polimorfismo**

Um dos conceitos mais complicados de se entender, e também um dos mais importantes, é o Polimorfismo. O termo polimorfismo é originário do grego e significa "muitas formas". Na orientação a objetos, isso significa que um mesmo tipo de objeto, sob certas condições, pode realizar ações diferentes ao receber uma mesma mensagem. Ou seja, apenas olhando o código fonte não sabemos exatamente qual será a ação tomada pelo sistema, sendo que o próprio sistema é quem decide qual método será executado, dependendo do contexto durante a execução do programa.

## **2.4. Objetos**

Em uma linguagem procedural, a instância de uma estrutura é chamada de variável. Já uma instância de uma classe é chamada de objeto. A classe informa a cada objeto suas características (dados) e seu comportamento (métodos). Suas características são informadas pela declaração de variáveis de tipo primitivo (int, float, etc...) ou classes anteriormente definidas. Seus métodos são definidos por funções, possibilitando a “conversa” entre dois objetos.

## **2.5. Classes**

Uma classe definida pelo usuário, contém a especificação da classe, identificando seus atributos (dados) e seus métodos (comportamento da classe). A definição de uma classe não é responsável pela criação de nenhuma instância, portanto não ocupa uma parte da memória. Uma classe em linguagem orientada a Objeto tem possibilidade de combinar um único registro de importantes informações de um determinado objeto. [ Tatyana Bitencourt]

### **2.5.1. Serviços**

Um serviço pode ser definido com um trabalho realizado sobre os dados de um objeto, de certa forma serviços de um objeto são trabalhos públicos. Quando solicitado por outro objeto por meio de um paradigma de passagem de mensagem, esses serviços são definidos por protótipos. Protótipos são definidos por duas partes: nome das mensagens. [Richard C. Lee e William M.Tepfenhart]

### 2.5.2. Atributos

É uma abstração de uma característica individual aplicável ao domínio de negócios e é possuído por todas as entidades que, em seus próprios termos, eram abstraídas como objetos. Sob uma perspectiva técnica, um atributo é alguma variável (item de dados ou alguma variação de estado) para qual cada objeto (instância) tem seu próprio valor. Cada atributo deve ser provido de um nome exclusivo dentro de cada objeto/classe. Existem quatro tipos de atributos: descritivo, de identificação, de informação e o de estados referenciais. [Tatyana Bitencourt]

- Atributos Descritivos: são fatos intrínsecos a cada entidade;
- Atributos Identificação: são utilizados para nomear ou rotular uma entidade;
- Atributos de informação: são utilizados para manter uma história da entidade;
- Atributos Referenciais: são fatos que vinculam um objeto.

### 2.5.3. Métodos

É um conjunto detalhado de operações que um objeto executa quando outro objeto solicita um serviço. Entretanto, por definição, um comportamento é um conjunto de ações que um objeto é responsável por exibir alternativamente. Um método especifica um comportamento de um objeto assim, o método é similar a uma função da decomposição funcional. [ Lee & Tepfenhart, 2001]

## **2.6. Polimorfismo**

Polimorfismo caracteriza a criação de funções com um mesmo nome, mas códigos diferentes, facilitando a extensão de sistemas. Um tipo de polimorfismo é a redefinição de métodos para uma classe derivada. Para que isso aconteça, o método deve possuir o mesmo nome, tipo de retorno e argumentos do método sobrescrito.

## **2.7. Encapsulamento**

Encapsulamento consiste em esconder detalhes (dados) e oferecer uma interface comum aos objetos (métodos). Toda vez que uma alteração deve ser feita em alguma variável de instância, deve ser feita através de um método da classe. Caso alguma modificação ocorra em alguma variável de instância, sabe-se exatamente quais métodos interagiram com a variável, prevenindo alterações acidentais.

É permitido ao usuário ver apenas os dados necessários, todo o resto é mantido escondido, através de especificadores de acesso (public, private, protected). A utilização de encapsulamento também favorece a reusabilidade e manutenção do código.

### 3. UNIFIED MODELING LANGUAGE (UML)

A UML tem origem na compilação das "melhores práticas de engenharia" que provaram ter sucesso na modelagem de sistemas grandes e complexos. Sucedeu aos conceitos de Booch, OMT (Rumbaugh) e OOSE (Jacobson) fundindo-os numa única linguagem de modelagem comum e largamente utilizada. A UML pretende ser a linguagem de modelagem padrão para modelar sistemas concorrentes e distribuídos. A UML ainda não é um padrão da indústria, mas esse objetivo está a tomar forma sob os auspícios do Object Management Group (OMG).

O OMG pediu informação acerca de metodologias orientadas a objetos que pudessem criar uma linguagem rigorosa de modelagem de software. Muitos líderes da indústria responderam na esperança de ajudar a criar o padrão. Os esforços para a criação da UML tiveram início em outubro de 1994, quando Rumbaugh se juntou a Booch na Rational. Com o objetivo de unificar os métodos, Booch e OMT, decorrido um ano de trabalho, foi lançado, em outubro de 1995, o esboço da versão 0.8 do *Unified Process* - Processo Unificado (como era conhecido). Nesta mesma época, Jacobson se associou à Rational e o escopo do projeto da UML foi expandido para incorporar o método OOSE. Nasceu então, em junho de 1996, a versão 0.9 da UML.

#### 3.1. Objetivos da UML

A UML tem como objetivo:

- Ser umas linguagens visuais, expressivas, que se revele simples e eficaz.
- Não depende de nenhuma linguagem de programação.
- Não depende de processos
- É flexível.

## 3.2. Diagramas

Na UML existe vários tipos de diagramas, mais especificamente existem 9 tipos de diagrama: de classe, objetos, seqüência, colaboração, caso de uso, implementação, atividades, estado e componentes. Em todos os diagramas, os conceitos são representados como, símbolos, e os relacionamentos entre conceitos são representados como trajetórias (linha) conectando símbolos cada um desses elementos terá um nome. [Lee & Tepfenhart, 2001]

### 3.2.1. Diagramas de Use Case

Os modelos de Casos de Uso são descritos através de Diagramas de Casos de Uso na UML. De uma forma geral, cada projeto de *software* conterà um Diagrama de Casos de Uso. Para sistemas mais extensos, é possível decompor o diagrama em um conjunto de subdiagramas. Uma use case é composta por duas partes os atores e os casos de uso que se relacionam.

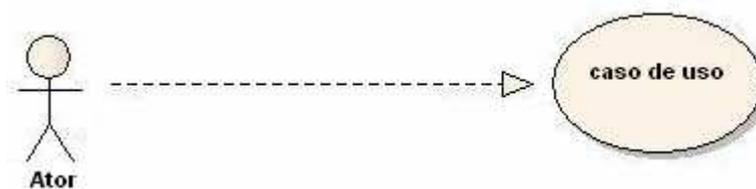


Figura 01 – exemplo de caso de uso

### 3.2.2. Diagramas de classe

As classes e objetos são as principais primitivas ou elementos de composição para *softwares* orientados a objetos. Na verdade, um sistema orientado a objetos é composto por classes e por um conjunto de objetos que colaboram ou interagem para execução dos serviços e processos, oferecidos pelo sistema. Classes podem ter atributos que geralmente são membros de dados primitivos de objetos. A visibilidade de atributos e operações de outros objetos pode ser definida, igualmente as suas assinaturas, incluindo tipos, valores padrão, parâmetros, tipos de parâmetros tipo de retorno.

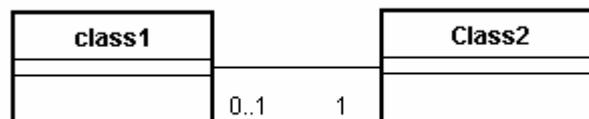


Figura 02 – Exemplo de Diagrama de classe

### 3.2.3. Diagramas de Objetos

O diagrama de objetos é uma variação do diagrama de classe. A diferença é que ao contrário do diagrama de classe o diagrama de objetos mostra os objetos que foram instanciados da classe. Portanto existe uma grande semelhança com o diagrama de classe, eles são escritos com seus nomes sublinhados onde todas as instâncias do relacionamento são mostradas.

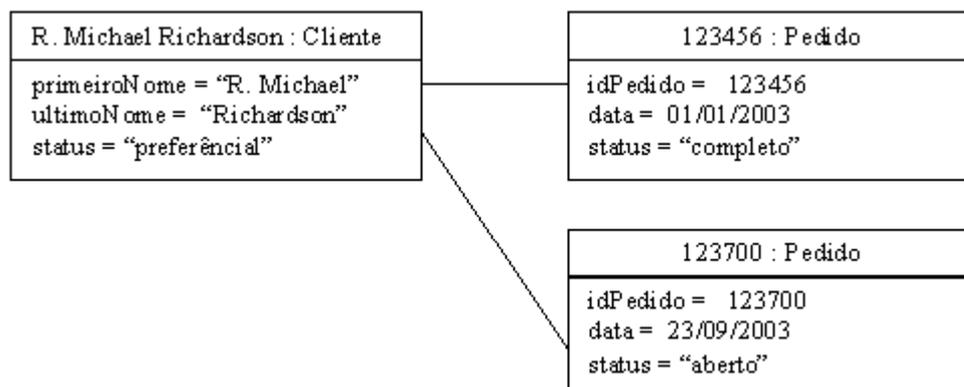


Figura 03 – Exemplo de Diagrama de Objetos

### 3.2.4. Diagramas de Estado

De acordo com a UML, a especificação da dinâmica do sistema deve ser feita através de diagramas de estados. Constrói-se um diagrama de estados descrevendo o comportamento de cada classe e eventuais diagramas complementares para descrever a dinâmica de todo o sistema ou de certos módulos. Deve-se observar que, no caso das classes, o diagrama de estados deve reunir o comportamento de uma classe com todas as suas responsabilidades, ou seja, com o seu comportamento completo em todos os casos de uso. Desta forma, o diagrama de estados de uma classe ou uma descrição global do comportamento dos objetos desta classe em todo o sistema. [ docs.kde.org]

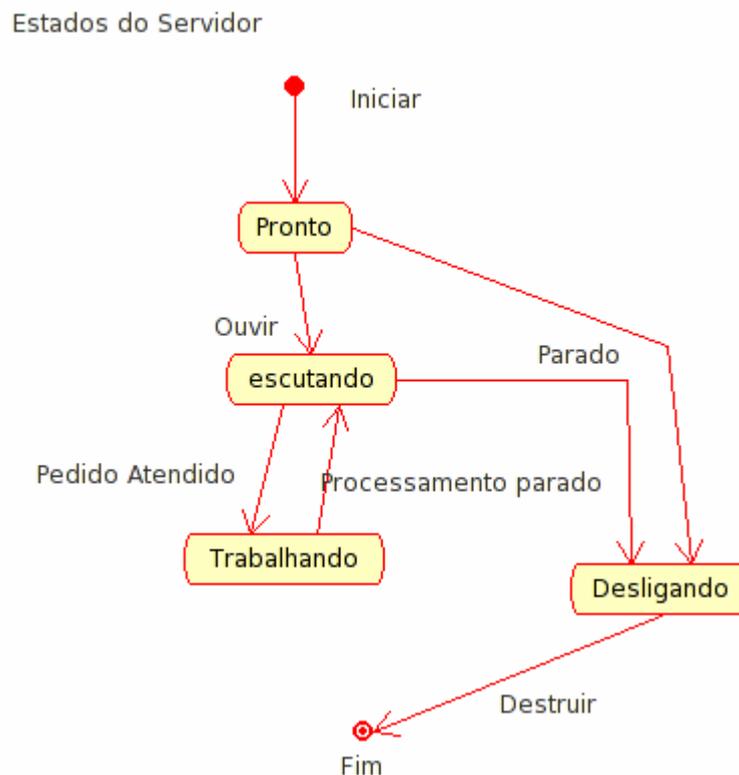


Figura 04 – Exemplo de Diagrama de Estado

### 3.2.5. Diagramas de Seqüência

Os Diagramas de Seqüência mostram a troca de mensagens entre diversos Objetos, em uma situação específica e delimitada no tempo. Objetos são instâncias de classes. Diagramas de Seqüência colocam ênfase especial na ordem e nos momentos nos quais mensagens para os objetos são enviadas. Em Diagramas de Seqüência objetos são representados através de linhas verticais tracejadas, com o nome do Objeto no topo. O eixo do tempo é também vertical, aumentando para baixo, de modo que as mensagens são enviadas de um Objeto para outro na forma de setas com a operação e os nomes dos parâmetros.[docs.kde.org]

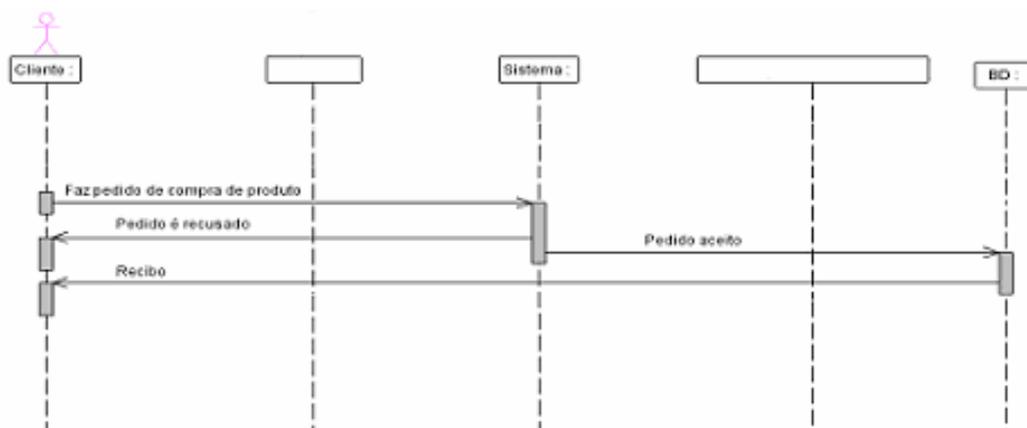


Figura 05 – Exemplo de Diagrama de Seqüência

### 3.2.6. Diagramas de Colaboração

Diagramas de Colaboração mostram as interações que ocorrem entre os objetos participantes numa situação específica. Isto é mais ou menos a mesma informação mostrada pelos Diagramas de Seqüência, Em Diagramas de Colaboração as mensagens enviadas de um objeto para outro são representadas por setas, mostrando o nome da mensagem, parâmetros, e a seqüência da mensagem. São especialmente indicados para mostrar um fluxo ou situação específica do programa e são um dos melhores tipos de diagrama para rapidamente demonstrar ou explicar um processo na lógica do programa. [docs.kde.org]

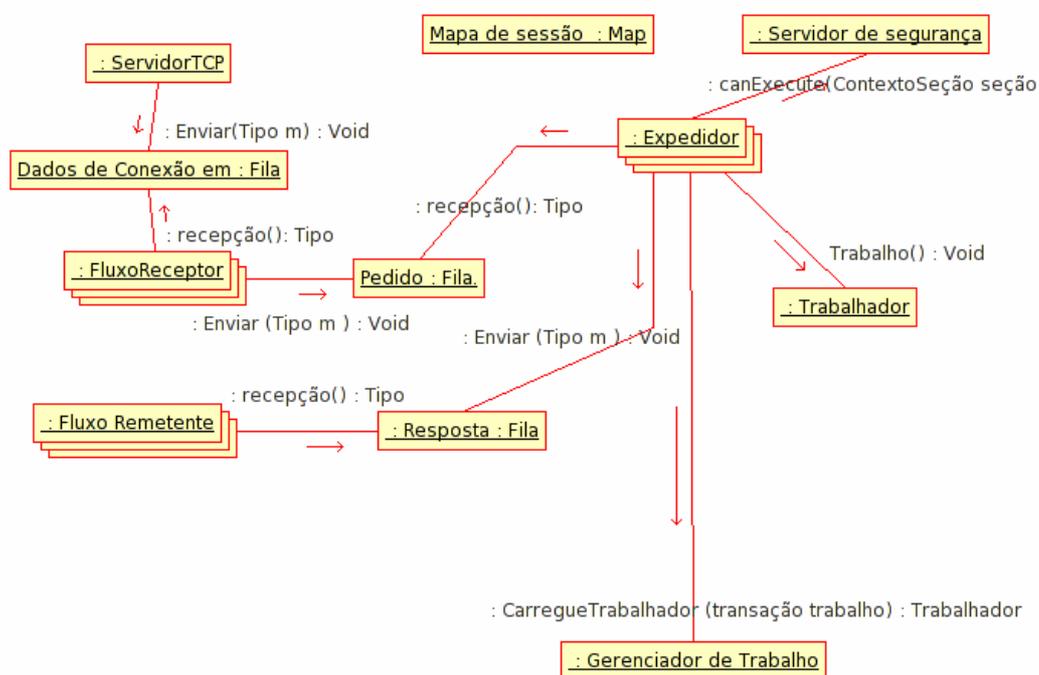


Figura 06 – Exemplo de Diagrama de Colaboração

### 3.2.7. Diagrama de Atividades

O Diagrama de Atividade descreve a seqüência de atividades num sistema com a ajuda as Atividades. Eles são uma forma especial de Diagramas de Estado, que somente (ou principalmente) contém Atividades. Diagramas de Atividade são similares as Diagramas de Fluxo de procedimentos, com a diferença de que todas as Atividades são claramente anexas aos Objetos, eles são sempre associados a uma Classe, uma Operação ou um Caso de Uso. [docs.kde.org]

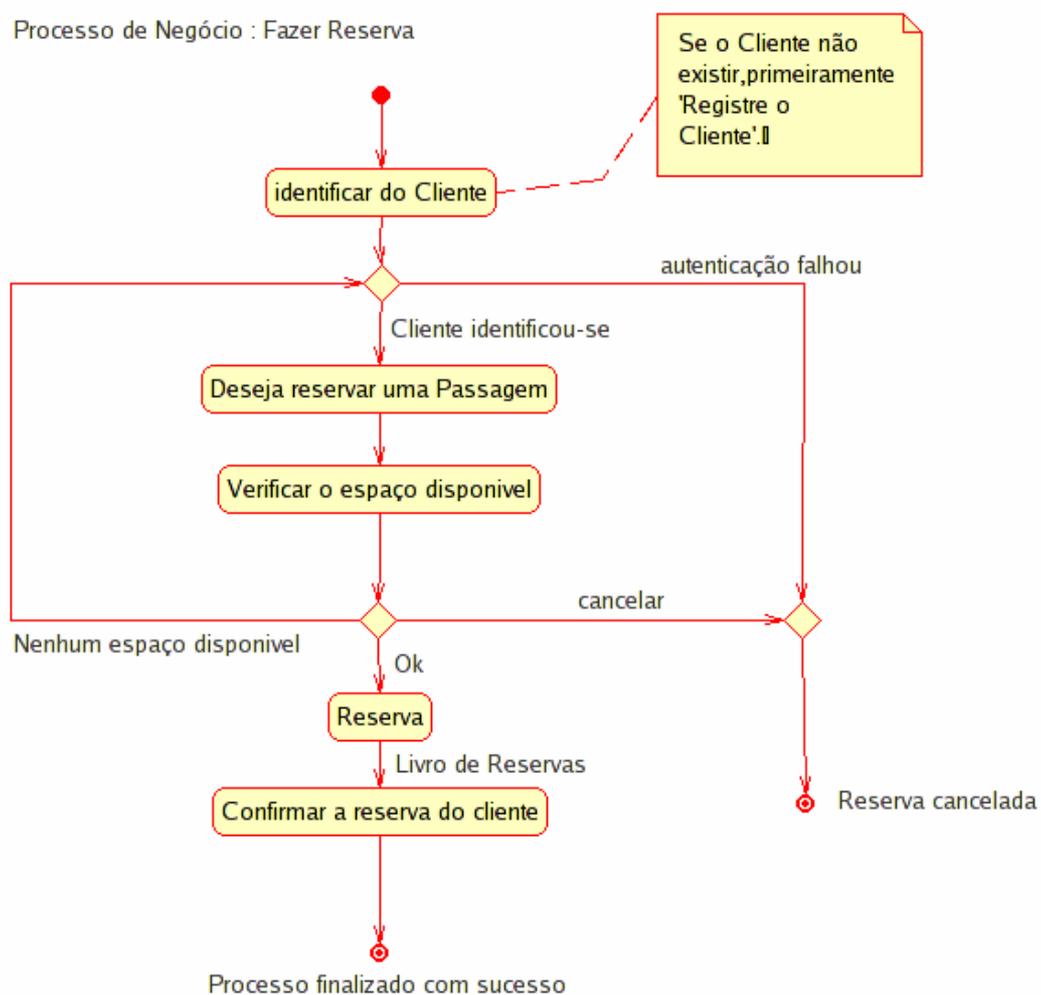


Figura 07 – Exemplo de Diagrama de Atividades

### 3.2.8. Diagrama de Componentes

O diagrama de componentes descreve a organização e a dependência entre componentes de implantação do Software. Ele mostra a estrutura dos componentes próprios e reais que serão utilizados para implementação do sistema.[Lee & Tepfenhart, 2001]

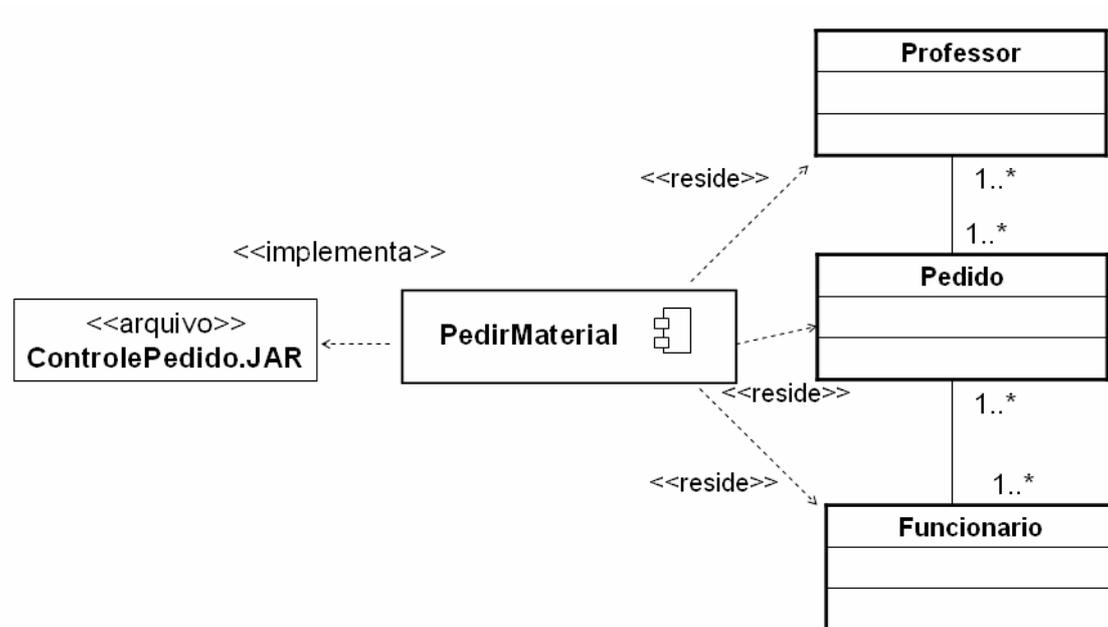


Figura 08 – Exemplo de Diagrama de Componentes

### 3.2.9. Diagramas de Implantação

Ele descreve a configuração do processamento de elementos e o mapeamento de componentes de implementação do software no decorrer do processamento dos elementos. O diagrama de implantação mostra a instancia dos nós do processador, suas interconexões, instancias dos componentes que executarão neles, e a dependência entre as instancias. [ Lee & Tepfenhart, 2001]

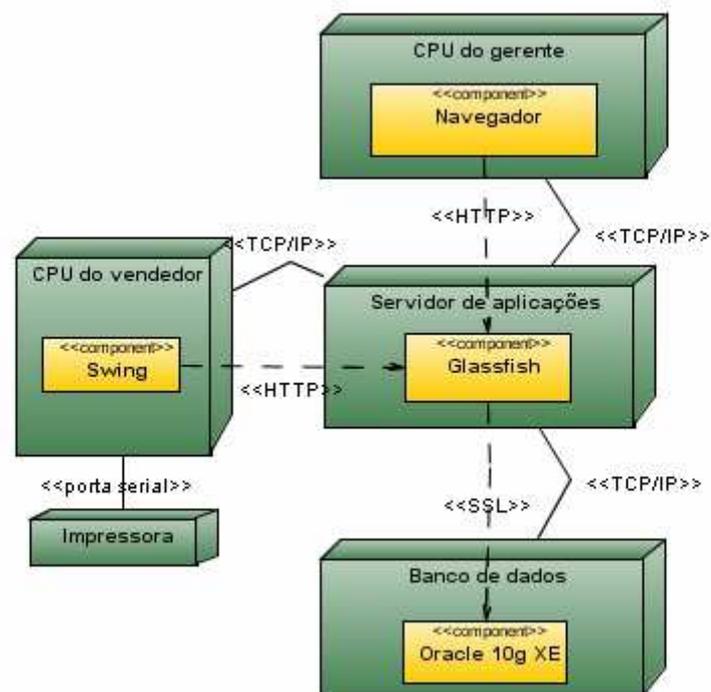


Figura 09 – Exemplo de Diagrama de Implantação

## 4. FERRAMENTAS PARA O DESENVOLVIMENTO DO SISTEMA

### 4.1. Linguagem de Programação

#### 4.1.1. Visual Studio

A estrutura da Microsoft é uma nova e completa maneira de programar na plataforma Microsoft. Os desenvolvedores da Microsoft sempre se preocupam principalmente com threads e memória (esse é basicamente o modelo de programação API). Esse modelo perdurou em todas as áreas de desenvolvimento, inclusive no desenvolvimento da Web, colocando um fardo pesado sobre os programadores.

O.NET é elaborado sobre a noção de tipos gerenciados. Os desenvolvedores códigos Windows clássicos (e códigos Web) escrevem classes usando a linguagem C++ ou Visual Basic. De certa forma, os tipos são semelhantes à noção de classes em C++ em que tipos são unidades de estados com funcionalidades anexadas a eles. Porém, a semelhança termina aí. Enquanto era incumbência do desenvolvedor gerar instâncias de classes, hoje os tipos são totalmente gerenciados pelos serviços do .NET Runtime, o Common Language Runtime (CLR).

Veja a área de trabalho do Visual Studio:

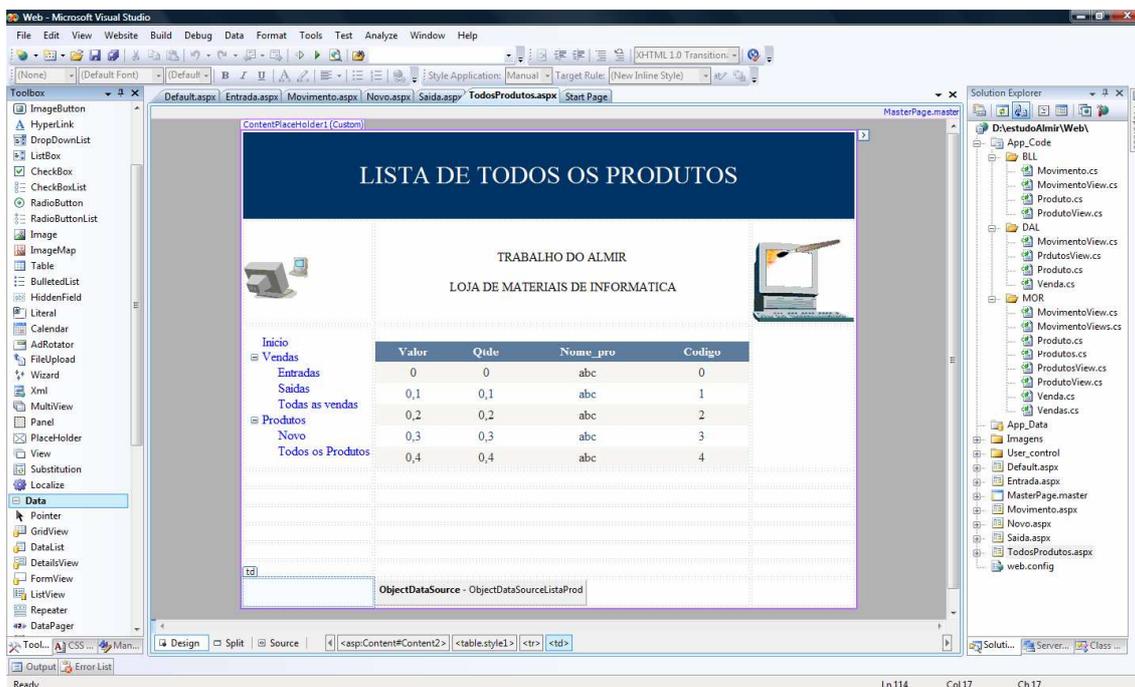


Figura 10 – Área de Trabalho do Visual Studio

## 4.2. Banco de Dados

### 4.2.1. SQL Server 2005 express

O sistema fará uso do SQL Server 2005 express para gerenciar a base de dados do web site, o SQL Server 2005 express é um SGBD (sistema gerenciador de banco de dados)

Veja a área de trabalho do SQL Server:

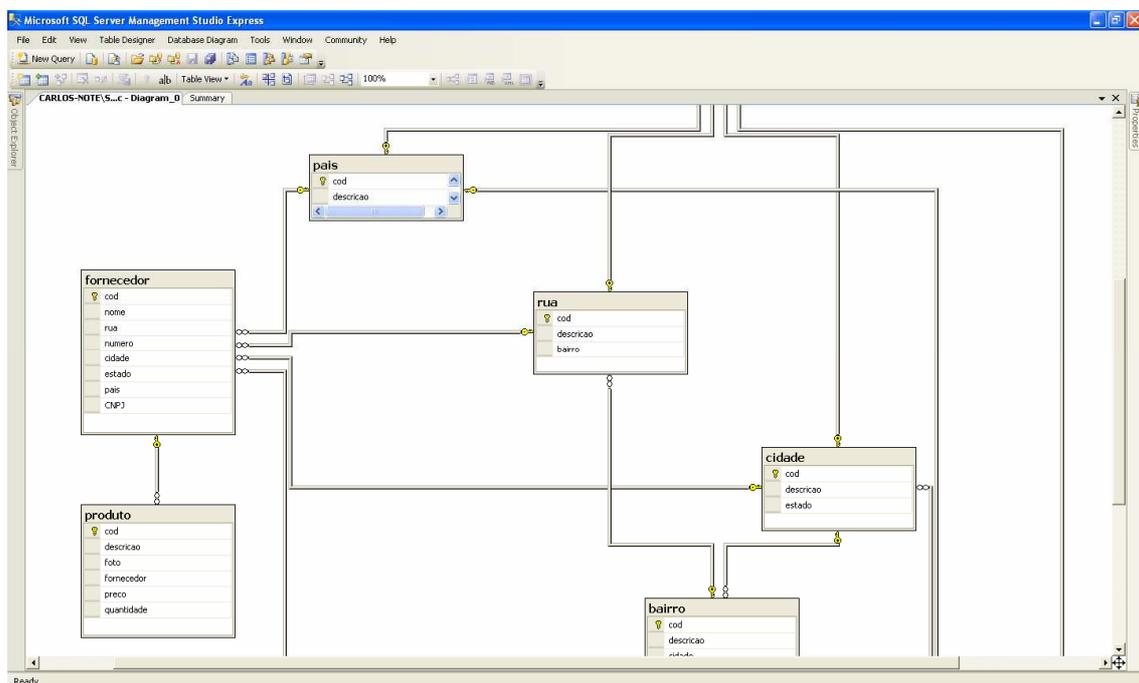


Figura 11 – Área de trabalho do SQL Server 2005 express

### 4.3. Ferramenta da Análise

#### 4.3.1. Netbeans

No Brasil, umas das ferramentas de desenvolvimento Java, que junto à plataforma Eclipse, sendo consagrada e usada pelos desenvolvedores é a IDE (Interface Development Environment) NetBeans.

O NetBeans é uma IDE madura, desenvolvida pela Sun, que concorre diretamente com a plataforma Eclipse, embora digam que não, e que vem se destacando cada vez mais por ser uma complexa, gratuita e ao mesmo tempo uma ferramenta de desenvolvimento multi plataforma.

Veja a seguir a área de trabalho do netBeans onde foi desenvolvido o diagrama de seqüência.

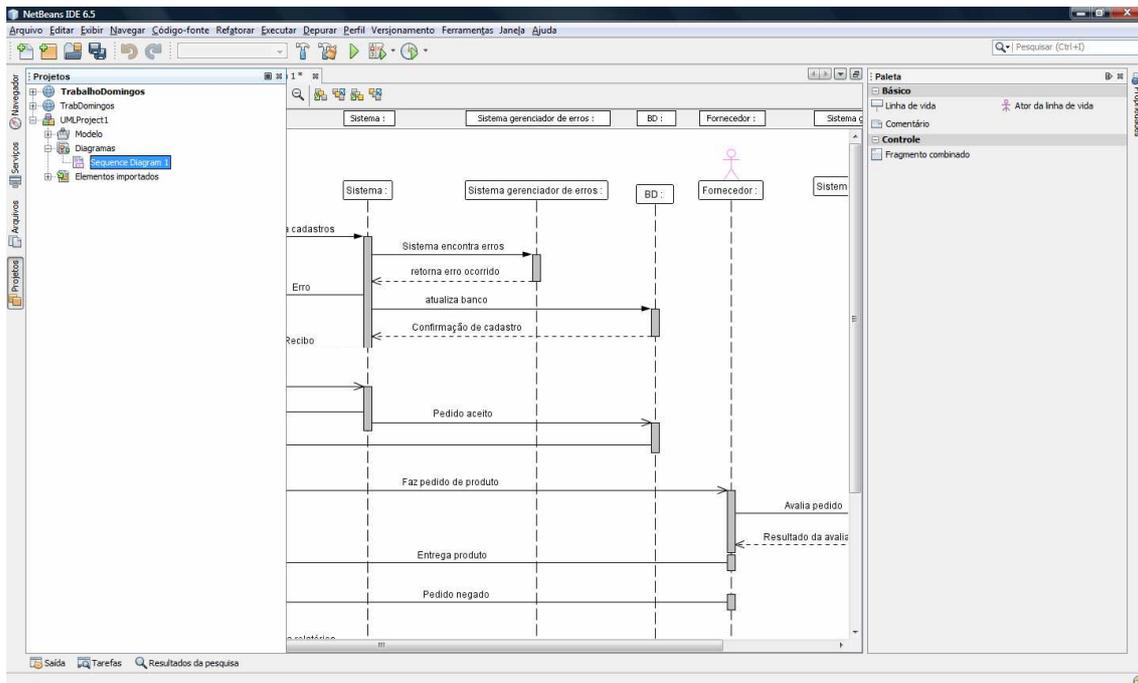


Figura 12 – Netbeans

### 4.3.2. EA (Enterprise Architect)

Enterprise Architect é uma ferramenta de modelagem de sistemas, concepção da plataforma é baseada em UML e normas associadas. Ágil, intuitivo e plenamente integrado, poderoso domínio específico com características próprias. Um Sistema de grandes soluções para visualização, análise, modelagem de sistemas, é utilizado para análise e manutenção de vários sistemas, software, processos e arquiteturas. Tem, como base, uma larga experiência comprovada por muitos usuários no mundo inteiro. O Enterprise Architect é escalável, comparativo e com ferramentas que é uma ótima opção para a arquitetura de se próximo projeto.

Veja agora como é a área de trabalho do EA:

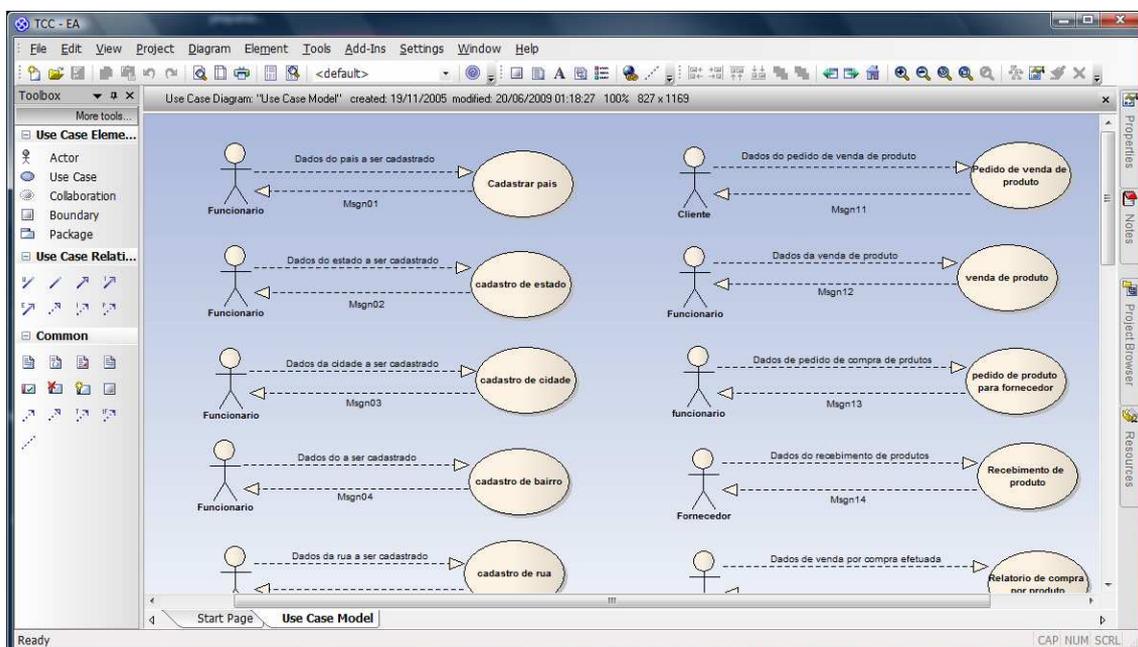


Figura 13 – EA (Enterprise Architect)

## 5. ANÁLISE E ESPECIFICAÇÃO DO SISTEMA

### 5.1. Levantamento de Requisitos

Cadastros de Pais, Estado, Cidade, Bairro, Cliente, Fornecedor, Funcionário, Cargo. O sistema deve ter um Controle sobre as vendas da empresa, o estoque, controlando tudo que entra e que sai. O mesmo terá muitos Relatórios de vendas, compras, cliente, produtos.

### 5.2. Declaração de Objetivos

O objetivo do trabalho é implantar um sistema que supra as necessidades CAS info empresa de informática, controlando entrada e saída de produtos, estoque, para que a empresa possa agilizar suas rotinas de trabalho possibilitando assim melhor desempenho da empresa.

### 5.3. Cronograma de atividades

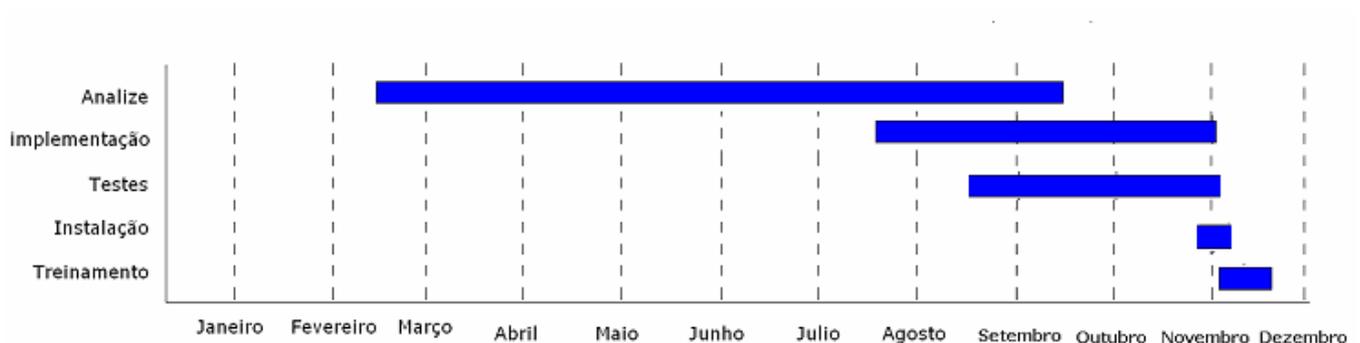


Figura 14 – Cronograma de atividades

## 5.4. Especificação de Prazos

### 5.4.1. Estrutura analítica do projeto

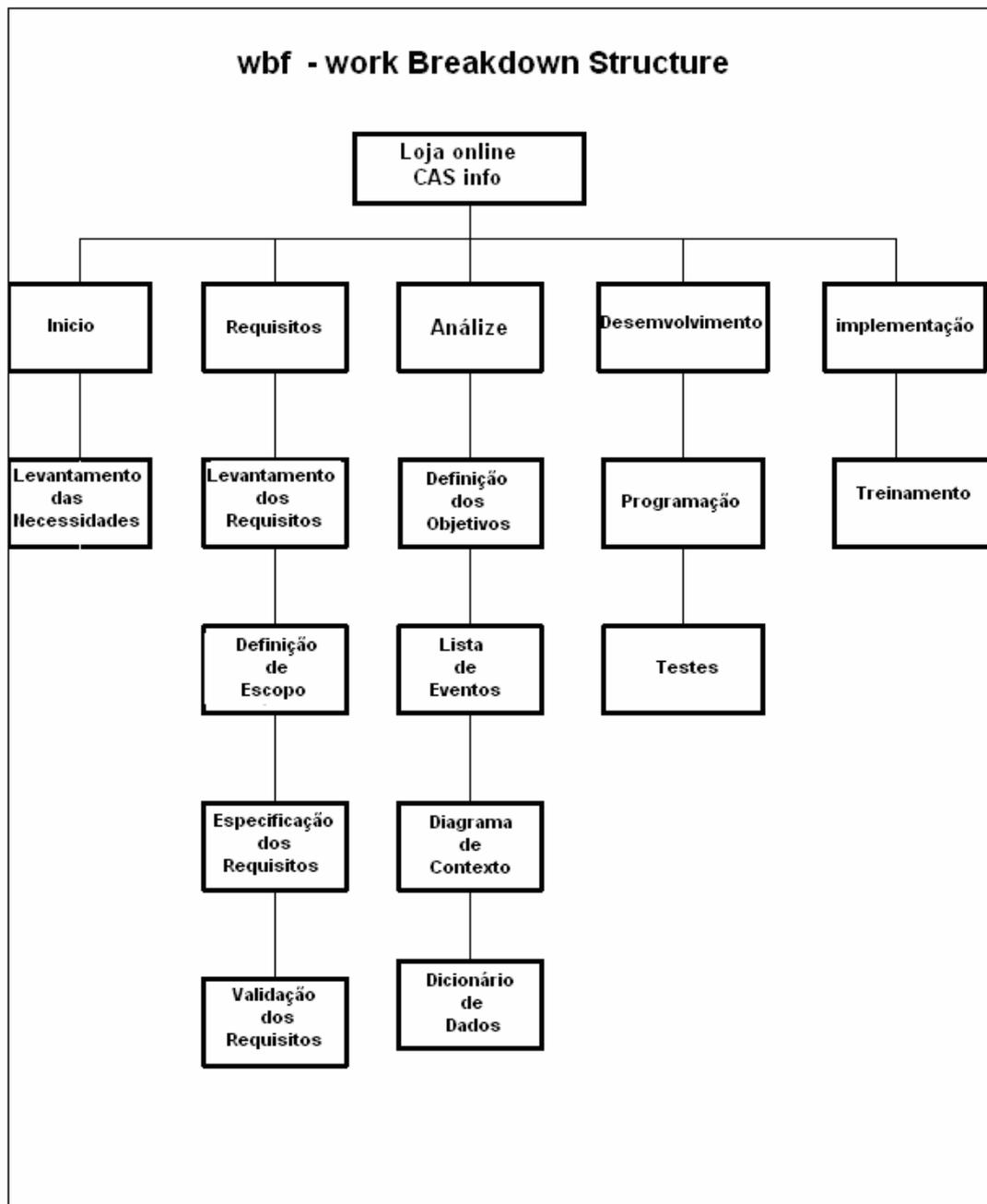


Figura 15 – WBF

## 5.4.2. Sequenciamento das Atividades

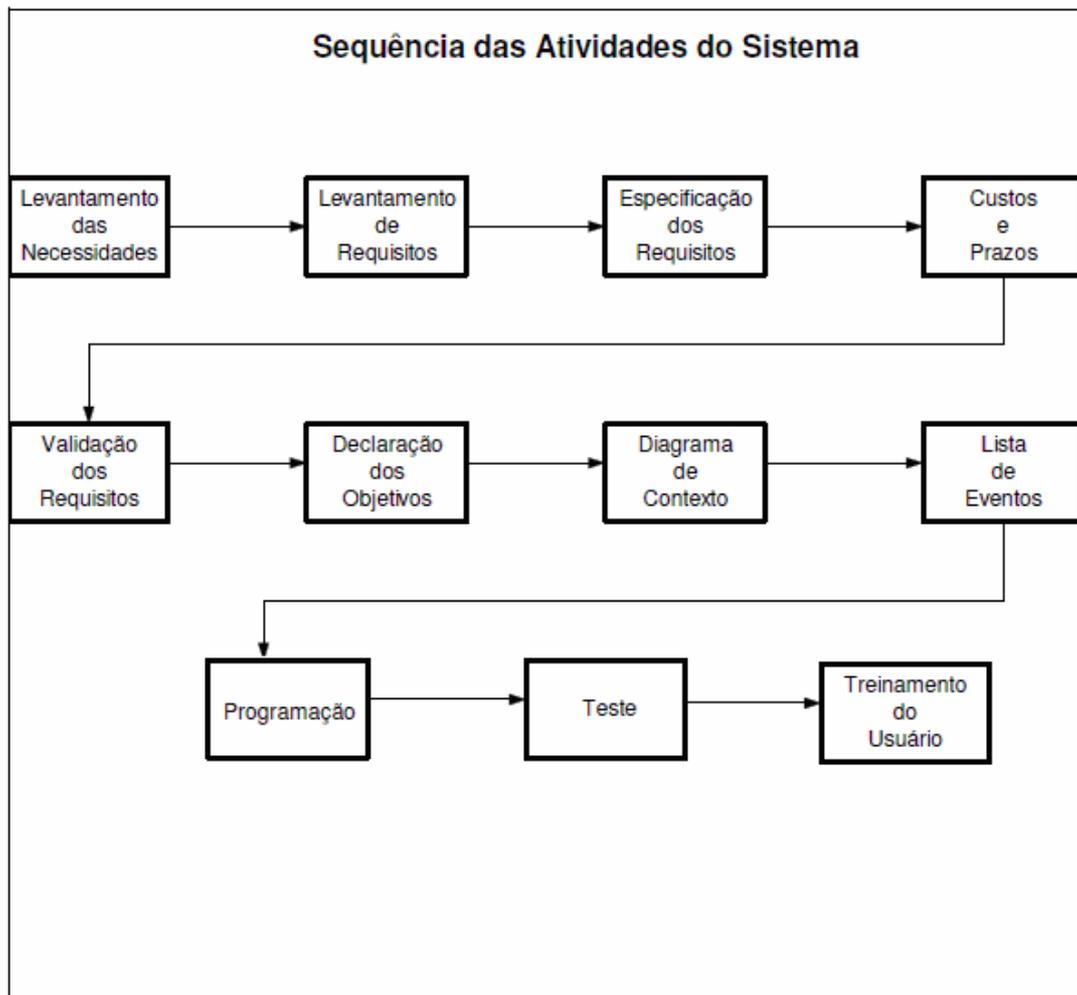


Figura 16 – Sequência de Atividades

## 5.4.3. Recursos necessários para desenvolvimento do Sistema

Pessoas	01 Desenvolvedor
Equipamentos	01 notebook celeron M HD 60 GB – 2 GB de RAM 01 Impressora
Software	Microsoft Office 2003 Microsoft Visual Studio SQL Server 2005 Express

Tabela 01 – Recursos necessarios

## 5.4.4. Estimativa de custos

**Pessoal:**  $160 * R\$ 25,00 = R\$ 4.000,00$

**Notebook:** R\$ 1.700,00

Depreciação de 2 anos:  $R\$ 1700/24 = R\$ 70,83$  Mensal e 2,36 Diário

**Microsoft Office 2003:** R\$ 500,00

Depreciação de 2 anos:  $R\$ 500,00/24 = R\$ 20,83$  mensal ou R\$ 0,69 Diário

Custo de utilização de 90 dias:  $90 * R\$ 0,69 = R\$ 62,10$

**Impressora:** R\$ 480,00

Depreciação de 2 anos:  $R\$ 480,00/24 = R\$ 20,00$  mensal ou 0,66 Diário

Custo de utilização de 30 dias:  $30 * R\$ 0,66 = R\$ 19,8$

**Total de Custos do Sistema: R\$ 5.781,9**

## 5.4.5. Orçamento do Projeto

Recursos	Valor em R\$
Pessoal	4.000,00
<b>Materiais</b>	
Notebook	1.700,00
Microsoft Office 2003	62,10
Impressora	19,8
<b>Total</b>	<b>5.781,9</b>

Tabela 02 – Orçamento do projeto

### 5.5. Lista de Eventos

<b>Numero</b>	<b>Eventos</b>
01	Funcionário cadastra Pais
02	Funcionário cadastra Estado
03	Funcionário cadastra Cidade
04	Funcionário cadastra Bairro
05	Funcionário cadastra Rua
06	Funcionário cadastra Cliente
07	Funcionário cadastra Fornecedor
08	Funcionário cadastra Funcionário
09	Funcionário cadastra Produto
10	Funcionário cadastra Cargo
11	Cliente Adiciona produto ao Carrinho
12	Cliente efetua a compra do Produto
13	Funcionário compra produto
14	Funcionário vê relatório de produto por compra
15	Funcionário vê relatório de produto vendido
16	Funcionário vê relatório de produto no estoque
17	Funcionário vê relatório de vendas por cliente
18	Funcionário vê relatório do Caixa
19	Funcionário vê relatório de produtos comprados no mês
20	Funcionário vê relatório de Compras por fornecedor

Tabela 03 – Lista de eventos

## **5.6. Escopo do Sistema**

### **Definição de scopo do trabalho**

O sistema fará o controle de todos os produtos que entra e que sai da Empresa, ele fará o controle do estoque mostrando todos os produtos disponíveis para a venda. O escopo também visa mostrar a maneira na qual é descrito o projeto e a sua idéia geral. Irá mostrar como se interagir com o mundo externo a sua visão e transformações de funcionalidades.

### 5.7. Tabela de Caso de uso (Use Case)

<b>Descrição</b>	<b>Eventos</b>	<b>Use Case</b>	<b>Resposta</b>
Funcionário cadastra Pais	Dados do Pais	Cadastro de pais	Msgn01
Funcionário cadastra Estado	Dados do Estado	Cadastro de do Estado	Msgn02
Funcionário cadastra Cidade	Dados da Cidade	Cadastro de cidade	Msgn03
Funcionário cadastra Bairro	Dados do bairro	Cadastrar de bairro	Msgn04
Funcionário cadastra Rua	Dados da Rua	Cadastrar rua	Msgn05
Funcionário cadastra Cliente	Dados do Cliente	Cadastrar Cliente	Msgn06
Funcionário cadastra Fornecedor	Dados do Fornecedor	Cadastrar Fornecedor	Msgn07
Funcionário cadastra Funcionário	Dados do Funcionário	Cadastrar Funcionário	Msgn08
Funcionário cadastra Produto	Dados do produto	Cadastrar Produto	Msgn09
Funcionário cadastra Cargo	Dados de Cargo	Cadastrar Cargo	Msgn10
Cliente adiciona produtos ao Carrinho de Compras	Dados do Carrinho	Carrinho de Compras	Msgn11
Cliente faz a compra do Produto	Dados da venda de produtos	Venda de Produtos	Msgn12
Funcionário efetua compra de novo Produto	Dados do novo Produto	Compra de Produtos	Msgn13
Funcionário vê relatório de produto por compra	Dados de venda por compra efetuada	Relatório de compra por produto	Msgn14
Funcionário vê relatório de produto vendido	Dados de produtos vendidos	Produtos vendidos	Msgn15
Funcionário vê relatório de produto no estoque	Dados de produtos do estoque	Produtos no estoque	Msgn16
Funcionário vê relatório do Caixa da Empresa	Dados do Caixa	Relação do Caixa	Msgn17
Funcionário vê relatório de vendas por mês	Dados das vendas mensais	Relação de vendas mensais	Msgn18
Funcionário vê relatório de produtos comprados no mês	Dados de produtos comprados no mês	Relação de produtos comprados no mês	Msgn19
Funcionário vê relatório de Compras por fornecedor	Dados de compra por fornecedor	Relação de compra por fornecedor	Msgn20

Tabela 04 – Tabela de caso de uso (use case) continuação

## 6. ANÁLISE ORIENTADA A OBJETOS

### 6.1. Diagrama de Casos de Uso (Use-Case)

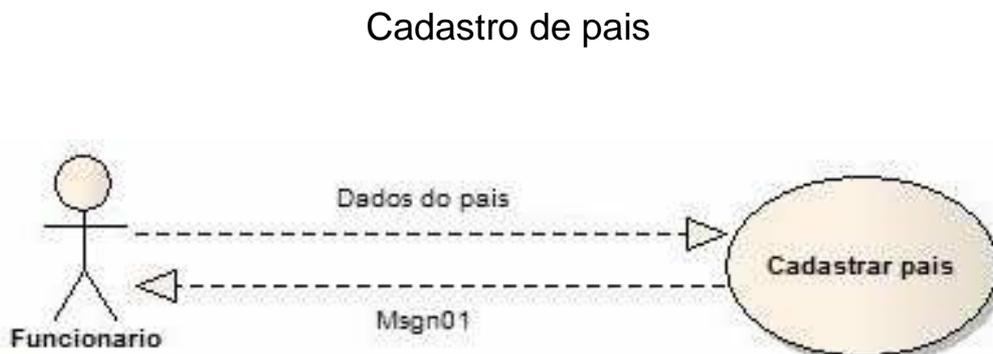


Figura 17 – diagrama Use case – cadastro de pais

<b>Nome da Use Case</b>	Cadastro de pais
<b>Ator (es)</b>	Funcionário
<b>Descrição</b>	O ator solicitará ao sistema informações de cadastro para um novo país.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1 – O ator solicita um formulário de cadastro para país.</li> <li>2 – O sistema gera o formulário de cadastro.</li> <li>3 – O ator preenche os campos corretamente todos os campos obrigatórios.</li> <li>4 – O ator envia as informações para o sistema.</li> <li>5 – O sistema verifica campos obrigatórios.</li> <li>6 – É concluído o cadastro.</li> <li>7 – O sistema gera um recibo do cadastro.</li> </ol>
<b>Cenário Alternativo</b>	O sistema encontra um campo errado e não gera o cadastro.

Tabela 05 – Cadastro de pais

## Cadastro de Estado

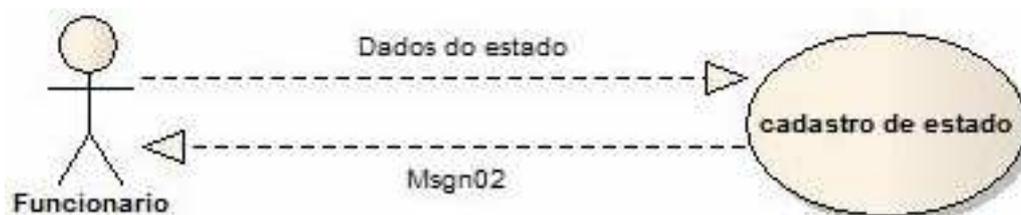


Figura 18 – diagrama Use case – cadastro de estado

<b>Nome da Use Case</b>	Cadastro de Estado
<b>Ator (es)</b>	Funcionário
<b>Descrição</b>	O ator solicitará ao sistema informações de cadastro para um novo Estado.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1 – O ator solicita um formulário de cadastro para Estado.</li> <li>2 – O sistema gera o formulário de cadastro.</li> <li>3 – O ator preenche os campos corretamente todos os campos obrigatórios.</li> <li>4 – O ator envia as informações para o sistema.</li> <li>5 – O sistema verifica campos obrigatórios.</li> <li>6 – É concluído o cadastro.</li> <li>7 – O sistema gera um recibo do cadastro.</li> </ol>
<b>Cenário Alternativo</b>	O sistema encontra um campo errado e não gera o cadastro.

Tabela 06 – cadastro de estado

## Cadastro de Cidade

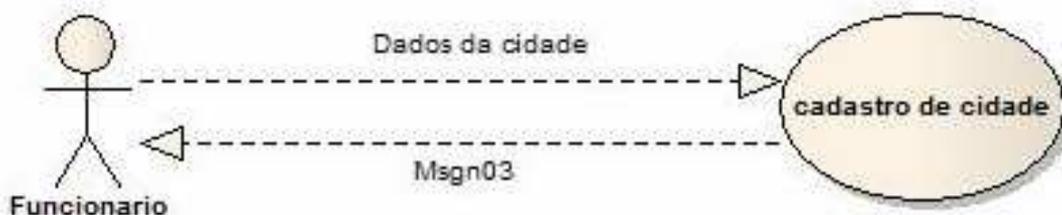


Figura 19 – diagrama Use case – cadastro de cidade

<b>Nome da Use Case</b>	Cadastro de Cidade
<b>Ator (es)</b>	Funcionário
<b>Descrição</b>	O ator solicitará ao sistema informações de cadastro para uma nova Cidade.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1 – O ator solicita um formulário de cadastro para uma nova Cidade.</li> <li>2 – O sistema gera um formulário de cadastro para um novo Funcionário.</li> <li>3 – O ator preenche os campos corretamente todos os campos obrigatórios.</li> <li>4 – O ator envia as informações para o sistema.</li> <li>5 – O sistema verifica campos obrigatórios.</li> <li>6 – É concluído o cadastro.</li> <li>7 – O sistema gera um recibo do cadastro.</li> </ol>
<b>Cenário Alternativo</b>	O sistema encontra um campo errado e não gera o cadastro.

Tabela 07 – Cadastro de cidade

## Cadastro de Bairro

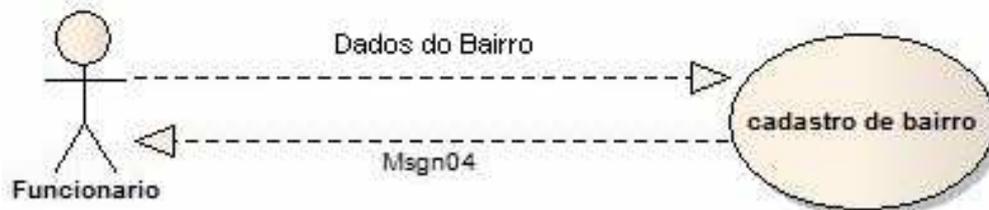


Figura 20 – diagrama Use case – cadastro de bairro

<b>Nome da Use Case</b>	Cadastro de bairro
<b>Ator (es)</b>	Funcionário
<b>Descrição</b>	O ator pede ao sistema que forneça informações de cadastro, para um novo bairro.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1 – O ator solicita um formulário de cadastro para o novo bairro a ser cadastrado.</li> <li>2 – O sistema gera o formulário de cadastro.</li> <li>3 – O ator preenche os campos corretamente todos os campos obrigatórios.</li> <li>4 – O ator envia as informações para o sistema.</li> <li>5 – O sistema verifica campos obrigatórios.</li> <li>6 – É concluído o cadastro.</li> <li>7 – O sistema gera um recibo do cadastro.</li> </ol>
<b>Cenário Alternativo</b>	O sistema encontra um campo errado e não gera o cadastro.

Tabela 08 – cadastro de bairro

## Cadastro de rua

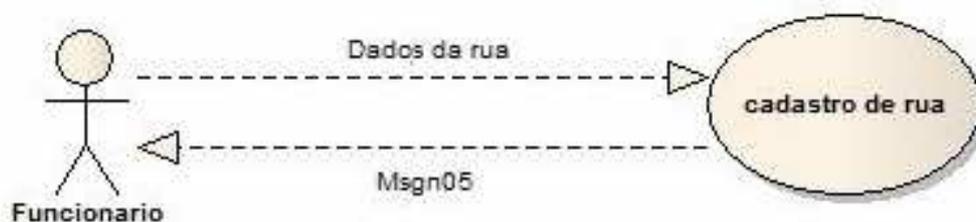


Figura 21 – diagrama Use case – cadastro de rua

<b>Nome da Use Case</b>	Cadastro de rua
<b>Ator (es)</b>	Funcionário
<b>Descrição</b>	O ator pede que o sistema forneça informações de cadastro para um novo cadastro de rua.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1 – O ator solicita um formulário de cadastro para a nova rua a ser cadastrado.</li> <li>2 – O sistema gera o formulário de cadastro.</li> <li>3 – O ator preenche os campos corretamente todos os campos obrigatórios.</li> <li>4 – O ator envia as informações para o sistema.</li> <li>5 – O sistema verifica campos obrigatórios.</li> <li>6 – É concluído o cadastro.</li> <li>7 – O sistema gera um recibo do cadastro.</li> </ol>
<b>Cenário Alternativo</b>	O sistema encontra um campo errado e não gera o cadastro.

Tabela 09 – cadastro de rua

## Cadastro de funcionário

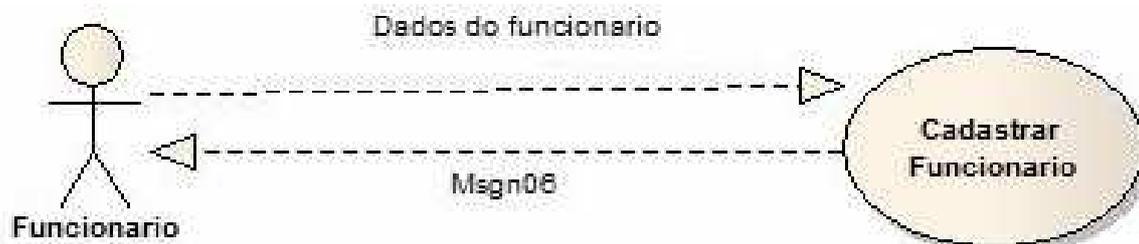


Figura 22 – diagrama Use case – cadastro de Funcionário

<b>Nome da Use Case</b>	Cadastro de funcionário
<b>Ator (es)</b>	Funcionário
<b>Descrição</b>	O ator pede que o sistema forneça informações de cadastro para um novo cadastro de funcionário.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1 – O ator solicita um formulário de cadastro para o novo funcionário a ser cadastrado.</li> <li>2 – O sistema gera o formulário de cadastro.</li> <li>3 – O ator preenche os campos corretamente todos os campos obrigatórios.</li> <li>4 – O ator envia as informações para o sistema.</li> <li>5 – O sistema verifica campos obrigatórios.</li> <li>6 – É concluído o cadastro.</li> <li>7 – O sistema gera um recibo do cadastro.</li> </ol>
<b>Cenário Alternativo</b>	O sistema encontra um campo errado e não gera o cadastro.

Tabela 10 – cadastro de funcionário

## Cadastro de cliente

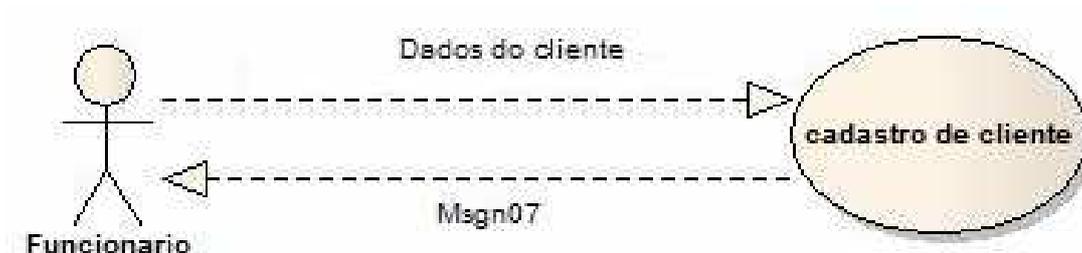


Figura 23 – diagrama Use case – cadastro de cliente

<b>Nome da Use Case</b>	Cadastro de cliente
<b>Ator (es)</b>	Funcionário
<b>Descrição</b>	O ator pede que o sistema forneça informações de cadastro para um novo cadastro de cliente.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1 – O ator solicita um formulário de cadastro para o novo cliente a ser cadastrado.</li> <li>2 – O sistema gera o formulário de cadastro.</li> <li>3 – O ator preenche os campos corretamente todos os campos obrigatórios.</li> <li>4 – O ator envia as informações para o sistema.</li> <li>5 – O sistema verifica campos obrigatórios.</li> <li>6 – É concluído o cadastro.</li> <li>7 – O sistema gera um recibo do cadastro.</li> </ol>
<b>Cenário Alternativo</b>	O sistema encontra campos errado e não gera o formulário de venda.

Tabela 11 – Cadastro de cliente

## Cadastro de produto

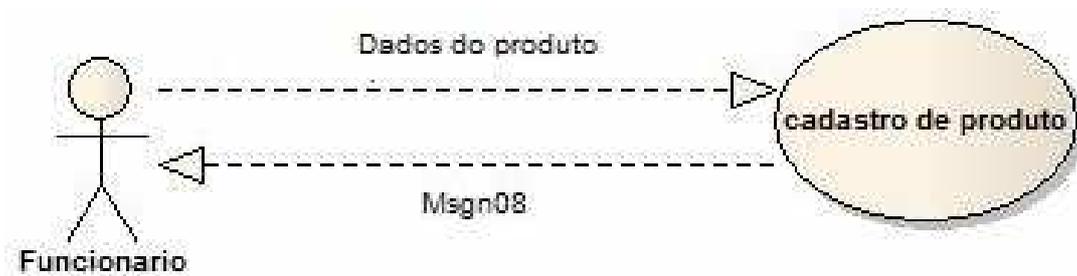


Figura 24– diagrama Use case – cadastro de produto

<b>Nome da Use Case</b>	Cadastro de pais
<b>Ator (es)</b>	Funcionário
<b>Descrição</b>	O ator pede que o sistema forneça informações de cadastro para um novo cadastro de produto.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1 – O ator solicita um formulário de cadastro para o novo produto a ser cadastrado no sistema.</li> <li>2 – O sistema gera o formulário de cadastro.</li> <li>3 – O ator preenche os campos corretamente todos os campos obrigatórios.</li> <li>4 – O ator envia as informações para o sistema.</li> <li>5 – O sistema verifica campos obrigatórios.</li> <li>6 – É concluído o cadastro.</li> <li>7 – O sistema gera um recibo do cadastro.</li> </ol>
<b>Cenário Alternativo</b>	O sistema encontra um campo errado e não gera o cadastro.

Tabela 12 – cadastro de produto

## Cadastro de fornecedor

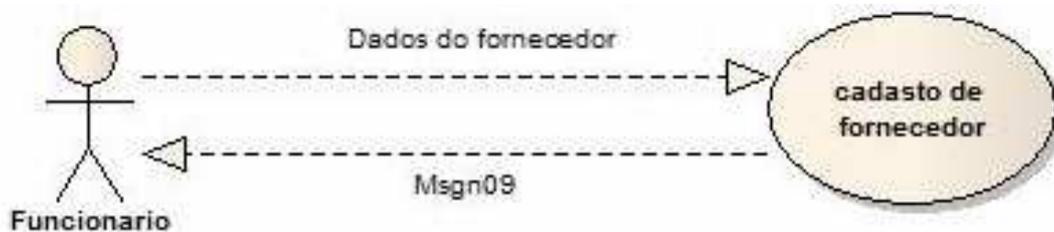


Figura 25 – diagrama Use case – cadastro de fornecedor

<b>Nome da Use Case</b>	Cadastro de fornecedor
<b>Ator (es)</b>	Funcionário
<b>Descrição</b>	O ator pede que o sistema forneça informações de cadastro para um novo cadastro de fornecedor.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1 – O ator solicita um formulário de cadastro para o novo fornecedor a ser cadastrado.</li> <li>2 – O sistema gera o formulário de cadastro.</li> <li>3 – O ator preenche os campos corretamente todos os campos obrigatórios.</li> <li>4 – O ator envia as informações para o sistema.</li> <li>5 – O sistema verifica campos obrigatórios.</li> <li>6 – É concluído o cadastro.</li> <li>7 – O sistema gera um recibo do cadastro.</li> </ol>
<b>Cenário Alternativo</b>	O sistema encontra um campo errado e não gera o cadastro.

Tabela 13 – Cadastro de fornecedor

## Cadastro de cargo

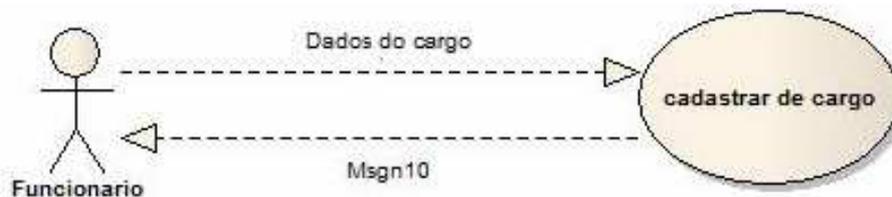


Figura 26 – diagrama Use case – cadastro de cargo

<b>Nome da Use Case</b>	Cadastro de cargo
<b>Ator (es)</b>	Funcionário
<b>Descrição</b>	O ator pede que o sistema forneça informações de cadastro para um novo cadastro de cargo.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1 – O ator solicita um formulário de cadastro para o novo cargo a ser cadastrado.</li> <li>2 – O sistema gera o formulário de cadastro.</li> <li>3 – O ator preenche os campos corretamente todos os campos obrigatórios.</li> <li>4 – O ator envia as informações para o sistema.</li> <li>5 – O sistema verifica campos obrigatórios.</li> <li>6 – É concluído o cadastro.</li> <li>7 – O sistema gera um recibo do cadastro.</li> </ol>
<b>Cenário Alternativo</b>	O sistema encontra um campo errado e não gera o cadastro.

Tabela 14 – cadastro de cargo

## Carrinho de compra

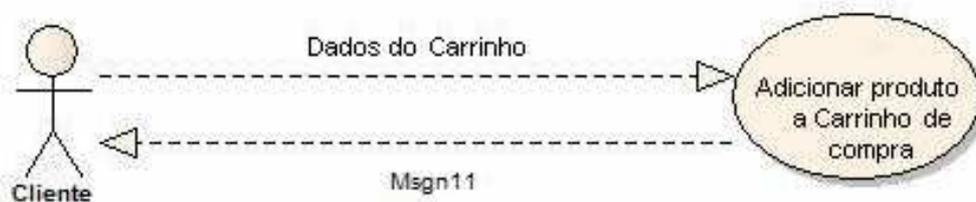


Figura 27 – diagrama Use case – Carrinho de Compra

<b>Nome da Use Case</b>	Carrinho de compra
<b>Ator (es)</b>	Cliente
<b>Descrição</b>	O ator cliente adiciona produtos ao carrinho.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1 – O ator solicita um formulário para fazer o pedido de compra.</li> <li>2 – O sistema gera um formulário para ser feito o adiconamento do produto no carrinho.</li> <li>3 – O ator preenche os campos, com todos os dados necessários.</li> <li>4 – O ator conclui sua operação</li> <li>5 – O sistema verifica a consistência dos dados informados.</li> </ol>
<b>Cenário Alternativo</b>	O sistema encontra inconsistência no formulário e informa o cliente.

Tabela 15 – Carrinho de Compra

## Venda de Produtos

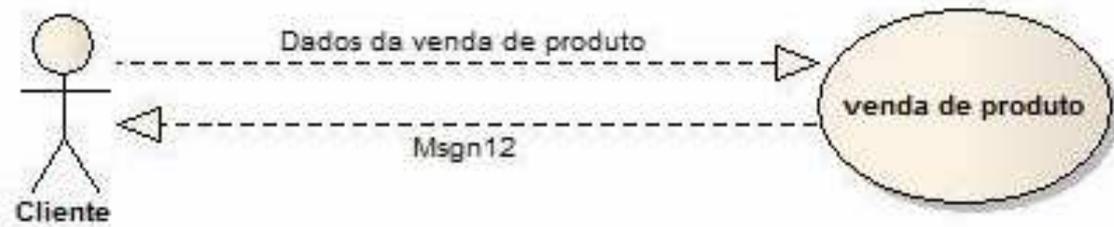


Figura 28– Diagrama de use case – venda de produto

<b>Nome da Use Case</b>	Venda de Produto
<b>Ator (es)</b>	Cliente
<b>Descrição</b>	O ator cliente compra produtos.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1 – O ator solicita um formulário para executar compra do produto.</li> <li>2 – O sistema gera um formulário para a compra do mesmo.</li> <li>3 – O ator preenche os campos, com todos os dados necessários.</li> <li>4 – O ator conclui sua operação</li> <li>5 – O sistema verifica a consistência dos dados informados.</li> </ol>
<b>Cenário Alternativo</b>	O sistema encontra erros e não conclui a venda.

Figura 16 – Venda de Produtos

## Funcionário efetua compra de produtos

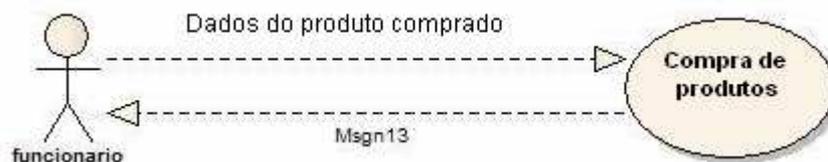


Figura 29 – diagrama Use case – Compra de Produtos

<b>Nome da Use Case</b>	Pedido de produto para fornecedor
<b>Ator (es)</b>	Funcionário
<b>Descrição</b>	Nesse caso o ator faz um pedido de produto.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1 – O ator solicita formulário de pedido de compra de produtos.</li> <li>2 – O sistema gera um formulário de pedido.</li> <li>3 – O ator preenche os campos.</li> <li>4 – O ator conclui a operação.</li> <li>5 – O sistema verifica se não há erros nos dados.</li> <li>6 – O sistema gera um recibo de pedido de produto.</li> </ol>
<b>Cenário Alternativo</b>	O sistema encontra erros não conclui a operação.

Tabela 17 – Compra de Produtos

## Relatório de compra por produto

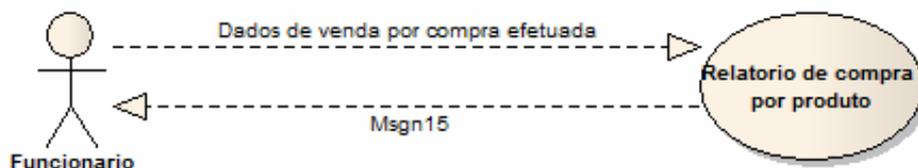


Figura 30 – diagrama Use case – relatório de compra por produto

<b>Nome da Use Case</b>	Relatório de compra por produto
<b>Ator (es)</b>	Funcionário
<b>Descrição</b>	O ator funcionário com cargo de gerente confere as informações de todas as compras agrupadas por produto.
<b>Cenário Principal</b>	1 – O ator solicita as informações para o sistema. 2 – O sistema gera todas as informações e mostra na tela a solicitação do ator.
<b>Cenário Alternativo</b>	O sistema não encontra informações do fornecedor desejado.

Tabela 18 – relatório de compra por produto

## Produtos vendidos no dia

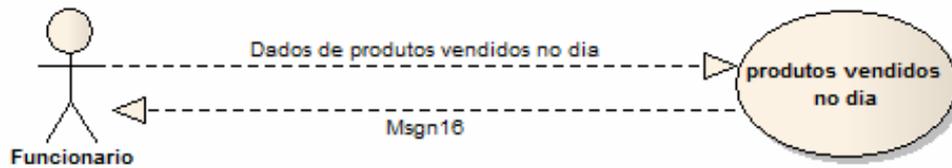


Figura 31 – diagrama Use case – produtos vendidos no dia

<b>Nome da Use Case</b>	Produtos vendidos no dia
<b>Ator (es)</b>	Funcionário
<b>Descrição</b>	O ator confere as informações de todas as vendas realizadas no dia atual.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1 – O ator solicita as informações para o sistema.</li> <li>2 – O sistema confere se há vendas no dia atual.</li> <li>3 - O sistema gera todas as informações e mostra na tela a solicitação do ator.</li> </ol>
<b>Cenário Alternativo</b>	O sistema não encontra informações no atual dia.

Tabela 19 – produtos vendidos no dia

## Produtos do estoque

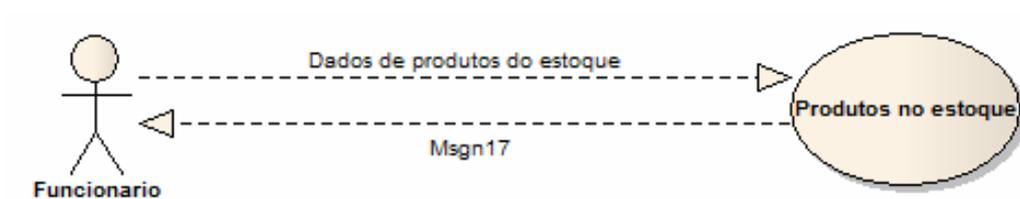


Figura 32 – diagrama Use case – produtos do estoque

<b>Nome da Use Case</b>	Produtos do estoque
<b>Ator (es)</b>	Funcionário
<b>Descrição</b>	Agora o ator checará todos os produtos que existe no estoque.
<b>Cenário Principal</b>	1 – O ator solicita as informações para o sistema. 2 – O sistema gera todas as informações e mostra na tela a solicitação do ator.
<b>Cenário Alternativo</b>	O sistema não encontra informações no estoque.

Tabela 20 – produtos do estoque

## Caixa da Empresa

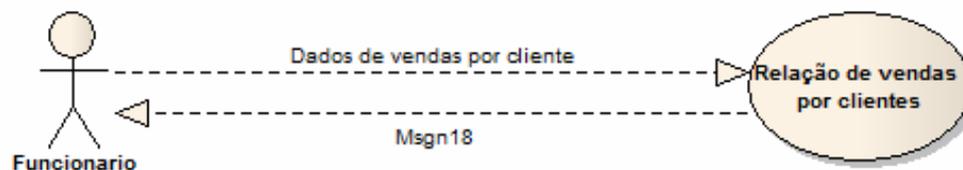


Figura 33 – diagrama Use case – relação de vendas por clientes

<b>Nome da Use Case</b>	Caixa da Empresa
<b>Ator (es)</b>	Funcionário
<b>Descrição</b>	O ator confere as informações financeiras da empresa
<b>Cenário Principal</b>	1 – O ator solicita as informações para o sistema. 2 – O sistema gera as informações solicitadas pelo ator e as mostra na tela a solicitação do ator.
<b>Cenário Alternativo</b>	O sistema não encontra informações.

Tabela 21 – relação de vendas por clientes

## Relação de vendas mensais por cliente

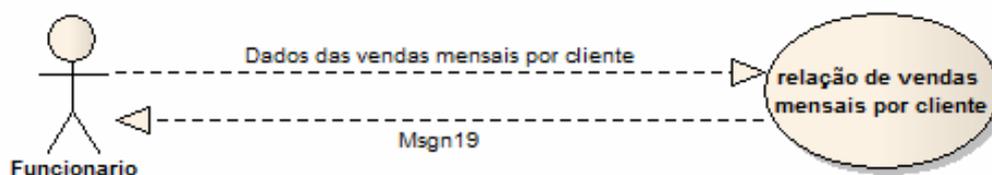


Figura 34 – diagrama Use case – relação de vendas mensais por cliente

<b>Nome da Use Case</b>	Relação de vendas mensais por cliente
<b>Ator (es)</b>	Funcionário
<b>Descrição</b>	O ator confere todas as vendas para clientes agrupadas por mês.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1 – O ator solicita as informações para o sistema.</li> <li>2 – O sistema confere as informações, para ver se existe vendas no mês desejado.</li> <li>3 – O sistema gera todas as informações as informações desejadas.</li> </ol>
<b>Cenário Alternativo</b>	O sistema não encontra as informações do mês desejado.

Tabela 22 – relação de vendas mensais por cliente

## Relação de produtos comprados por mês

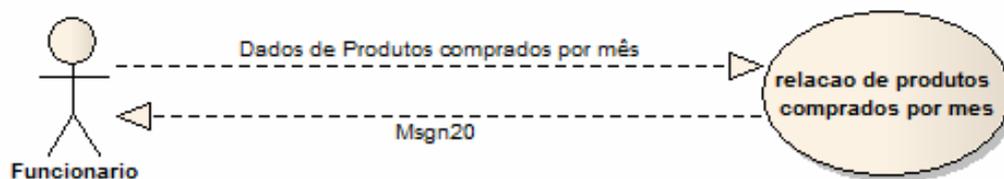


Figura 35 – diagrama Use case – relação de produtos comprados por mês

<b>Nome da Use Case</b>	Relação de produtos comprados por mês
<b>Ator (es)</b>	Funcionário
<b>Descrição</b>	Esse caso, o sistema gerará um relatório de todos os produtos comprados no mês desejado.
<b>Cenário Principal</b>	<ol style="list-style-type: none"> <li>1 – O ator solicita um relatório do sistema.</li> <li>2 – O sistema verifica se encontra as informações desejadas.</li> <li>3 – O sistema gera o relatório de todas as compras efetuadas no mês desejado.</li> </ol>
<b>Cenário Alternativo</b>	O sistema não encontra nenhuma informação no mês procurado.

Tabela 23 – relação de produtos comprados por mês

## Relação de produtos comprados por fornecedor

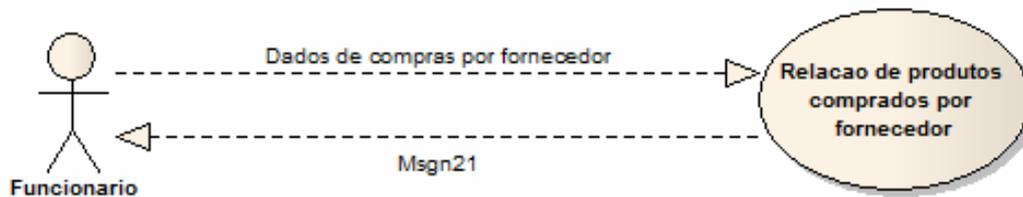


Figura 36 – diagrama Use case – relação de produtos comprados por fornecedor

<b>Nome da Use Case</b>	Relação de produtos comprados por fornecedor
<b>Ator (es)</b>	Funcionário
<b>Descrição</b>	Neste caso o funcionário com cargo de gerente confere as informações de todos os produtos agrupados por fornecedor
<b>Cenário Principal</b>	1 – O ator solicita as informações para o sistema. 2 – O sistema gera todas as informações e mostra na tela a solicitação do ator.
<b>Cenário Alternativo</b>	O sistema não encontra informações do fornecedor desejado.

Tabela 24 – relação de produtos comprados por fornecedor

## 6.2. Diagrama de Classe

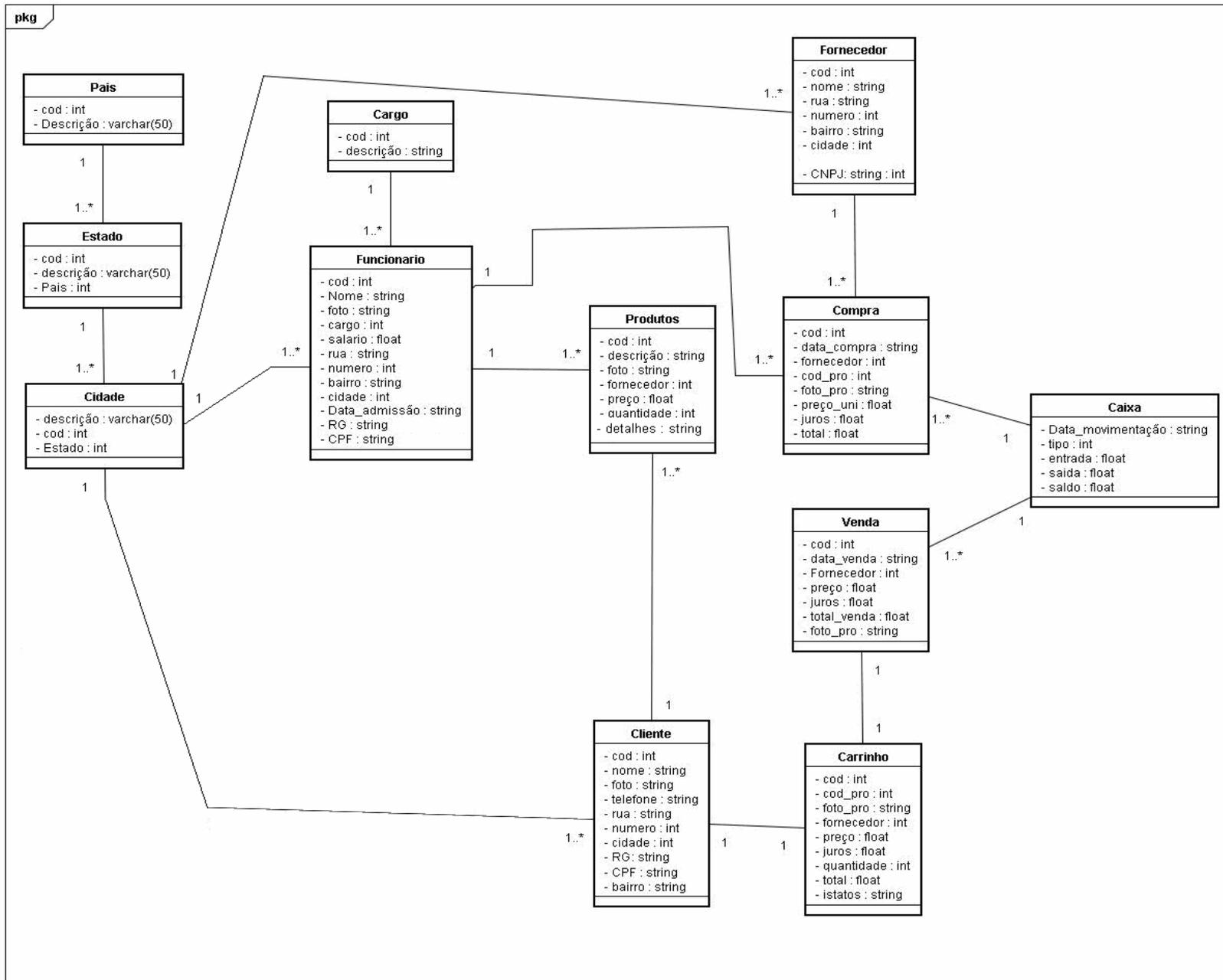


Figura 37 – diagrama classe

### 6.3. Diagrama de Seqüência

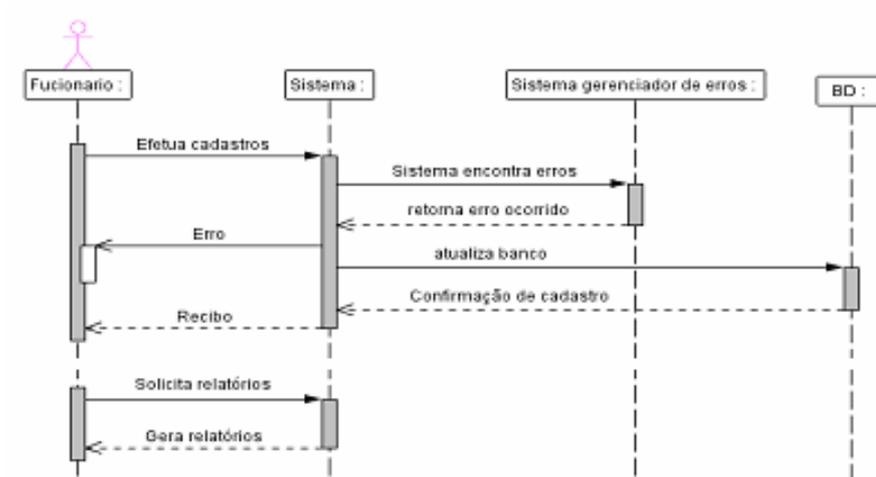


Figura 38 – diagrama de Seqüência – Funcionário

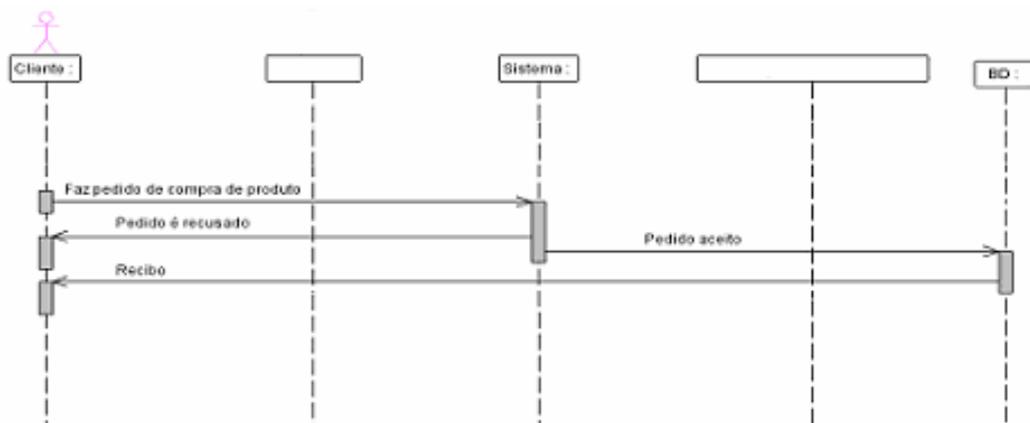


Figura 39 – diagrama de Seqüência - Cliente

## 7. INTERFACES DO SISTEMA

Esta é a Home page é a tela inicial do sistema, o lugar onde o cliente e funcionário têm acesso ao sistema.

**CAS info**  
computadores & acessórios

INICIO      GERENCIA

 Quantidade: 21 Preço: 1200.01 Fornecedor: 1 Descrição: Computador Cod: 1	 Quantidade: 32 Preço: 3.01 Fornecedor: 2 Descrição: Fone de Ouvido Cod: 2	 Quantidade: 4 Preço: 10.04 Fornecedor: 3 Descrição: mouse Cod: 3
<b>Detalhes</b> 	<b>Detalhes</b> 	<b>Detalhes</b> 
 Quantidade: 21 Preço: 50 Fornecedor: 1 Descrição: Web Can Cod: 4	 Quantidade: 12 Preço: 700 Fornecedor: 1 Descrição: Monitor 17 Cod: 5	 Quantidade: 23 Preço: 23 Fornecedor: 1 Descrição: teclado Cod: 6
<b>Detalhes</b> 	<b>Detalhes</b> 	<b>Detalhes</b> 

CAS info - todos os direitos reservados

Figura 40 – Tela Inicial do Web Site

**Nessa tela se localiza o cadastro de Produtos  
Na área Administrativa do sistema.**

início Relatórios Bairros Cargos Cidades Clientes Estados Fornecedor Funcionarios Pais Produtos Ruas

## Cadastro de Produtos

Descrição:

Foto:

Fornecedor:

Preço:

Quantidade:

Detalhes:

1 tera de hd, 4gb de RAM,  
processador 4 core



Area Administrativa

**Figura 41 – Tela para Cadastro de Produtos**

Nessa tela se localiza o um Relatório de fluxo de caixa  
Também na área Administrativa do Sistema.

Logout

início	Cadastros	Produtos no Estoque	Produto por compra	Prod. vendido	Caixa	Vendas por Mês	Prod. Comprados no Mês	C. por Fornecedor
--------	-----------	---------------------	--------------------	---------------	-------	----------------	------------------------	-------------------

## Caixa

Código	Tipo	Entradas	Saidas	Saldo
136	entrada	1760		9882,4
135	entrada	1650		8122,4
134	entrada	30800		6472,4
133	saida	0	60000	-24327,6
132	entrada	440		35672,4
131	entrada	110		35232,4
130	entrada	2308,9		35122,4
129	entrada	5,5		32813,5
128	entrada	44		32808
127	entrada	1868,9		32764
126	entrada	3300		30895,1
125	entrada	2200		27595,1
124	entrada	6600		25395,1
123	entrada	660		18795,1
122	entrada	1650		18135,1
121	entrada	1650		16485,1
118	entrada	550		14835,1
117	entrada	7700		14285,1

**Figura 42 – Tela de Relatório do fluxo de caixa**

**Nessa tela se localiza o Cadastro de Clientes  
Também na área Administrativa do Sistema.**

início Relatórios Bairros Cargos Cidades Clientes Estados Fornecedor Funcionários Pais Produtos Ruas

Código:

Nome:

Foto:

Telefone:

Rua:

Número:

Cidade:

Estado:

Pais:

RG:

CPF:

Dados Salvos Com Sucesso

**Figura 43 – Tela para Cadastro de Clientes**

Nessa tela se localiza o Relatório de Produtos do Estoque Também na área Administrativa do Sistema.

início Cadastros

## Produtos do Estoque

	Descrição	Detalhes	Fornecedor	Preço	Quantidade
	Web Can		das	40	15
	Teclado	teclado com saída usb, padrão do brasil	das	30	21
	Notebook	1 tera de hd, 4gb de RAM, processador 4 core	das	3000	2
	Notebook	340gb de hd, 4gb de RAM, processador 2 core	das	2000	3
	Notebook	340gb de hd, 4gb de RAM, processador 2 core	das	111	1
	Notebook	340gb de hd, 4gb de RAM, processador 2 core	das	2000	2
	Mouse		das	1	1
	Impressora	Impressora hp, saída para rede, a tonner, colorida	das	500	2
	Fone de Ouvido		das	12	11
	Estabilizador		das	1	1

12

Area Administrativa

Figura 44 – Tela para Relatório de todos os produtos do Estoque

## 8. CONCLUSÃO

Após uma longa análise das necessidades da empresa, a solução mais viável para a mesma, será a informatização dos processos referentes às vendas da empresa, pois é onde foi encontrada a maior deficiência da empresa encontrada nesse departamento, por tanto concluo que deve ser feita a informatização do setor de vendas da empresa CAS info, a fim de agilizar processos, aumentando assim a lucratividade da empresa.

Após a implantação do sistema na CAS info, concluo que esse poderá satisfazer toda a necessidade do(s) usuário(s), e que os problemas ocorridos anteriormente era por não ter dados importantes na de decisão, tipo:

- Entradas e Saídas de Produtos;
- Clientes Cadastrados;
- Fornecedores Cadastrados;
- Controle de Funcionários;
- Relatórios diversos;

As informações sobre cadastros, entradas e saídas de produtos vendidos corretamente possibilitam um melhor controle do estoque para que a empresa possa ter maior controle sobre seus produtos, um atendimento dinâmico para a satisfação dos clientes e Fornecedores, visto que todo o processo da empresa era feito de forma manual.

Portanto para o proprietário da empresa o sistema demonstrará a importância da informatização, além dos benefícios que o trará para empresa em um determinado tempo.

Futuramente será implantado o sistema de cartão de crédito no sistema, inicialmente será implantado um sistema de vendas apenas com boletos e futuramente será desenvolvida a venda de produtos com cartão de credito. O Banco de Dados será todo mudado para Oracle.

## 9. BIBLIOGRAFIA

Guiadohardware.net,. Orientação a objetos

<http://www.guiadohardware.net/artigos/programacao-orientada-objetos/>

Wikipédia. UML

<http://pt.wikipedia.org/wiki/UML>

Wikipédia. Oracle

<http://pt.wikipedia.org/wiki/Oracle>

UML, Fundamentos

[http://docs.kde.org/kde3/pt\\_BR/kdesdk/umbrello/umlbasics.html](http://docs.kde.org/kde3/pt_BR/kdesdk/umbrello/umlbasics.html)

Silva, B. G. F.,. **Análise Orientada a Objeto**

<http://www.rootlinux.com.br/documentos/downloads/analise.pdf>

UML, Fundamentos

[http://docs.kde.org/kde3/pt\\_BR/kdesdk/umbrello/umlbasics.html](http://docs.kde.org/kde3/pt_BR/kdesdk/umbrello/umlbasics.html)

URI, Universidade Regional Integrada do Alto Uruguai e das Missões, **Análise Orientada a Objeto**,

<http://www.urisan.tche.br/~pbetencourt/engsoftll/orientacaoobjeto.doc>.

Lee, R. C. & Tepfenhart, W. M. **UML e C++ : Guia Prático de Desenvolvimento Orientado a Objeto**, traduzido por Celso Roberto Paschoa, Makron Books LTD., 2002

Elementos UML

[http://docs.kde.org/stable/pt\\_BR/kdesdk/umbrello/uml-elements.html](http://docs.kde.org/stable/pt_BR/kdesdk/umbrello/uml-elements.html)

Diagrama de Componente

<http://fafica.codethe.net/comp.ppt#10>