



**UNIVERSIDADE DO SUL DE SANTA CATARINA**

**FABRÍCIO RIEG VIANA**

**MÁRCIO SOARES**

**DESENVOLVIMENTO CENTRADO NO USUÁRIO**

**A PARTIR DA METODOLOGIA ICONIX**

**Palhoça**  
**2008**

**FABRÍCIO RIEG VIANA**

(fabricorieg@gmail.com)

**MÁRCIO SOARES**

(mscinho@gmail.com)

**DESENVOLVIMENTO CENTRADO NO USUÁRIO  
A PARTIR DA METODOLOGIA ICONIX**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade do Sul de Santa Catarina, como requisito parcial à obtenção do título de Bacharel em Ciência da Computação.

Orientadora Prof<sup>a</sup>. Msc. Vera Rejane Niedersberg Schuhmacher

(vera.schumacher@gmail.com)

**Palhoça**

**2008**

**FABRÍCIO RIEG VIANA**  
**MÁRCIO SOARES**

**DESENVOLVIMENTO CENTRADO NO USUÁRIO**  
**A PARTIR DA METODOLOGIA ICONIX**

Este Trabalho de Conclusão de Curso foi julgado adequado à obtenção do título de Bacharel em Ciência da Computação e aprovado em sua forma final pelo Curso de Ciência da Computação, da Universidade do Sul de Santa Catarina.

\_\_\_\_\_, \_\_\_\_ de \_\_\_\_\_ de 2008  
Local, dia mês ano

---

Prof<sup>a</sup>. Msc. Vera Rejane Niedersberg Schuhmacher  
Universidade do Sul de Santa Catarina - UNISUL

---

Prof. Dr. Eng<sup>o</sup>. Aran Bey Tcholakian Morales  
Universidade do Sul de Santa Catarina – UNISUL

---

Prof. Msc. Ivo Zimmermann  
Universidade do Sul de Santa Catarina - UNISUL

---

Rosemberg Tartari  
Universidade do estado de Santa Catarina - UDESC

Dedico este trabalho às pessoas que foram e  
que são os maiores responsáveis pela minha  
formação acadêmica e apoiadores desta longa  
caminhada: meus pais, Arcemiro e Natalenia.

Márcio Soares.

Dedico este trabalho às pessoas que foram e  
que são os maiores responsáveis pela minha  
formação e foram de grande apoio nesta longa  
caminhada: meus pais, Mário e Armelinda.

Fabício Rieg Viana.

## **AGRADECIMENTOS**

Muitas pessoas ajudaram para a conclusão deste projeto, bem como as pessoas que tornaram possível a realização do mesmo as quais citamos a seguir:

Aos nossos pais, principais responsáveis da obra, que sempre nos deram apoio, motivação, coragem e orientação, além do investimento que lhes coube.

A nossa orientadora e amiga, Prof<sup>a</sup>. Msc. Vera Rejane Niedersberg Schuhmacher, em vários momentos nós pensávamos em desistir do projeto, e ela estava lá para nos dar apoio e ânimo para continuar nessa caminhada.

Aos professores do Curso de Ciência da Computação da Universidade do Sul de Santa Catarina, que estavam juntos na nossa caminhada, em especial, à Dr<sup>a</sup>. Maria Inés Castiñeira, que nos apoiou em vários momentos durante o curso.

Ao Rodrigo Dias de Azevedo que nos ajudou no momento em que estávamos com maior dificuldade no desenvolvimento, e foi com esse grande apoio que o projeto pode ser concluído.

Ao Rosemberg Tartari, que nos ajudou nos processos de desenvolvimento nos mostrando o caminho que devíamos seguir durante o decorrer do projeto.

A todas as pessoas que contribuíram direta ou indiretamente para a conclusão deste trabalho, sejam elas professores, colegas ou amigos.

Fabício Rieg Viana

Márcio Soares

“Tudo o que um sonho precisa para  
ser realizado é alguém que acredite  
que ele possa ser realizado.”

(Roberto Shinyashiki)

## **RESUMO**

Neste projeto, iniciou-se o estudo da possível integração de dois modelos: o modelo de desenvolvimento ICONIX e o DCU – Design centrado no usuário. A metodologia de desenvolvimento de software ICONIX possui um alto grau de aceitação em empresas de software por sua racionalização em questões relacionadas à documentação. Observou-se, no entanto, o fraco contato com o usuário final durante o desenvolvimento do projeto. O ICONIX analisa os requisitos junto ao cliente, sendo que uma nova integração só ocorre na entrega do produto software, sugerindo um volume grande de alterações, retrabalho na modelagem e desenvolvimento. O método design centrado no usuário interage com o usuário final, desde o início do desenvolvimento, na análise de requisitos até a entrega do projeto, possuindo, assim, uma integração com o usuário durante o processo de desenvolvimento. O objetivo do projeto é a integração de ambos os métodos. O estudo de caso usado foi o controle de um plano de saúde animal, desenvolvido basicamente com a tecnologia Flex e Java. Na monografia são apresentados os passos da metodologia com sua integração e os resultados de sua aplicação.

Palavras chave: ICONIX, Design centrado no usuário, usabilidade.

## **ABSTRACT**

During this project, the study of the possible integration of two models started: the development of the models ICONIX and DCU (Centered Design on the user). The methodology of development of the ICONIX Software has a very high level of acceptance among software companies because of its rationalization of matters related to documentation. However, the weak contact with the final user during the development of the project has been observed. The ICONIX analyzes the requirements with the customer and provides a new integration that will only occur in the delivery of the product software, suggesting this way a large amount of changes, remake of the model and development. The DCU interacts with the final user since the beginning of the development, during the requirements analyzes until the delivery of the project, providing this way an interaction with the user throughout the development process. The goal of the project is the integration of both methods. The study case used was the control of a veterinarian medical plan that was designed basically with the Flex and Java technology. During this dissertation, the steps of the methodology and its integration and the results outcomes of its application will be shown / displayed / explained.

Key words: ICONIX, Centered Design on the User, usage.



## LISTA DE ILUSTRAÇÕES

Figura 1 - Solução para o Estudo de Caso .....	20
Figura 2 - Estudo de Integração entre a Metodologia e o Método .....	20
Figura 3 – Engenharia de Software em Camadas.....	24
Figura 4 – Modelo Cascata.....	26
Figura 5 – Modelo Evolucionário.....	27
Figura 6 – Modelo Espiral.....	28
Figura 7 – Modelo Incremental. ....	29
Figura 8 – Processo Iconix. ....	34
Figura 9 - Análise de Requisitos.....	36
Figura 10 - Atividades da análise e do projeto preliminar. ....	37
Figura 11 - Atividades do Projeto.....	38
Figura 12 – Modelo de Integração ao Design Centrado no Usuário. ....	42
Figura 13 – Experiência com Computador X Domínio na Facilidade de Aprendizagem X Facilidade de Uso.....	46
Figura 14 – Relacionamentos entre Atributos.....	48
Figura 15 – Estrutura integração metodologia ICONIX e método DCU.....	53
Figura 16 – Atores do sistema. ....	56
Figura 17 – Requisitos funcionais. ....	58
Figura 18 – Requisitos não funcionais.....	60
Figura 19 – Aplicação do Jogo de cartas (1). ....	61
Figura 20 – Aplicação do Jogo de cartas (2). ....	61
Figura 21 – Aplicação do Jogo de cartas (3). ....	62
Figura 22 – Definição dos novos requisitos funcionais, após a realização do jogo de cartas..	66
Figura 23 – Protótipo de tela de Identificação do Modelo 01. ....	72
Figura 24 – Protótipo de tela de Identificação do Modelo 02. ....	73
Figura 25 – Protótipo de tela de Identificação do Modelo 03 .....	74
Figura 26 - Protótipo de tela de Identificação do Modelo 04.....	75
Figura 27 – Protótipo de tela Modelo 01. ....	76
Figura 28 – Protótipo de tela Modelo 02. ....	77
Figura 29 – Protótipo de tela Modelo 03. ....	78
Figura 30 – Protótipo de tela Aprovado Pelo Cliente.....	79

Figura 31 – Novo Protótipo de tela Aprovado Pelo Cliente. ....	80
Figura 32 – Definição dos pacotes de Casos de Uso. ....	81
Figura 33 – Diagrama de Casos de Uso – Pacote plano de saúde. ....	82
Figura 34 – Diagrama de Casos de Uso – Pacote prestadora de serviço. ....	83
Figura 35 – Diagrama de Casos de Uso – Pacote proprietário.....	85
Figura 36 – Diagrama de atividade – plano de saúde. ....	98
Figura 37 – Diagrama de robustez – Cadastro tipo serviço. ....	99
Figura 38 – Diagrama de seqüência – Cadastro tipo serviço. ....	99
Figura 39 – Diagrama de robustez – inclusão/alteração de plano de saúde.....	100
Figura 40 – Diagrama de seqüência – inclusão/alteração de plano de saúde.....	100
Figura 40 – Diagrama de robustez – inclusão/exclusão tipo serviço no plano de saúde. ....	101
Figura 42 – Diagrama de seqüência – inclusão/exclusão tipo serviço no plano de saúde. ....	101
Figura 43 – Diagrama de robustez – inclusão/alteração prestador de serviço. ....	102
Figura 44 – Diagrama de seqüência – inclusão/alteração prestador de serviço. ....	102
Figura 45 – Diagrama de robustez – Tipos de serviços ao prestador de serviço.....	103
Figura 46 – Diagrama de seqüência – Tipos de serviços ao prestador de serviço. ....	103
Figura 47 – Diagrama de robustez – cadastro e alteração do proprietário.....	104
Figura 48 – Diagrama de seqüência – cadastro e alteração do proprietário.....	104
Figura 49 – Diagrama de robustez – cadastro e alteração do paciente. ....	105
Figura 50 – Diagrama de seqüência – cadastro e alteração do paciente. ....	105
Figura 51 – Diagrama de classe – plano saúde.....	106
Figura 52 – Ambiente de Desenvolvimento do Sistema.....	108
Figura 53 – Tela de Identificação do Administrador.....	114
Figura 54 – Tela de Gerenciamento de Tipo de Serviço.....	115
Figura 55 – Tela de cadastro de Serviços.....	116
Figura 56 – Tela de Gerenciamento dos Planos de Saúde. ....	117
Figura 57 – Tela Gerenciamento de Planos com seus respectivos serviços. ....	118
Figura 58 – Tela para buscar os serviços e os cadastrar no plano de saúde.....	119
Figura 59 – Tela de Gerenciamento de Proprietário.....	120
Figura 60 – Tela de Cadastro do Proprietário. ....	121
Figura 61 – Tela de Gerenciamento Paciente.....	122
Figura 62 – Tela do Cadastro de Paciente.....	123
Figura 63 – Tela de Gerenciamento do Prestador de Serviços.....	124
Figura 64 – Tela de Cadastro do Prestador de Serviço.....	125

Figura 65 – Tela de Controle de Tipo de Serviço.....	126
Figura 66 – Tela de Inclusão de Serviço ao Prestador de Serviço. ....	127

## LISTA DE QUADROS

Quadro 1 – Alguns fatores que afetam o design de interface do usuário.....	47
Quadro 2 – Questionamentos levantados durante entrevista com o cliente.....	55
Quadro 3 – Descrição dos atores do sistema.....	57
Quadro 4 – Descrição dos requisitos funcionais. ....	59
Quadro 5 – Resultado final do jogo de cartas. ....	65
Quadro 6 – Descrição dos requisitos funcionais do pacote plano de saúde.....	69
Quadro 7 – Descrição dos requisitos funcionais do pacote prestadora de serviço. ....	70
Quadro 8 – Descrição dos requisitos funcionais do pacote proprietário.....	71
Quadro 9 – Descrição dos casos de uso do pacote plano de saúde.....	83
Quadro 11 – Descrição dos casos de uso do pacote proprietário. ....	85
Quadro 13 – Documentação do Caso de Uso - Gerência de relatórios ao financeiro.....	87
Quadro 14 – Documentação do Caso de Uso - Gerência Proprietário/Animal.....	87
Quadro 15 – Documentação do Caso de Uso - Gerencia Prestador de Serviço.....	88
Quadro 16 – Documentação do Caso de Uso - Gerência de tipo de serviço prestado.....	89
Quadro 17 – Documentação do Caso de Uso - Gerência de tipos de plano de saúde. ....	90
Quadro 18 – Documentação do Caso de Uso - Gerência de contas de usuário. ....	91
Quadro 19 – Documentação do Caso de Uso - Gerência de login de usuário. ....	91
Quadro 21 – Documentação do Caso de Uso - Gerência de prestação de serviço.....	93
Quadro 22 – Documentação do Caso de Uso - Gerência de inclusão de prestador de serviço.....	94
Quadro 23 – Documentação do Caso de Uso - Gerência de contas de usuário da Prestserv...	94
Quadro 24 – Documentação do Caso de Uso - Gerência de relatórios financeiros.....	95
Quadro 26 – Documentação do Caso de Uso - Cadastro de proprietário. ....	97
Quadro 27 – Documentação do Caso de Uso - Gerencia de Animal.....	98

## SUMÁRIO

<b>1.</b>	<b>INTRODUÇÃO .....</b>	<b>16</b>
1.1	ESTUDO DE CASO.....	17
1.2	PROBLEMA.....	17
1.3	OBJETIVOS .....	18
<b>1.3.1</b>	<b>Objetivo Geral.....</b>	<b>18</b>
<b>1.3.2</b>	<b>Objetivo específico.....</b>	<b>18</b>
1.4	JUSTIFICATIVA.....	18
1.5	PROPOSTA DE SOLUÇÃO .....	19
<b>1.5.1</b>	<b>Solução para o Estudo de Caso.....</b>	<b>19</b>
<b>1.5.2</b>	<b>Solução para a Integração das Metodologias .....</b>	<b>20</b>
1.6	DELIMITAÇÃO.....	21
1.7	ESTRUTURA DA MONOGRAFIA.....	21
<b>2.</b>	<b>ENGENHARIA DE SOFTWARE.....</b>	<b>23</b>
2.1	O DESENVOLVIMENTO EM CAMADAS.....	23
2.2	ARCABOUÇO DE PROCESSO DE SOFTWARE.....	24
<b>2.2.1</b>	<b>Modelos de processos de software .....</b>	<b>25</b>
2.2.1.1	Modelo cascata.....	25
2.2.1.2	Modelo Evolucionário.....	26
2.2.1.3	Modelo Espiral.....	27
2.2.1.4	Modelo Incremental .....	28
<b>2.2.2</b>	<b>Processo de Desenvolvimento Ágil.....</b>	<b>29</b>
2.2.2.1	Extreme Programming (XP).....	31
2.2.2.2	Scrum.....	32
2.3	ICONIX .....	33
<b>2.3.1</b>	<b>Questões fundamentais da metodologia Iconix .....</b>	<b>34</b>
<b>2.3.2</b>	<b>Características fundamentais da metodologia Iconix .....</b>	<b>35</b>
<b>2.3.3</b>	<b>Desenvolvimento Aplicado à Metodologia Iconix .....</b>	<b>35</b>
2.3.3.1	Análise de Requisitos .....	35
2.3.3.2	Análise e Projeto Preliminar .....	36
2.3.3.3	Projeto .....	37
2.3.3.4	Implementação.....	38
<b>2.3.4</b>	<b>Modelos de Diagramas usados na metodologia Iconix.....</b>	<b>38</b>
2.3.4.1	Modelo de Domínio .....	39

2.3.4.2	Modelo de Caso de Uso.....	39
2.3.4.3	Análise de Robustez.....	40
2.3.4.4	Diagrama de Classe.....	41
2.3.4.5	Modelagem de Interação .....	41
2.4	DESIGN CENTRADO NO USUÁRIO.....	41
<b>2.4.1</b>	<b>Técnicas de Design Centrado no Usuário.....</b>	<b>43</b>
2.4.1.1	Cenários .....	43
2.4.1.2	Análise de Tarefas Utilizando Cenário .....	44
2.4.1.3	Prototipação .....	44
2.4.1.4	Entrevista .....	45
2.4.1.5	Questionário.....	45
<b>2.4.2</b>	<b>Design da interface com usuário.....</b>	<b>46</b>
2.4.2.1	Contexto de Uso.....	47
2.4.2.2	Foco no Usuário .....	48
2.4.2.3	Integração com o Design .....	49
2.4.2.4	Teste com Usuário.....	49
<b>2.4.3</b>	<b>Qualidades Ergonômicas para IHC.....</b>	<b>49</b>
<b>3</b>	<b>MODELAGEM .....</b>	<b>53</b>
3.1	APLICAÇÃO DA METODOLOGIA .....	53
<b>3.1.1</b>	<b>Identificação do Objetivo do Sistema .....</b>	<b>54</b>
<b>3.1.2</b>	<b>Realização de Entrevistas com o Usuário Alvo (DCU) .....</b>	<b>54</b>
<b>3.1.3</b>	<b>Entrevista de Identificação de Características do Público Alvo.....</b>	<b>55</b>
<b>3.1.4</b>	<b>Identificação de Atores (ICONIX).....</b>	<b>56</b>
<b>3.1.5</b>	<b>Identificação de Requisitos Funcionais (ICONIX).....</b>	<b>57</b>
<b>3.1.6</b>	<b>Identificação de Requisitos Não Funcionais (ICONIX) .....</b>	<b>60</b>
<b>3.1.7</b>	<b>Jogo de Cartas com o Usuário – Requisitos Funcionais (DCU).....</b>	<b>60</b>
3.1.7.1	Pacote plano de saúde.....	66
3.1.7.2	Pacote Prestador de Serviço.....	69
3.1.7.3	Pacote Proprietário .....	71
<b>3.1.8</b>	<b>Prototipação das Interfaces (ICONIX+DCU) .....</b>	<b>71</b>
3.1.8.1	Estrutura da Tela .....	72
3.1.8.2	Definição Contextual da Tela .....	75
<b>3.1.9</b>	<b>Validação das Interfaces com Usuário Final (DCU) .....</b>	<b>78</b>
3.1.9.1	Segunda Validação com o Usuário Final .....	80
<b>3.1.10</b>	<b>Definição dos Casos de Uso (ICONIX) .....</b>	<b>81</b>
<b>3.1.11</b>	<b>Casos de Uso – Pacote Plano de Saúde (ICONIX) .....</b>	<b>81</b>

3.1.12	Casos de Uso – Pacote Prestador de Serviço (ICONIX) .....	83
3.1.13	Casos de Uso – Pacote Proprietário (ICONIX) .....	85
3.1.14	Casos de Uso – Pacote Plano de Saúde (ICONIX) .....	86
3.3.15	Casos de Uso – Pacote Prestador de Serviço (ICONIX) .....	92
3.3.16	Casos de Uso – Pacote Proprietário (ICONIX) .....	96
3.3.17	Diagramas Complementares do Pacote Plano de Saúde (ICONIX).....	98
4.	DESENVOLVIMENTO E VALIDAÇÃO .....	107
4.1	AMBIENTE DE DESENVOLVIMENTO .....	107
4.1.1	Enterprise Architect 7.0 .....	108
4.1.2	Adobe Flex Builder 3.0.....	109
4.1.3	Eclipse 3.0 .....	110
4.1.4	Java .....	110
4.1.5	Apache Tomcat.....	111
4.1.6	DbDesigner .....	111
4.1.7	MySQL .....	111
4.1.8	Corel Draw X3.....	112
4.1.9	BlazeDS.....	113
4.2	APRESENTAÇÃO DO SISTEMA.....	113
4.2.1	Tela de Login.....	113
4.2.2	Tela Gerenciamento Tipo de Serviço.....	114
4.2.3	Tela Gerenciamento Plano de Saúde.....	116
4.2.4	Tela Gerenciamento Plano de Planos com seus Respectivos Serviços.....	118
4.2.5	Tela para Buscar os Serviços e Cadastrar no Plano de Saúde .....	119
4.2.6	Tela de Gerenciamento de Proprietário .....	120
4.2.7	Tela de Cadastro de Proprietário .....	120
4.2.8	Tela de Gerenciamento de Paciente.....	121
4.2.9	Tela do Cadastro de Paciente.....	122
4.2.10	Tela de Gerenciamento de Prestador de Serviço .....	123
4.2.11	Tela de Cadastro do Prestador de Serviço .....	124
4.2.12	Tela de Controle do Tipo de Serviço .....	125
4.2.13	Tela de Inclusão de Serviço ao Prestador de Serviço.....	126
4.3	PROBLEMAS ENCONTRADOS NO DESENVOLVIMENTO .....	127
4.4	VALIDAÇÃO DO PROTÓTIPO .....	127
4.5	CONCLUSÕES RELACIONADAS AO USO DA MÉTODOLOGIA .....	129

4.6	CONSIDERAÇÕES FINAIS.....	129
<b>5.</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS .....</b>	<b>130</b>
	<b>REFERÊNCIAS.....</b>	<b>132</b>



## 1. INTRODUÇÃO

A tecnologia está-se tornando um facilitador para as empresas que a usam como apoio para o seu dia-a-dia. O sistema web oferece rapidez, agilidade e comunicação, podendo ser acessado em qualquer lugar para a transição e gerenciamento de dados.

Segundo Orlandini (2007), a importância do sistema de informação é assegurar a qualidade e facilitar as informações aos seus clientes. Uma grande vantagem, no sistema de informação, proporcionada pela tecnologia, é a capacidade de processar um grande número de dados simultâneos, tornando disponíveis as informações na web.

Segundo Antônio Ferreira (2007), a importância da tecnologia faz com que a comunicação seja ferramenta fundamental em áreas como na venda de produtos e permite que o vendedor/fornecedor faça chegar ao cliente as informações necessárias a respeito de seus serviços.

Hoje existe um número razoável de sistemas que melhoram a vida do usuário, mas existe uma grande dificuldade quanto ao uso dos sistemas. Em alguns casos, o usuário tem dificuldades em tarefas simples, como realizar uma compra *online* ou realizar a marcação de uma consulta. Isso acontece pela dificuldade na interação homem – máquina. Diferentes segmentos da população apresentam características e necessidades diferenciadas, o que é fácil e bom para um pode ser difícil para a maioria. Nesse segmento, temos o conceito de usabilidade, que é sinônimo de facilidade de uso, o usuário aprende mais rápido quando o sistema é mais fácil de usar e a interação entre o mesmo se torna mais eficaz e rápida.

Para atender melhor, o usuário final é fundamental conhecê-lo. A importância do usuário, no projeto, é estimulada pela compreensão de suas necessidades trazendo, clareza ao projetista. A construção de um projeto de software se dá nos dias de hoje permeada pelo uso de metodologias, a metodologia de desenvolvimento é normalmente voltada ao processo ou a função. Uma das metodologias que despontam, no cenário, é o ICONIX. Para Rosenberg (2007 *apud* SILVA, 2007), a metodologia Iconix “consiste na produção incremental e, em paralelo, de um conjunto de artefatos que retratam as visões dinâmicas e estáticas de um sistema, privilegiando a “rastreabilidade” e a robustez“. Apesar de sua abordagem considerada atual, o usuário final é relegado a um papel secundário durante o processo de

desenvolvimento de software. O uso do método não garante a usabilidade do produto desenvolvido.

Este trabalho propõe o estudo da metodologia Iconix e o Método Design Centrado no Usuário e a união dos mesmos com a finalidade de obter um projeto de software adequado às necessidades do usuário.

## 1.1 ESTUDO DE CASO

A empresa estudo de caso é uma Gerenciadora de planos de saúde para animais domésticos com uma rede credenciada de profissionais, clínicas, hospitais, laboratórios e outros órgãos da área veterinária, em Santa Catarina, que prestam serviços aos clientes associados, dentro das limitações de cada plano de saúde, estabelecidos em contrato com o cliente. Os credenciados recebem para tal o pagamento direto da Gerenciadora, dentro do valor e do prazo pré-estabelecidos em contrato no ato do credenciamento.

A empresa não existe no momento, pois todas as idéias sobre o desenvolvimento do sistema serão estabelecidas por um stakeholder que possui todo o conhecimento necessário para o desenvolvimento da gerenciadora de plano de saúde.

## 1.2 PROBLEMA

A descrição da problemática se faz em dois segmentos, o problema da empresa, estudo de caso e o problema da metodologia que não atende a todos os pré-requisitos na parte do desenvolvimento centrado no usuário.

As metodologias tradicionais não se preocupam com a inserção do usuário final durante o processo de desenvolvimento do projeto. A participação está restrita à análise do problema. Isso causa um grande impacto no projeto, que, sem a participação do usuário final, é confrontado, muitas vezes, com uma interface que não atende às suas necessidades.

A metodologia Iconix é uma metodologia que visa a simplificar o desenvolvimento de software, e classificar-se como um meio termo entre metodologias tradicionais e métodos ágeis. Seu processo simplificado toma, como base inicial, os protótipos de interface e começa a desenvolver os casos de uso com base nos requisitos do usuário.

O Iconix vem sendo largamente usado em muitas empresas, mas observa-se, no mesmo, a falta de participação do usuário nas etapas do projeto. Esse fato deflagra problemas,

gerando um usuário insatisfeito com o projeto que foi desenvolvido, assim, gerando alterações após a entrega do projeto.

Estabelece-se, então, a pergunta de pesquisa:

Será possível adequar métodos centrados no usuário à metodologia ICONIX, resultando em um método mais suscetível aos requisitos de usabilidade?

## 1.3 OBJETIVOS

Os objetivos do projeto são divididos em objetivo geral e específico;

### 1.3.1 Objetivo Geral

O objetivo deste projeto é aplicação da metodologia ICONIX, adaptando-se a ela elementos do design centrado no usuário, utilizando-se, como estudo de caso, um sistema de gerenciamento de planos de saúde para animais domésticos.

### 1.3.2 Objetivo específico

São objetivos específicos do projeto:

- analisar a possibilidade de integração dos diferentes modelos;
- fazer uma proposta integrada;
- avaliar o desempenho do método integrado;
- desenvolver um sistema para uma gerenciadora de plano de saúde para animais.

## 1.4 JUSTIFICATIVA

Assim como a definição do problema, este projeto justifica-se por dois ângulos:

Sob o ponto de vista da metodologia de desenvolvimento, o projeto se justifica pela melhoria do processo de desenvolvimento, fazendo uso do Iconix integrado a técnicas de design centrado no usuário. A integração dos métodos deve permitir projetos que tragam uma

maior satisfação das necessidades do usuário final, tendo em vista este estar envolvido em todo o processo de desenvolvimento.

Sob o ponto de vista da empresa, o estudo de caso do sistema deve facilitar a gerência dos serviços da clínica e a melhor gestão dos planos de atendimento animal em questões vitais, como armazenamento de dados dos diferentes clientes, veterinários e serviços prestados.

## 1.5 PROPOSTA DE SOLUÇÃO

Na proposta de solução, serão abordados dois cenários:

- solução para o estudo de caso;
- solução da integração das metodologias.

### 1.5.1 Solução para o Estudo de Caso

O sistema será desenvolvido para a web. Para que o controle do sistema seja simples e eficaz, serão definidos níveis de acesso para cada usuário. Os dados serão verificados em um banco de dados que se encontra em um servidor. Após a entrada, o usuário pode realizar suas tarefas, normalmente, e gerenciar seus dados e aplicativos de acordo com seu nível de acesso.

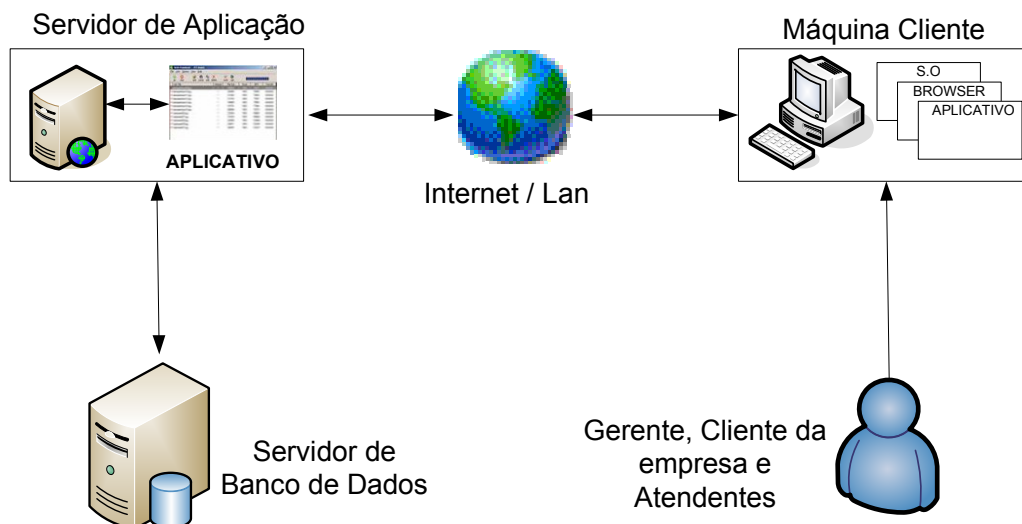


Figura 1 - Solução para o Estudo de Caso

Fonte: elaboração dos autores (2007)

### 1.5.2 Solução para a Integração das Metodologias

Para a realização da integração entre a metodologia Iconix e o método design centrado no usuário, serão realizados estudos de cada um deles, considerando-se a etapa de integração entre as duas metodologias de desenvolvimento. Seguem, na figura 2, as etapas desse estudo.

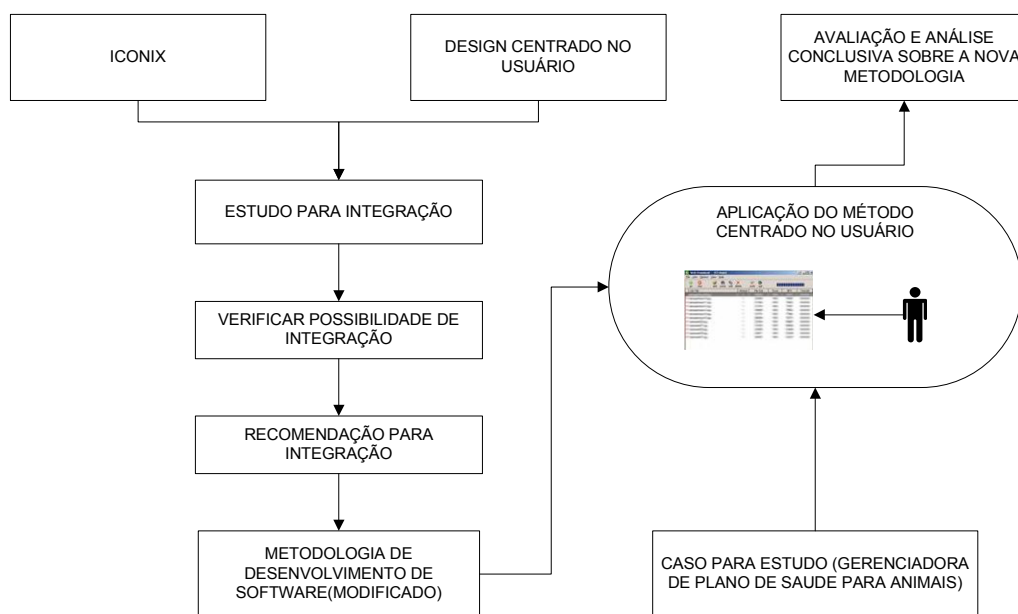


Figura 2 - Estudo de Integração entre a Metodologia e o Método

Fonte: elaboração dos autores (2007)

Os autores iniciarão o projeto a partir da revisão bibliográfica do Iconix e do design centrado no usuário. O entendimento sobre esses temas deve possibilitar a análise para definir a possibilidade de integração de pontos que tornem o Iconix um método mais adequado sob o ponto de vista usuário final.

Finalizada a análise e definidos os métodos possíveis de serem integrados à equipe, irão iniciar a etapa de modelagem, aplicando a integração. Finalizado o desenvolvimento, serão analisados os resultados a partir de pontos positivos e negativos da aplicação dos métodos.

## 1.6 DELIMITAÇÃO

A seção tem como objetivo definir as fronteiras do contexto trabalhado, delimitando o que será realizado no período compreendido para a realização da monografia:

- O desenvolvimento, em termos de codificação, compreenderá somente o pacote de gerência do plano de saúde.
- O sistema será modelado como um todo até os diagramas de casos de uso e os demais diagramas compreenderão apenas o pacote gerencia de plano de saúde.
- A integração do design centrado ao usuário e o método ICONIX será realizado, conforme descrição da metodologia definida no capítulo 3.

## 1.7 ESTRUTURA DA MONOGRAFIA

Este projeto está subdividido em cinco capítulos. A seguir, apresentaremos a estrutura do trabalho:

- Capítulo 1 – Introdução: Esse capítulo apresenta a introdução, justificativa, objetivos, problemas e a solução proposta como trabalho.
- Capítulo 2 – Revisão Bibliográfica: Nesse capítulo, será desenvolvido o domínio do problema relacionado à engenharia de software e a modelos de desenvolvimento para empresas de software e ao design centrado no usuário.
- Capítulo 3 – Modelagem: Nesse capítulo, serão descritos os passos realizados durante a modelagem, uso da metodologia compartilhada, apresentando gráficos e descrições necessárias para a compreensão do protótipo, utilizando-se a análise orientada a objetos.
- Capítulo 4 – Desenvolvimento e Validação: Nesse capítulo, serão apresentados as ferramentas utilizadas para o desenvolvimento do estudo de caso e a apresentação do sistema.

- Capítulo 5 – Conclusões e trabalhos futuros: Nesse capítulo, serão apresentados as considerações finais sobre o desenvolvimento do projeto e os trabalhos futuros sobre o mesmo.

## 2. ENGENHARIA DE SOFTWARE

O software, hoje em dia, está em todos os âmbitos, e em constante crescimento. Cada vez mais pessoas necessitam do produto software. Com esse crescimento, o software ganhou um papel muito importante na sociedade.

Quando o cliente recebe o produto (software) quase nunca é considerado perfeito, há sempre uma correção a ser feita, ou seja, a manutenção é constante. O software passa por vários estágios. Aos poucos se deprecia, muitas vezes, a tal ponto que a empresa percebe que não tem como mais manter o produto no mercado, substituindo-o por um novo produto ou versão.

Segundo Sommerville (2003), engenharia de software são todos os processos de software desde o seu estado inicial, na produção, até a fase de manutenção do sistema.

Na engenharia de software, os engenheiros aplicam teorias, métodos de desenvolvimento e ferramentas em um processo seletivo, buscando soluções para os problemas mesmo quando não existem teorias aplicáveis e métodos que apóiam no desenvolvimento.

A engenharia de software adota uma abordagem sistemática para a produção de software de alta qualidade selecionando métodos, ferramentas e teorias para o melhor processo de desenvolvimento do software.

A engenharia de software não se destina somente a processos técnicos de desenvolvimento, dedica-se também, ao gerenciamento de projetos, desenvolvimento de ferramentas, métodos de desenvolvimento e teorias que apóiam a produção do software.

Segundo Pressman (2006), o software e o hardware possuem semelhanças no decorrer do desenvolvimento. Os dois conquistam a qualidade através de um bom projeto.

### 2.1 O DESENVOLVIMENTO EM CAMADAS.

Segundo Pressman (2006), a engenharia de software é uma tecnologia em camadas.



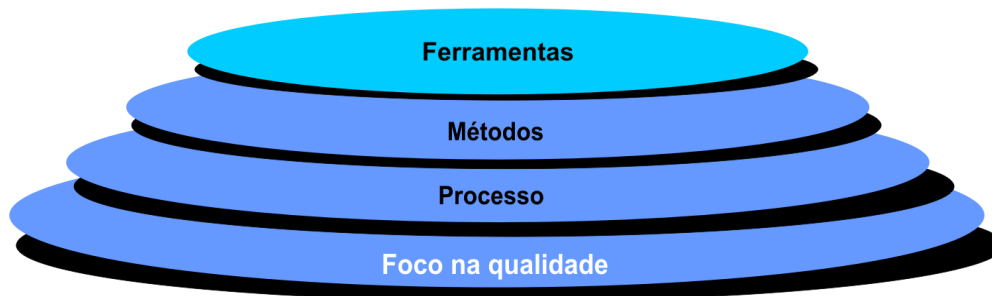


Figura 3 – Engenharia de Software em Camadas.  
Fonte: Pressman (2006, p.17).

O processo de engenharia de software é que mantém unido as camadas de tecnologia que permitem o desenvolvimento de software de computador. O processo define como deve ser utilizada a tecnologia de engenharia de software, formando uma base de controle gerencial nos projetos em que os métodos são aplicados nos produtos de trabalho, assim, os marcos são estabelecidos, e a qualidade é assegurada no decorrer do desenvolvimento.

Os métodos de engenharia de software definem as técnicas de como desenvolver o software, incluindo as modelagens de desenvolvimento e metodologias a serem aplicadas.

As ferramentas de engenharia de software apóiam o desenvolvimento de software baseadas em métodos. Por exemplo, existem ferramentas que apóiam a modelagem ou a descrição de processos, entre outras. Essa prática é chamada de “Engenharia de Software apoiada”.

## 2.2 ARCABOUÇO DE PROCESSO DE SOFTWARE.

Segundo Pressman (2006), um arcabouço de processo de software estabelece um alicerce para os processos de software que podem ser aplicados a projetos de qualquer tamanho. O arcabouço de processo é o desenvolvimento do projeto em blocos em que cada pessoa tem uma função naquele determinado bloco de atividades.

Na maioria dos projetos, existe um arcabouço genérico que é aplicado no desenvolvimento do projeto de software, dividido em cinco fases:

- **Comunicação:** abrange a atividade que envolve a comunicação entre o cliente e os interessados no levantamento de requisito e atividades relacionadas ao projeto.

- **Planejamento:** Abrange a descrição das tarefas técnicas a serem conduzidas, os riscos, os recursos necessários, os produtos de trabalho para serem produzidos e o cronograma do projeto.
- **Modelagem:** é uma atividade que inclui o desenvolvimento de modelos, requisitos de software para o melhor entendimento do cliente junto ao projeto o que será realizado durante as etapas do desenvolvimento.
- **Construção:** é a atividade que gera a implementação do código e o teste do produto em busca de erros durante o processo de desenvolvimento do aplicativo.
- **Implantação:** é o produto que o cliente recebe para avaliação. Neste momento, o cliente avalia o produto, e, caso surgirem novas solicitações, retorna para a fase de planejamento, (PRESSMAN, 2006).

Essas cinco atividades apresentadas podem fazer parte de um projeto de software grande ou pequeno, pois os processos se aplicam para qualquer tipo de projeto.

## 2.2.1 Modelos de processos de software

No decorrer dos últimos anos, vários modelos foram lançados no mercado. A seguir, são apresentados alguns deles.

### 2.2.1.1 Modelo cascata

Segundo Pressman (2006, p.38), o modelo cascata possui um ciclo de vida clássico em que são abordados os principais estágios do modelo de desenvolvimento, abordando, assim, um modelo sistemático e seqüencial.

Raros são os projetos que seguem um fluxo seqüencial, como o sugerido pelo modelo, mesmo que o modelo acomode iterações, as modificações podem causar confusão na equipe no decorrer do projeto. Um dos pontos é o fato de que, no início do projeto, é raro o cliente estabelecer todos os requisitos, e o modelo cascata exige esse conhecimento em uma etapa muito inicial.

Nesse modelo, o cliente precisa ter muita paciência, pois o executável do software é gerado no final da iteração.

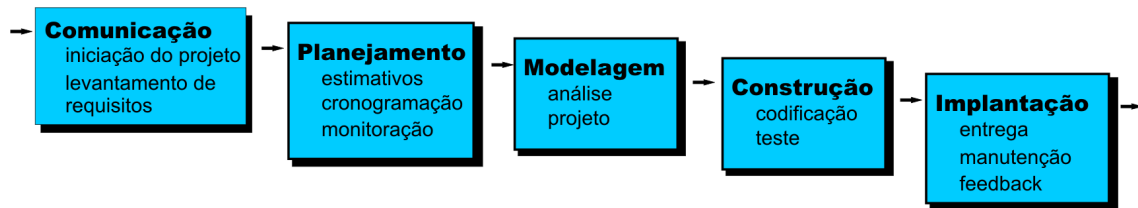


Figura 4 – Modelo Cascata.  
Fonte: Pressman (2006, p.39).

#### 2.2.1.2 Modelo Evolucionário

Segundo Sommerville (2003, p.39), o modelo evolucionário é o desenvolvimento de uma implementação inicial, apresentando o sistema para o cliente para realização de aprimoramentos. Esse método desenvolve várias versões até que o sistema esteja adequado às necessidades do cliente. O modelo pode ser classificado em:

- Exploratório: possui o objetivo de trabalhar com o cliente a fim de observar os requisitos e entregar uma versão final ao cliente, o sistema evolui de acordo com as especificações do cliente.
- Protótipos descartáveis: o objetivo é compreender os requisitos passados pelo cliente, criando protótipos do sistema a fim de melhor definir os requisitos que não foram bem compreendidos. Com esse artefato, inicia-se o desenvolvimento.

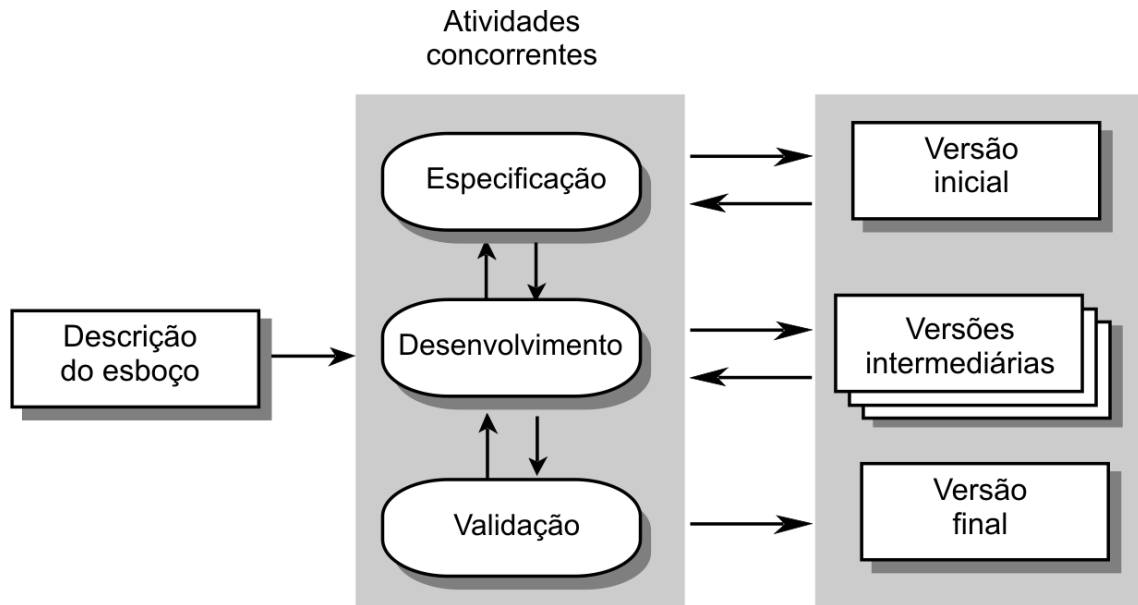


Figura 5 – Modelo Evolucionário.  
 Fonte: Sommerville (2003, p.39).

A abordagem, nesse método, é mais eficaz que o modelo cascata por atender melhor as necessidades dos clientes. A grande vantagem desse modelo é que o desenvolvimento é gradativo. Segundo Sommerville (2003), essa abordagem possui três problemas: O processo não é visível, os sistemas são, freqüentemente, mal estruturados e podem ser exigidas ferramentas e técnicas específicas.

#### 2.2.1.3 Modelo Espiral

Segundo Pressman (2006, p.44), modelo espiral é um modelo de desenvolvimento de software que utiliza o modelo evolucionário em parceria com o modelo cascata. Nesse modelo, as iterações são realizadas de uma forma de espiral, como mostra a figura a seguir.

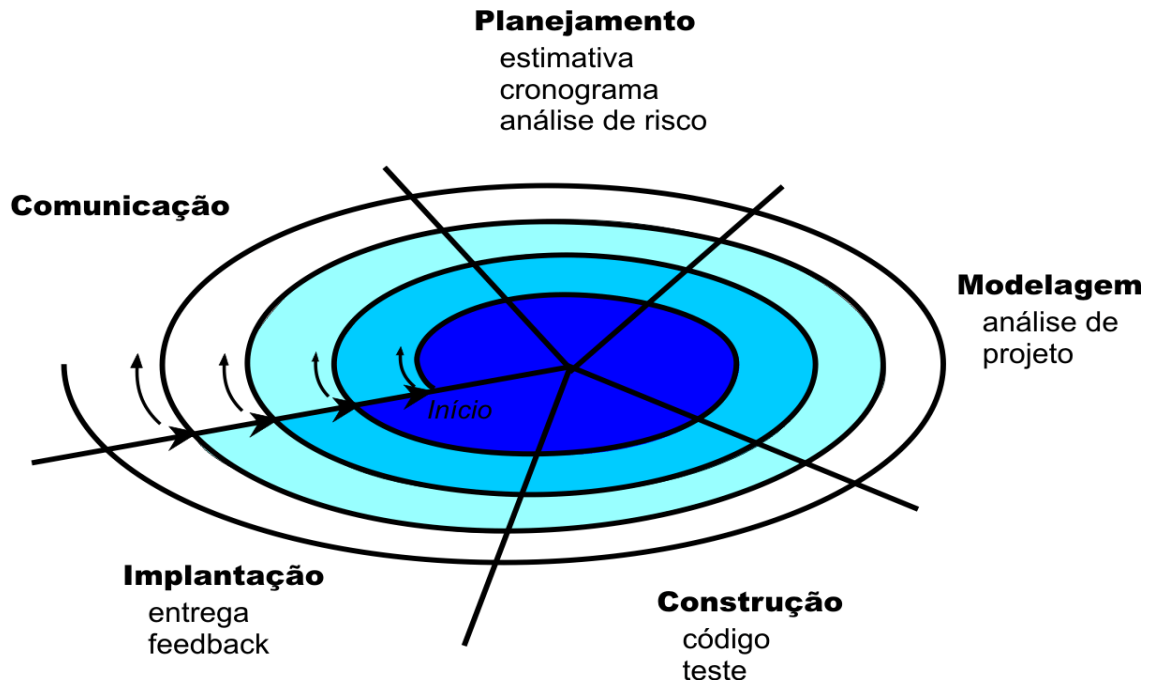


Figura 6 – Modelo Espiral.  
Fonte: Pressman (2006, p.45).

Esse modelo aborda o desenvolvimento de uma forma evolucionária, originando, assim, várias versões: o início se dá com uma nova iteração, assim, o software ganha novas versões a cada iteração, no final das iterações, o software é entregue ao cliente.

O modelo aborda o desenvolvimento de software de forma realista, em que em cada iteração, o software evolui tornando-se mais maduro.

#### 2.2.1.4 Modelo Incremental

Segundo Pressman (2006, p.40), o modelo incremental utiliza-se do modelo cascata nas suas iterações. Cada iteração do projeto segue o modelo cascata, incrementando-se e surgindo, assim, a cada final de estágio, unido uma nova versão do software.

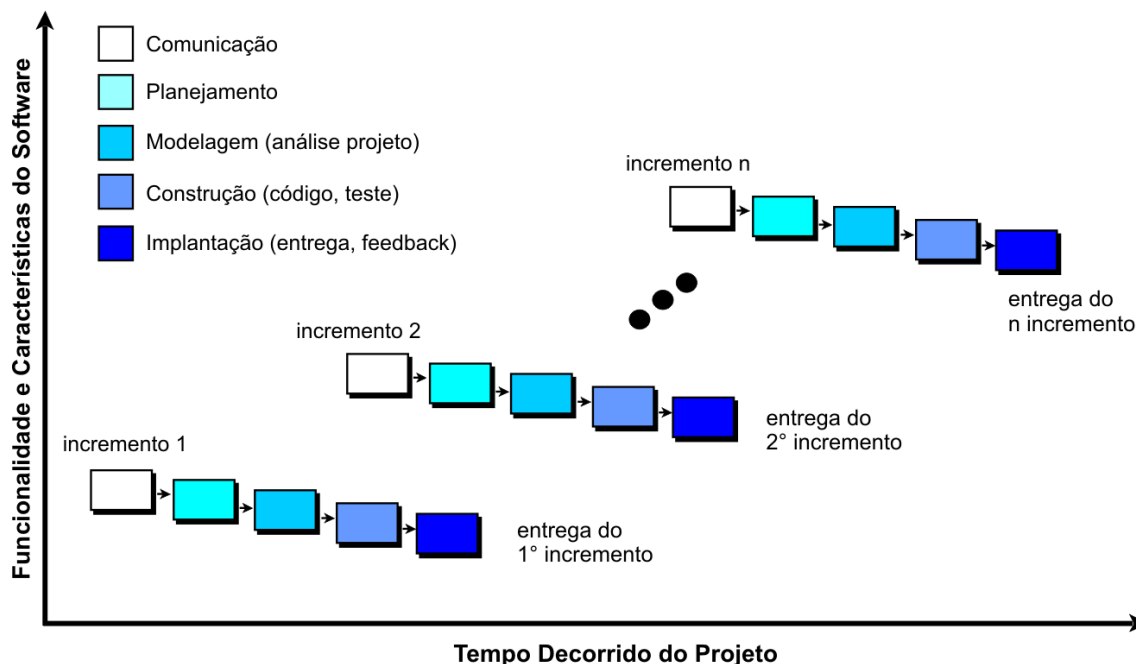


Figura 7 – Modelo Incremental.  
 Fonte: Pressman (2006, p.40).

No modelo incremental, o primeiro incremento ou a primeira iteração é usada como núcleo do produto. Neste núcleo, os requisitos básicos do software são satisfeitos na primeira iteração. Muitas vezes, o cliente já começa a usar o software. Esse modelo tem por base a melhoria contínua do software, pois o cliente já usa uma versão para avaliação. A utilização do cliente, logo em fases iniciais, faz com que sejam solicitadas novas funcionalidades e melhorias nos processos das atividades do software.

Apesar dos inúmeros modelos existentes, percebeu-se a necessidade de flexibilização no processo de desenvolvimento de software e, a partir dessa percepção, deu-se início ao movimento ágil.

### 2.2.2 Processo de Desenvolvimento Ágil

A agilidade, no desenvolvimento de software, é um tema recorrente para equipes de desenvolvimento. Essa ansiedade deu forma a um movimento chamado Movimento Ágil, que revolucionou a forma de desenvolver a software no mercado.

Segundo Jack (2002 *apud* PRESSMAN, 2006, p.59), o tema agilidade fornece uma discussão grande no desenvolvimento de software:

Agilidade tornou-se atualmente uma palavra mágica quando se descreve um processo moderno de software. Uma equipe ágil reconhece que o software é desenvolvido por indivíduos, trabalhando em equipe em que as especialidades dessas pessoas e a sua capacidade de colaborar estão no âmago do sucesso do projeto (Jack 2002 *apud* PRESSMAN, 2006, p.59).

De acordo com Jack (2002 *apud* PRESSMAN, 2006, p.59), “o acolhimento de modificações é o principal guia para a agilidade. Os engenheiros devem agir rapidamente se tiverem que acomodar rápidas modificações”.

O desenvolvimento ágil enfatiza a rápida entrega do software, mas, para que isso ocorra, o plano de projeto deve ser flexível no desenvolvimento do processo. Os conceitos do manifesto ágil são: “Indivíduos e interação, ao invés de processos e ferramentas, Software executável, ao invés de documentação, Colaboração do cliente, ao invés de negociação de contratos e respostas rápidas as mudanças ao invés de seguir planos” (SOARES, 2007).

A agilidade pode ser aplicada em qualquer processo de software, mas, para que isso ocorra, o processo deve ser projetado para que a equipe de projeto se adapte as tarefas.

Fower (2002 *apud* PRESSMAN p.60, 2006) comenta que uma grande dificuldade é prever os requisitos de software, antecipadamente, e saber quais serão modificados. Para isso, o processo ágil deve ser adaptável. Muitos tipos de softwares são construídos intercalados, sendo que análises, projetos construção e teste não são previsíveis e nem sempre o resultado é o esperado. Em um processo ágil de software, a adaptação incremental é fundamental. Para isso, uma equipe ágil requer feedback constante do cliente.

De acordo com Cocol (2002, *apud* PRESSMAN, 2006, p.61), “o desenvolvimento ágil enfoca os talentos e habilidades dos indivíduos, moldando o processo e equipes específicas”.

A auto-organização traz benefício à empresa, mas o objetivo principal é aumentar a moral da equipe de desenvolvimento. A equipe seleciona o trabalho que acredita poder realizar dentro da interação, e a equipe se compromete com o trabalho. Nada motiva tanto uma equipe quanto a aceitação da responsabilidade de cumprir os compromissos que ela própria estabeleceu.

### 2.2.2.1 Extreme Programming (XP).

Segundo Soares(2007), o XP é uma metodologia de desenvolvimento ágil para pequenas empresas, que devem desenvolver projetos com requisitos confusos, pois se modificam rapidamente. Entre as diferenças encontradas, nas outras metodologias, estão o feedback constante, a abordagem incremental e a comunicação.

O XP é uma metodologia de desenvolvimento que se baseia em 12 práticas no desenvolvimento de sistemas. As práticas serão mostradas a seguir:

- Planejamento: essa prática baseia-se no que deve ser feito em cada etapa e o que pode ser adiado durante as etapas do desenvolvimento.
- Entregas frequentes: nesta prática visa-se à construção do software. Conforme surgem novos requisitos e funcionalidades a ser implementadas, o software ganha novas atualizações (*releases*).
- Metáfora: são as descrições de um software, não utilizando os termos técnicos, com o intuito de orientar o desenvolvimento de software.
- Projeto Simples: No desenvolvimento do sistema, ele deve ser o mais simples possível, satisfazendo os requisitos atuais, sem ter a preocupação de novos requisitos.
- Testes: é focalizado na validação do projeto durante a realização de todo o processo de desenvolvimento. Os softwares são liberados em *releases* e já são testados.
- Refatoração: está focado no aperfeiçoamento do projeto do software. Essa prática está focada em todo desenvolvimento.
- Programação em pares: A implementação do código fonte é desenvolvido por dois programadores em um só computador, em que o código é desenvolvido pelos dois programadores em que um desenvolve e o outro está em constante observação.



- Propriedade coletiva: o código do projeto pertence a todos os membros da equipe. Significa que qualquer pessoa do grupo pode mexer no código fonte mesmo que não tenha desenvolvido.
- Integração contínua: nessa prática, os programadores entram em sintonia, pois o sistema de software constrói e interage várias vezes por dia, integrando um conjunto de informações.
- 40 horas de Trabalho semanal: no XP, assume-se que ninguém pode trabalhar mais de 40 horas semanais.
- Cliente presente: Durante toda a etapa do desenvolvimento, é fundamental a participação do cliente para que o mesmo possa tirar todas as dúvidas durante o desenvolvimento do projeto.
- Código padrão: o código é todo padronizado para que todos os programadores possam entender o que está escrito.

#### 2.2.2.2 Scrum

Segundo Soares (2007), o Scrum é outra metodologia de desenvolvimento que fornece uma abordagem para o desenvolvimento orientado a objeto. O modelo apresenta uma abordagem empírica que aplica a teoria de controle de processos industriais para o desenvolvimento de software, introduzindo as idéias de flexibilidade, adaptabilidade e produtividade.

O Scrum envolve muitas variáveis técnicas e do ambiente, tais, como, requisitos recursos e tecnologia que podem ser mudados durante o processo, com isso, o desenvolvimento se torna imprevisível e complexo, necessitando de flexibilidade para acompanhar todas as mudanças.

A metodologia Scrum tem por base princípios comuns aos da metodologia XP como equipes pequenas, requisitos não muito estáveis ou desconhecidos e iterações curtas para promover a visibilidade no desenvolvimento.

O Scrum possui um ciclo de vida baseado em três fases que são:

- Pré - planejamento: em que os requisitos são descritos em um documento que é chamado de *backlog*, após isso, são realizadas estimativas de esforço para o desenvolvimento de cada requisito.
- Desenvolvimento: as muitas variáveis técnicas e do ambiente identificadas previamente são controladas durante toda a etapa do projeto ao invés de outras metodologias que consideram só o início do projeto.
- Pós – planejamento: após a fase de conclusão do desenvolvimento, são realizadas reuniões para analisar o progresso do projeto e mostrá-lo aos clientes. Nessa fase, são realizados testes e a documentação.

## 2.3 ICONIX

Segundo Rosenberg et al. (1999 *apud* BONA, 2002), a metodologia ICONIX foi elaborada por Doug Rosenberg e Kendall Scott, a partir do processo simples e unificado dos pesquisadores Booch, Rumbaugh, e Jacobson. Essa união deu suporte e conhecimento para o desenvolvimento a metodologia a partir de 1993, conforme a figura 10.

Para Rosenberg et al. (1999 *apud* BONA, 2002), o Iconix é uma metodologia que visa a simplificar os processos do ciclo de vida no desenvolvimento. Essa metodologia aborda um conjunto de métodos de orientação a objetos de desenvolvimento prático e imediato.

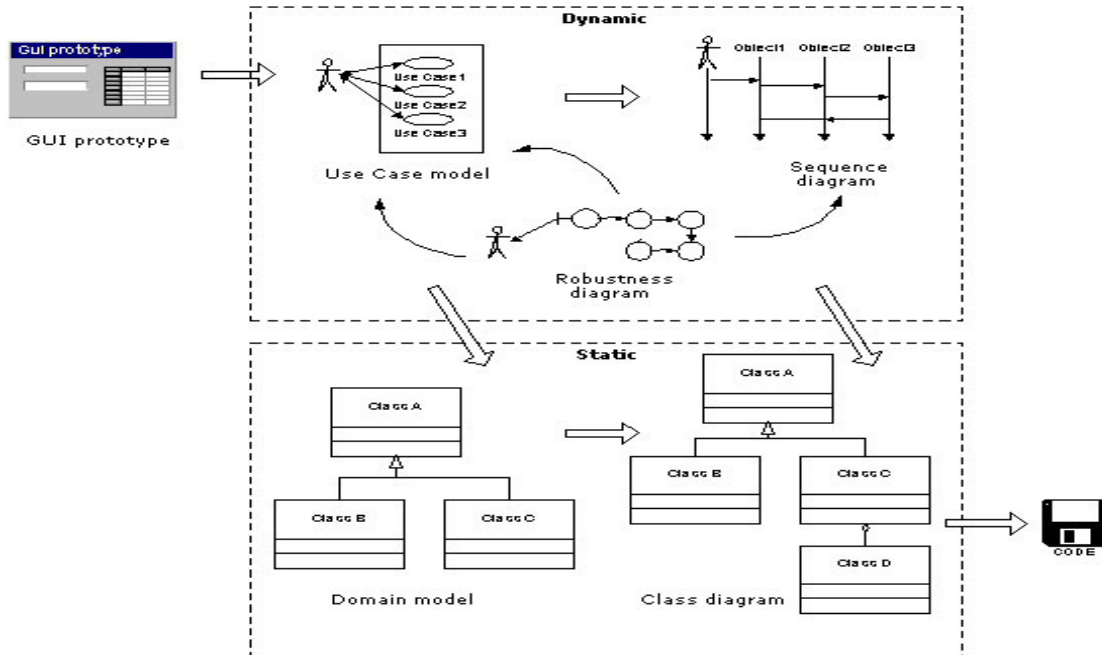


Figura 8 – Processo Iconix.  
Fonte: Bona (2000, p.60).

### 2.3.1 Questões fundamentais da metodologia Iconix

Para Rosenberg et al. (1999 *apud* BONA, 2002), a metodologia Iconix tem por base responder as seguintes questões para a realização do desenvolvimento de software:

- Quais são os usuários do sistema e quais tarefas eles realizam? (Elaboração de casos de uso).
- O que são, no mundo real, os objetos e as associações entre as tarefas? (Elaboração de diagrama de classe de alto nível).
- Quais são os objetos necessários para cada caso de uso? (Elaboração de análise de Robustez).
- Como os objetos estão colaborando e interagindo entre os outros objetos? (Elaboração do diagrama de sequência).
- Como serão manipulados os aspectos em tempo real na estrutura de controle? (Elaboração de diagrama de estado).

- Como será a construção do sistema no nível prático? (Elaboração do diagrama de classe de baixo nível).

### 2.3.2 Características fundamentais da metodologia Iconix

De acordo com Borillo (2000 *apud* BONA, 2002), a metodologia Iconix possui três características fundamentais:

**Iterativo e incremental:** em que são realizadas várias iterações durante o modelo de desenvolvimento do domínio e a identificação dos casos de uso.

**Rastreabilidade:** expõe como se relacionam os diferentes artefatos produzidos, observando o impacto que alterações do projeto podem produzir como um todo.

**UML (Unified Modeling Language):** a metodologia oferece uma aerodinâmica no processo de desenvolvimento, tais como, os diagramas de casos de uso, diagramas de sequência e colaboração, diagramas de robustez.

### 2.3.3 Desenvolvimento Aplicado à Metodologia Iconix

De acordo com Rosenberg et al. (1999 *apud* BONA, 2002), a metodologia possui as tarefas de análise e projeto preliminar, projeto e implementação.

#### 2.3.3.1 Análise de Requisitos

De acordo com Bona (2002, p.62); para a realização de análise de requisitos, são abordadas as seguintes iterações na etapa do projeto:

- Serão identificados os objetos no mundo real e todas as relações dentre eles como agregação e generalização, aplicando o modelo de domínio.
- Será apresentado uma prototipação ou um diagrama de navegação para que o cliente possa entender melhor o sistema que será proposto.
- Identificar os casos de uso, mostrando os atores envolvidos no sistema.

- Organizar os casos de uso em pacotes para possuir um melhor entendimento entre eles.
- Associar os casos de uso aos requisitos funcionais do sistema e os objetos de domínio.

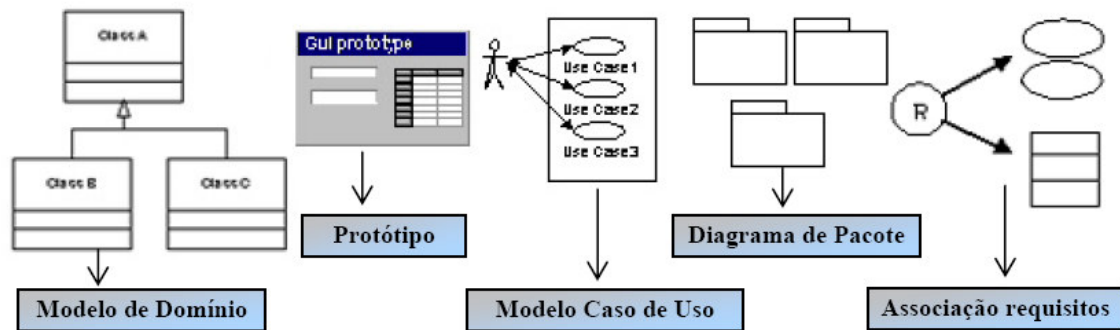


Figura 9 - Análise de Requisitos.  
Fonte: Bona (2000, p.62).

Na etapa de análise de requisito, são abordados modelos de domínio, protótipos, modelos de casos de uso, diagrama de pacote e associação dos requisitos.

### 2.3.3.2 Análise e Projeto Preliminar

De acordo com Bona (2002, p.63), a etapa de análise e projeto preliminar compõe as seguintes iterações:

- são escritos os casos de uso contendo fluxo principal das ações e, também, o fluxo alternativo e de exceção;
- apresentação de análise de robustez aplicado para cada caso de uso, identificando um conjunto de objetos, realizando a atualização do modelo de classe e de domínio;
- finalizar a atualização dos diagramas de classe.

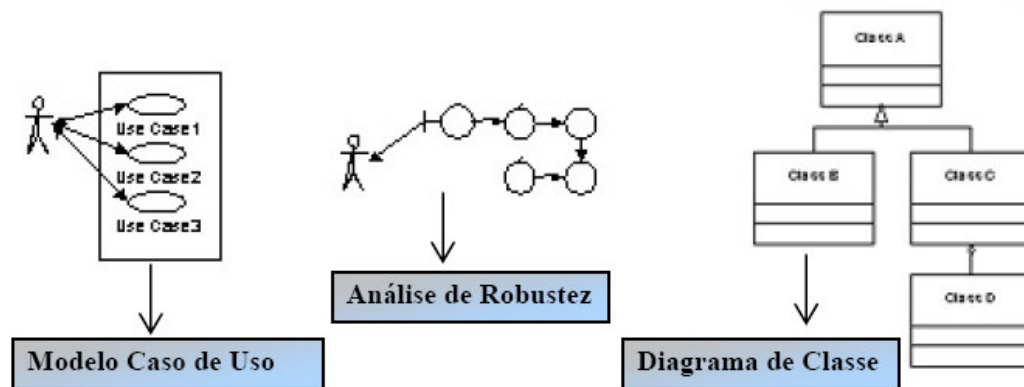


Figura 10 - Atividades da análise e do projeto preliminar.  
Fonte: Bona (2000, p.63).

Com essas tarefas realizadas, é finalizado o segundo macro com a apresentação de casos de uso, análise de robustez, atualização dos modelos de domínio, modelo de classe e diagramas de classes.

### 2.3.3.3 Projeto

Segundo Bona (2002, p.64), as tarefas para a realização desta etapa são compostas pelas seguintes atividades:

- realizar especificações de comportamento, utilizando o diagrama de seqüência, nos casos de uso identificando as mensagens entre os diferentes objetos, caso seja necessário, será utilizado o diagrama de colaboração para representar as transações entre as chaves e objetos. O diagrama de estado deve mostrar o comportamento de forma real;
- finalizar o modelo estático, adicionando detalhes ao projeto, utilizando o diagrama de classe;
- realizar a verificação sobre os requisitos apresentados.

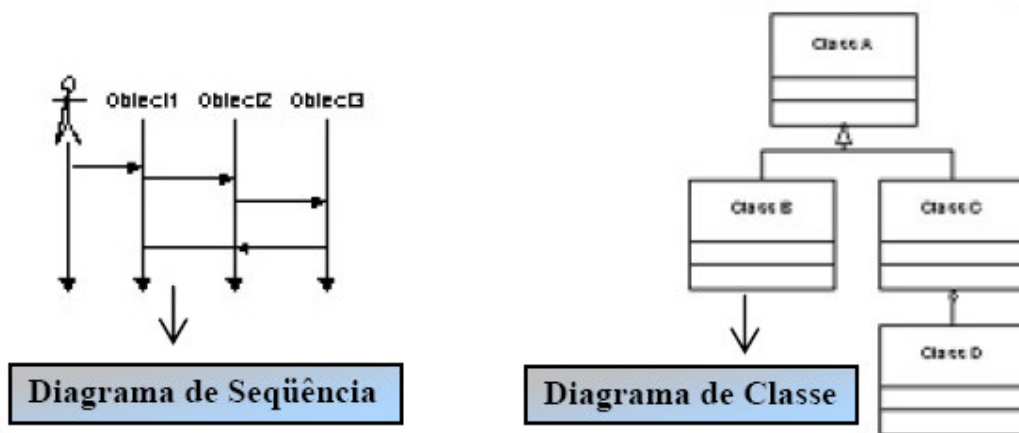


Figura 11 - Atividades do Projeto.  
 Fonte: Bona (2000, p.64).

Com essa etapa, dá-se o término do terceiro macro, finalizando a etapa de projeto passando-se para a próxima etapa: a implementação.

#### 2.3.3.4 Implementação

Segundo Bona (2002, p.64), as etapas de implementação são compostas pelas seguintes atividades:

- serão realizados os diagramas de componentes como apoio para a fase de desenvolvimento;
- implementar (escrever) o código;
- realizar os testes para verificação de integrações;
- realizar testes com o usuário, verificando a aceitação.

Com a realização dessa etapa, inicia-se a quarta macro que é a entrega do sistema ao usuário final.

#### 2.3.4 Modelos de Diagramas usados na metodologia Iconix

São descritos, a seguir, os diagramas usados pela metodologia Iconix.

#### 2.3.4.1 Modelo de Domínio

De acordo com Blaha et al. (2006, p.189), para a criação do modelo de domínio, deve-se analisar os requisitos, montar a estrutura estática do sistema de acordo com o mundo real e organizá-las em partes utilizáveis, mostrando as relações e iterações entre os objetos.

Para se estabelecer o modelo de domínio, é necessário:

- descobrir quais classes;
- dar início à montagem do dicionário de dados;
- encontrar as associações entre as classes;
- adicionar os atributos nas classes e realizar as ligações entre os objetos;
- verificar a existência de caminhos de acesso para realização de consultas;
- refinar o modelo;
- considerar o nível de abstração;
- adicionar as classes em pacotes.

Segundo Borillo (2000, *apud* BONA 2002), alguns passos podem auxiliar no desenvolvimento do modelo de domínio:

- para descobrir as classes, deve-se ler os requisitos de baixo nível, problemas de alto nível;
- eliminar as classes que são desnecessárias;
- construir a relação de generalização;
- construir a relação associação entre uma classe e outra, mostrando as dependências entre as classes e não as ações entre elas.

#### 2.3.4.2 Modelo de Caso de Uso

Segundo Blaha et al. (2006, p.139), um caso de uso é a relação entre os atores e as tarefas em que cada caso de uso corresponde a uma determinada tarefa que pode ter vários atores.

Um sistema corresponde a um conjunto de casos de uso e atores. Cada caso de uso corresponde a uma funcionalidade do sistema, cada ator se comporta diferente dentro do



sistema. Um caso de uso corresponde a interação entre o ator e a funcionalidade que está sendo aplicada.

Os casos de uso representam uma seqüência de funcionalidades na realização de tarefas.

#### 2.3.4.3 Análise de Robustez

Análise de robustez consiste em analisar cada caso de uso e transformá-lo em objetos. Os objetos são classificados em: fronteira, identidade e de controle.

Segundo Blaha et al. (2006, p.401), a análise de robustez deve observar as seguintes situações, no momento em que estiver elaborando:

- **Proteja-se de erros:** O software deve ser protegido contra entrada de dados errados, realizados por usuários, para evitar a resposta inapropriada para o usuário e evitar quedas do sistema. Os erros são divididos em dois tipos:
  - Erros do hardware ou sistema operacional, em que o sistema deve lidar com erros de hardware como impressoras, falta de alocação de memória, entrada/saída de dados, o sistema deve tratar de todas essas ocorrências, para que o sistema mostre uma resposta apropriada para o usuário ou encerre o programa de uma forma “mais elegante possível”, caso nada possa ser feito.
  - Erros de *bugs* do programa construído: o sistema deve ter uma boa interação nos erros, mostrando ao usuário os erros e facilitando sua correção.
- **Otimize, após a execução do programa:** A otimização do programa só deve ser feita depois que o programa estiver funcionando.
- **Validação dos argumentos:** o argumento público deve estar protegido contra entrada de dados inválidos, pois os usuários podem violar as restrições, os métodos privados ou públicos.
- **Evitar limites pré-definidos:** a alocação de memória, sempre que for possível, deve ser dinâmica, pois as linguagens de programação têm boas facilidades de alocação dinâmica de memória.

#### 2.3.4.4 Diagrama de Classe

Diagrama de classe é onde são representados os atributos, métodos e os relacionamentos com outros objetos. Os objetos são agrupados por suas características mais semelhantes. Os objetos são classes vistas pelo mundo real, pois são extraídas de várias análises e comportamentos do mundo real. A classe está agrupada em atributos, e os atributos e métodos podem ser privados, protegidos ou públicos.

#### 2.3.4.5 Modelagem de Interação

Segundo Rosenberg et al. (1999 *apud* BONA, 2002), a modelagem de interação visa a construir as linhas de execução do sistema, unindo os objetos com os atores relacionados, analisando, assim, se o sistema terá um comportamento útil. Essa modelagem se dá no uso do diagrama de seqüência, em que o diagrama nos permite a colaboração dinâmica entre vários objetos do sistema. Segue, abaixo, um modelo de diagrama de seqüência:

### 2.4 DESIGN CENTRADO NO USUÁRIO

Para realização de um bom projeto com perspectiva de sucesso ao final, é necessária a integração com o usuário final. Para isso, não basta conhecê-lo, mas identificar os pontos que apóiam o projeto a fim de atingir suas metas.

Segundo Microsoft (2000 *apud* AGNES et al., 2007), o ciclo do design centrado no usuário da empresa envolve as seguintes fases:

- **Desenhar:** Nessa fase, são analisadas, projetadas as telas, modelar as aplicações a serem feitas.
- **Prototipar:** Ocorre, nessa etapa, a aplicação das telas no sistema, que são refeitas várias vezes se necessário até que a etapa seja concluída.
- **Testar:** Os testes são realizados de forma que se encontrem erros e que sejam corrigidos.

- **Redesenhar (refinar):** Etapas em que se aplicam o melhoramento das interfaces, com a visibilidade, usabilidade e compatibilidade.

Segundo Agner et al. (2007), um *site* bem sucedido deve respeitar seus objetivos de usuário, comportamento e obter uma interface adequável e útil ao usuário de forma clara e entendível.

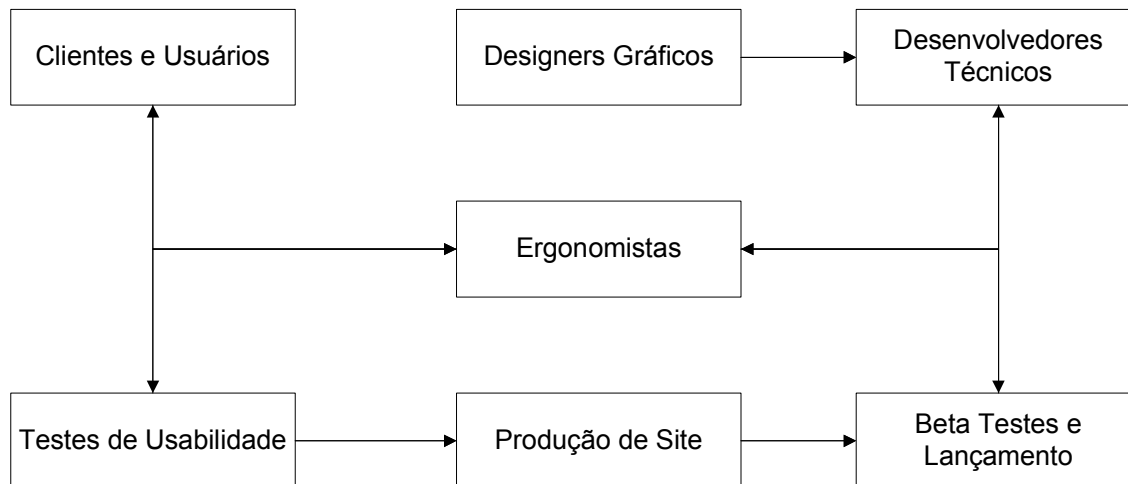


Figura 12 – Modelo de Integração ao Design Centrado no Usuário.  
Fonte: Bishu (2000 *apud* AGNER et al.).

A figura, acima, mostra que o ergonomista é o foco de interação com clientes/usuários e com os desenvolvedores técnicos.

O ergonomista é quem faz a entrevista com o cliente, então, é preciso entender o que está sendo pedido, observar regra e o comportamento de forma a fazer melhoramentos no processo de concepção. Feito isso, inicia-se a etapa de desenvolvimento. Os ergonomistas aplicam, por sua vez, testes de usabilidades, finalizados os testes, realiza-se a produção do site e uma nova bateria de testes, a seguir é lançada uma versão beta<sup>1</sup>. Segundo Rational (2007), para desenvolver sistemas que satisfaçam as necessidades do usuário, é necessário um levantamento de análise de requisitos o qual consome um grande esforço. Esse esforço no design centrado no usuário é focado em usuários finais. Esses usuários podem ser vistos a partir dos seguintes papéis e ações :

<sup>1</sup> Beta é uma versão do site com mínimos detalhes de erros.

- **Atores de Negócios:** Um ator de negócio representa um papel designado a fazer algo no ambiente de negócio ou um negócio para alguém.
- **Trabalhadores do negócio:** São trabalhadores de negócio que interagem entre si.
- **Atores:** Descrevem um indivíduo que está envolvido no projeto.
- **Modelagem de negócios:** Realiza a apresentação de um fluxograma de trabalho.
- **Diretrizes:** Promovem uma transição de modelos de negócios para sistemas.

Um ponto importante é que os artefatos, citados acima, devem ser compreendidos e descritos de acordo com seus papéis.

Beyer (1998 *apud* RATIONAL 2007) sugere "...vá aonde o cliente trabalha, observe o cliente enquanto ele trabalha e converse com o cliente sobre o trabalho."

### 2.4.1 Técnicas de Design Centrado no Usuário

A seguir, serão apresentadas algumas técnicas de design centrado no usuário.

#### 2.4.1.1 Cenários

Segundo Souza (2007, p.25), cenários são narrados no contexto. Com isso, pode-se obter um bom levantamento de requisitos com foco no trabalho específico de um usuário para poder esclarecer ao projetista as tarefas do usuário. É uma técnica que permite analisar o usuário, explicitando como realiza seu trabalho, pois, em muitos casos, em um levantamento de requisitos, é difícil compreender o que o usuário quer passar para o desenvolvedor. Com essa técnica, as situações são fictícias, cenas em que o usuário faz as atividades do seu dia-a-dia. Um cenário pode ser informal como o apresentado no exemplo para um sistema de uma biblioteca:

Um aluno chega à biblioteca para procurar livros-texto dos cursos que está frequentando. Ele entra no sistema e seleciona os cursos, e obtém uma lista de todos os livros-texto e sua localização na biblioteca. Seleciona a opção de 'bibliografia

complementar', e uma nova lista de livros e artigos lhe é apresentada. Ele então manda imprimir todas as referências encontradas, (SOUZA, 2007, p26).

#### 2.4.1.2 Análise de Tarefas Utilizando Cenário

Segundo Souza (2007, p.25), as informações contidas através de narrações só podem ser realizadas se o processo for compreendido pelo analista e projetista de modo sistemático.

- **Geração do Cenário:** Os cenários devem ser escritos por um especialista, e os analistas podem fazer perguntas para motivá-los.
- **Elaboração da rede de proposições:** No cenário, a melhor forma é simplificar, expressando através de proposições e dividindo em termos a cada ação que o usuário faz.
- **Análise:** A partir de cada termo, pode-se determinar a descrição das tarefas e do usuário.
- **Questionamento sistemático:** São elaboradas novas questões de proposições anteriores, nas quais se destacam:

por quê...? – Revela consequência e causa;

como...? – Aprofunda-se em maiores detalhes ou especifica sub-tarefas;

o que é...? – Diz a respeito a distinguir atributos e hierarquia de objetos;

e então, isto é...? – Refere-se a uma verificação para obter a certeza da correção.

#### 2.4.1.3 Prototipação

O protótipo baseia-se numa visão evolutiva no desenvolvimento de software, buscando a aprovação de documentos gerados ao término na primeira etapa. O processo de desenvolvimento do protótipo pode ser classificado, como, (SOUZA, 2007, p.42):

- **Apresentação:** Nessa etapa, o cliente recebe uma percepção do sistema pelo designer, sendo que a interface seja adequado aos requisitos do usuário final.
- **Autêntico:** Tem a função de ilustrar a interface nos pontos específicos, em que ajuda a entender os problemas que nela é envolvido.

- **Funcional:** O grupo de desenvolvimento é ajudado a compreender questões que fazem parte da construção do projeto. E esse modelo de domínio não interessa ao usuário.
- **Sistema Piloto:** Esse método é usado para uma maior ilustração, pois contém um núcleo básico que é experimentado com o usuário.

A prototipação tem 3 modos de classificação, que são (SOUZA 2007, p.43):

- **Exploratória:** Esse modo é usado para melhorar o levantamento de requisitos, isso quer dizer que abrange uma variedade de opções de designer para que o usuário opte por um.
- **Experimental:** São resultados experimentais para que se tomem decisões de design e implementação pelo desenvolvedor.
- **Evolutiva:** Ela avalia o impacto de novas tecnologias, nesse caso, os designers devem trabalhar em cooperação com os usuários em um processo de reengenharia.

Uma grande vantagem do uso de protótipos é permitir que os designers experimentem idéias juntamente com o usuário.

#### 2.4.1.4 Entrevista

Segundo Cybis (2003), a entrevista propõe alcançar seu objetivo com perguntas e respostas, procurando identificar opiniões do cliente ou realçar o conhecimento sobre a tarefa do usuário.

Para realizar uma boa entrevista, é necessário que haja um clima de confiança com o usuário. É importante, também, preparar a entrevista com antecedência, para que não se perca o foco.

#### 2.4.1.5 Questionário

Segundo Souza (2007), o questionário é uma forma de coletar dados, para alcançar os recursos disponíveis tanto técnicos quanto físicos. Já, com o contexto da tarefa, o analista

utiliza o vocabulário de acordo com o conhecimento do usuário. É importante conhecer os níveis de usuários, diferenciar entre os experientes e os novatos, saber os recursos de aprendizagem em informática. Para uma elaboração de questionário, é necessário focar as funcionalidades do sistema, buscando conhecer o que causa maior impacto no usuário. E observar, também, que não se deve fazer duas perguntas em uma mesma questão.

## 2.4.2 Design da interface com usuário

Segundo Goud (1988 *apud* RATIONAL 2007), os usuários devem interagir com o design durante sua concepção. A terminologia e conceitos apresentados devem ser adequados às necessidades do usuário. Observa-se que os requisitos ou casos de usos detalhados são as análises dos usuários e ambientes em que se opera o sistema.

A interface de usuário deve atender duas dimensões: a experiência com o computador e a experiência no domínio da aplicação.

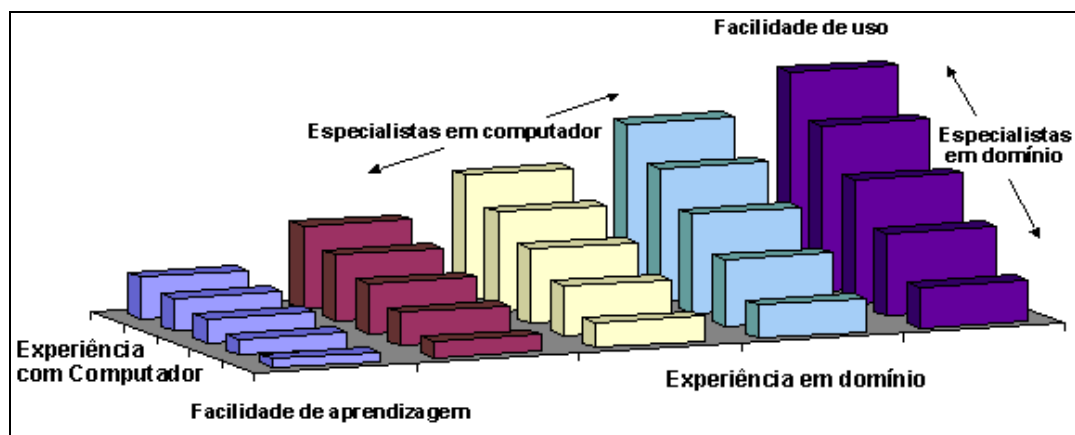


Figura 13 – Experiência com Computador X Domínio na Facilidade de Aprendizagem X Facilidade de Uso.  
Fonte: Rational (2007).

A experiência com o computador inclui a familiaridade em geral e, também, o sistema desenvolvido. Usuários com pouca experiência em computador ou no domínio do problema exigem um novo modelo de interface, ao contrário dos usuários que possuem um bom domínio.

Para que o usuário, sem experiência tornar-se experiente, é preciso que ele use freqüentemente o sistema, obtendo uma alta motivação e um baixo nível de complexidade durante a utilização.

Para se alcançar esse objetivo, devem ser observados os seguintes requisitos apontados no quadro abaixo:

	Baixa	Alta
<b>Experiência com computador</b>	Perguntas e respostas simples, preenchimento de formulário simples, estilo de interface da Web (com hyperlinks) ou de menus	Preenchimento de formulário complexo, estilo de interface da Web (com hyperlinks) ou de menus (pergunta e resposta ou preenchimento de formulário simples seria muito frustrante para usuários experientes)
<b>Experiência com domínio</b>	Terminologia e conceitos descrito de forma mais usual ou genérica.	Terminologia e conceitos específicos ao domínio
<b>Treinamento</b>	Foco na facilidade de aprendizagem (consistente, previsível, memorável)	Foco na facilidade de uso (direto, personalizável, não intrusivo)
<b>Freqüência de uso</b>	Fácil de aprender e lembrar, estilo de interface simples	Fácil de usar, vários atalhos e técnicas para permitir o controle pelo usuário
<b>Motivação</b>	Uso recompensado, poderoso sem parecer complexo	Sofisticado com muitas características avançadas e personalizáveis

Quadro 1 – Alguns fatores que afetam o design de interface do usuário.  
Fonte: Rational (2007).

#### 2.4.2.1 Contexto de Uso

De acordo com a norma ISO 13407 (International Standards Organization), o contexto de uso identifica a compreensão e especificação do contexto de uso. “... os atributos sugeridos



nessa identificação são: tarefas - papéis - usuários e ambientes” ISO (1999 *apud* RATIONAL 2007).

As tarefas são metas do uso do sistema a cumprir com suas obrigações. São recursos de segurança e atividades tecnológicas a serem padronizadas.

O papel será representado por cada usuário, isso significa que cada usuário representará o seu tipo de papel, melhor dizendo, a sua função.

Para cada usuário, obter seu conhecimento, sua experiência, sua habilidade para que se consiga chegar a meta de identificá-lo.

Os ambientes relacionam-se com a parte de hardware, software e materiais que serão utilizados.

Na figura abaixo, é possível identificar os relacionamentos entre os itens referenciados acima.

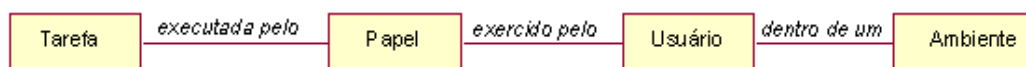


Figura 14 – Relacionamentos entre Atributos  
Fonte: Rational (2007).

No contexto citado acima, cada atributo tem uma importância significativa para o design centrado no usuário, mesmo sendo um sistema de pequeno porte, pode atingir um número grande de variações de contexto.

#### 2.4.2.2 Foco no Usuário

Segundo Gould (1988, *apud* RATIONAL 2007), para focar o design no usuário, é preciso conversar com esse usuário, observá-lo em seu ambiente de trabalho. Aprender como é feita a organização do seu trabalho. Para documentar de forma padronizada o contato, técnicas, como entrevistas e questionários, devem ser considerados. Uma opção interessante, também, é o uso de gravações de vídeo com o usuário trabalhando ou mesmo das entrevistas realizadas.

#### 2.4.2.3 Integração com o Design

Segundo Gould (1988, *apud* RATIONAL 2007), no desenvolvimento de design, técnicas de usabilidade devem ser aplicadas desde o começo do projeto. Argumenta, também, que “a usabilidade deve ser a responsabilidade de um grupo.”

Uma importante característica do design integrado é o modelo geral: o framework. O framework estabelece um modelo de domínio que assegura toda a tecnologia e conceitos que aparecem em uma interface. É necessário um grande cuidado com o modelo de domínio quanto à organização, o modelo deve ser de fácil identificação.

#### 2.4.2.4 Teste com Usuário

Segundo Rational (2007), uma forma de realizar testes é por meio de protótipos. Protótipos iniciais evoluem, desde a primeira etapa, novos desenhos, novos protótipos devem surgir aperfeiçoando o projeto. Para criar esses protótipos, é necessário que sejam criados cenários reais com o usuário, detalhar os procedimentos da tarefa e como realizar um trabalho para, então, aplicar esse conhecimento no design de interface.

### 2.4.3 Qualidades Ergonômicas para IHC

Segundo Mendes (2007), para apresentar uma boa página de web, é necessário apresentá-la de uma forma que ajude ao usuário final mostrá-la interessante e organizada. É importante observar que nem todos utilizam banda larga, por esse motivo, deve-se ter cuidado com as animações muito carregadas, e não exagerar em animações que não são úteis à página. Para Cybis (2003), “o sucesso de qualquer atividade de concepção ou de avaliação depende do emprego de critérios bem definidos.” Algumas qualidades importantes são:

- **Feedback**

Segundo Cybis (2003), feedback é o momento em que o usuário faz uma ação (ex.: clicar em um botão), e espera uma resposta. Para esse tipo de ação e reação, o tempo de

resposta tem que ser imediato, caso contrário, o usuário pode provocar uma série de erros no sistema, tornando a interação frustrante.

- **Legibilidade**

A legibilidade em um *web site* é fundamental, pois, para atingir o público em geral, é necessário clareza. Para Cybis (2003), “letras claras em um fundo escuro são mais fáceis de serem lidas, principalmente, por pessoas com problemas de visão. Para essas pessoas, o fundo brilhante pode ofuscar completamente as letras escuras.” A boa legibilidade promove uma boa leitura.

- **Acessibilidade**

Segundo Mendes (2007), há algumas metas que precisam ser vistas, uma delas é a acessibilidade. É necessário que a busca seja facilitada, pois muitos usuários estão à procura de informações específicas.

Temos que observar que os usuários tendem a procurar informações de diferentes formas, então, uma boa organização de informações de forma hierárquica ajudará muito.

- **Navegação/Orientação**

Segundo Cybis (2003), para uma navegação de boa qualidade, uma página deve ser bem orientada e dinâmica. Como os conteúdos de informações são distribuídos em links, é correto dizer que os links têm que fornecer uma descrição simples e objetiva.

- **Linguagem**

Segundo Mendes (2007), usar frases longas e termos técnicos podem ser enfadonho para o usuário, fazendo com que o mesmo deixe de ler ou se desinteresse. O correto é o uso de frases curtas e de estrutura simples.

Para apresentar uma boa página de web, é necessário organizar a informação de uma forma que ajude o usuário na interpretação, mostrá-la interessante ao usuário, porém organizada.

- **Adaptabilidade**

Segundo Cybis (2003), a adaptabilidade é proporcionar vários percursos de tarefas com o mesmo objetivo a serem executadas, deixando, assim, a opção de escolha pelo melhor caminho para o usuário.

- **Flexibilidade**

Segundo Cybis (2003), essa qualidade refere-se à interface personalizada que cumpre seu papel de exercer várias formas de realizar uma mesma tarefa, deixando, assim, ao usuário a escolha sobre a melhor maneira de realizá-lo.

- **Brevidade**

Segundo Cybis (2003), o ser humano possui uma capacidade de memória mínima. Então, quanto mais breves forem as entradas dos caminhos ou “links”, menor a possibilidade de erros serem cometidas pelo usuário. Caminhos muito extensos provocam esquecimentos.

O objetivo da brevidade é diminuir a carga de leitura, simplificar caminhos extensos, fazendo com que o usuário se sinta a vontade para entrar e sair, contanto que ele saiba onde prosseguir e se identificar no portal. O critério Brevidade se divide em duas qualidades elementares: Concisão e Ações Mínimas.

- **Gestão de Erros**

Segundo Cybis (2003), o bom gerenciamento da gestão de erros deve evitar as ocorrências de erros. Todos comentem erros e, quando isso ocorre, é necessário favorecer a correção, evitando mensagens inadequadas ou mesmo inúteis para a correção da tarefa.

- **Compatibilidade**

Segundo Cybis (2003), compatibilidade refere-se a apresentações de informações que são diretamente utilizadas. Um bom exemplo é o uso de animações. A animação deve ser utilizada em situações que são úteis dentro do contexto. No caso da *web*, o portal deve ser compatível com a missão, organização e metas da empresa.

- **Confiabilidade**

Segundo a Rational (2007), falhas são cometidas, freqüentemente, pelo usuário. Um erro pode ser fatal para o sistema. Para minimizar essa situação, seria interessante a possibilidade de recuperação.

### 3 MODELAGEM

Neste capítulo, serão descritos e apresentados os gráficos referentes à modelagem do sistema, fazendo uso da linguagem de notação UML, utilizando-se do método Iconix e do Design Centrado no Usuário (DCU).

#### 3.1 APLICAÇÃO DA METODOLOGIA

A partir do estudo dos métodos, definiu-se a estrutura de integração das metodologias ICONIX e Design Centrado no Usuário (DCU). As técnicas associadas devem se concretizar, na seguinte seqüência:

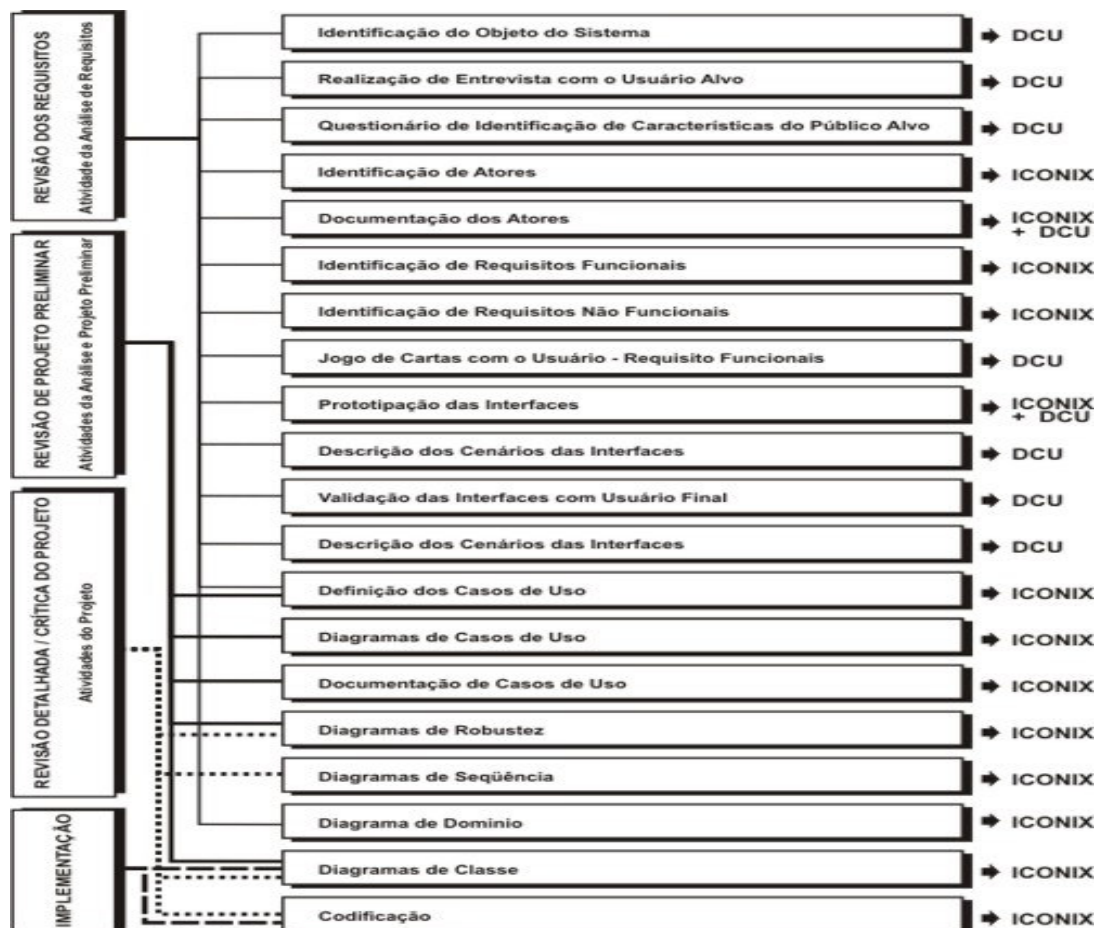


Figura 15 – Estrutura integração metodologia ICONIX e método DCU.

Fonte: Elaborado pelos autores (2007)

As cinco fases do ICONIX esta dividido em cinco marcos que são: modelo de domínio, modelo de caso de uso, análise robusta, diagrama de seqüência e diagrama de classe.

- O modelo de domínio se encontra na etapa de revisão de requisitos.
- O modelo de caso de uso se encontra na etapa de revisão de requisitos e revisão de projeto preliminar.
- A análise robusta se encontra na etapa de revisão de projeto preliminar e revisão detalhada / crítica do projeto.
- O diagrama de seqüência se encontra na etapa de revisão detalhada / crítica do projeto.
- O diagrama de classe se encontra na etapa de revisão detalhada / crítica do projeto e implementação.

### **3.1.1 Identificação do Objetivo do Sistema**

O objetivo do sistema é facilitar e garantir a armazenagem e recuperação dos dados dos animais domésticos atendidos e seus proprietários, evitando grande desembolso em casos emergenciais. Apoiar na gerência de um plano de vacinação adequada aos animais. Aos credenciados, a empresa oferecerá a possibilidade de consultar uma rede de clientes maior. Proporcionar, ainda, uma maior eficiência nos serviços veterinários pela integração das diferentes áreas de atuação e especialização dos profissionais dentro da medicina veterinária de pequenos animais.

### **3.1.2 Realização de Entrevistas com o Usuário Alvo (DCU)**

O objetivo inicial das entrevistas foi identificar os requisitos funcionais do sistema, a partir de necessidades e características do problema. As entrevistas foram realizadas com o *stakeholder*<sup>2</sup>: o Sr. Rosemberg e os desenvolvedores do projeto. O tempo das entrevistas foi de uma hora e quarenta e cinco minutos.

---

<sup>2</sup> Usuário final.

QUESTIONARIO APRESENTADO AO CLIENTE NA PRIMEIRA ENTREVISTA
Quais são os cadastros disponíveis no sistema?
Como funciona o plano de saúde? (classificação dos planos).
Qual o público alvo que pretende atingir?
Como serão realizadas as formas de pagamento?
Como os prestadores de serviço serão credenciados no sistema?
Quais são os relatórios essenciais para o sistema?
Como serão realizadas as consultas para o atendimento dos animais domésticos que estão ativos no plano?
O que o cliente necessita para realizar o cadastro do plano de saúde – Como será realizado?
Quais são os relatórios fundamentais para o sistema nessa primeira etapa?
O proprietário terá limite de atendimentos, cirurgias no plano de saúde?
Haverá venda de produtos para os animais domésticos?
Como funcionam os contratos dentro dos planos de saúde?
Quais são os níveis de permissão de acesso?

Quadro 2 – Questionamentos levantados durante entrevista com o cliente.  
 Fonte: Elaborado pelos autores (2007)

Após a realização da entrevista, os projetistas do sistema já conseguiram identificar, parcialmente, os requisitos funcionais, assim como entender o processo de funcionamento do plano de saúde. Na entrevista, identificaram-se os processos de atendimento, pagamento dos planos, os relatórios essenciais e as operações realizadas com os prestadores de serviço.

### 3.1.3 Entrevista de Identificação de Características do Público Alvo

Nessa etapa, realiza-se por meios de entrevistas a identificação dos usuários potenciais do sistema. Identificaram-se a faixa etária, o nível de conhecimento em informática, deficiências físicas ou mentais.

Os usuários identificados, na entrevista, foram: administrador, veterinário, atendente da clínica veterinária, proprietário, prestadores de serviço.

- **Administrador:** possui formação superior e noções de informática, conhece todos os processos do plano de saúde e não tem nenhum tipo de deficiência física ou mental.



- **Veterinário:** possuem formação superior, mas nem todos possuem noções de informática, não conhecem os processos do plano de saúde e não têm nenhum tipo de deficiência física ou mental.
- **Atendente da clínica veterinária:** Não possui formação superior, possui noções básicas de informática, não conhece os processos do plano de saúde, não tem nenhum tipo de deficiência física ou mental.
- **Proprietário:** possui todos os tipos de clientes podem que podem ou não ter deficiência física.
- **Prestadores de serviços:** possui formação superior e noções de informática, conhecem todos os processos do plano de saúde e não tem nenhum tipo de deficiência física ou mental.

### 3.1.4 Identificação de Atores (ICONIX)

“Um ator é um usuário externo do sistema – um objeto ou um conjunto de objetos que se comunica diretamente com o sistema, mas que não é parte dele, cada ator representa aqueles objetos que se comportam de uma maneira específica em relação ao sistema.” (BLAHA, 2006, p.137).

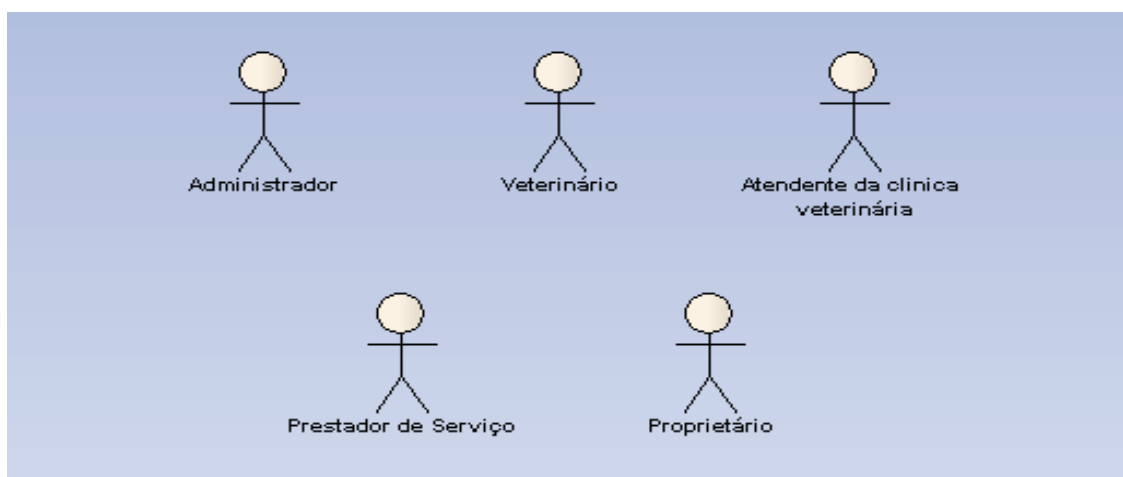


Figura 16 – Atores do sistema.

Fonte: Elaborado pelos autores (2007)

Nome	Descrição dos Atores
------	----------------------

Administrador	<p>É o usuário do sistema na área de gerência do plano de saúde.</p> <p>Esse usuário será dividido em 3 níveis de acesso que são:</p> <p><b>Administrador - Admin:</b> Que terá todo acesso do sistema de gerência do plano de saúde.</p> <p><b>Gerente – Admin:</b> Tem acesso restrito ao sistema, esse ator realiza as operações de gerência do plano de saúde como autorizações de inclusões de gerência do plano de saúde.</p> <p><b>Financeiro – Admin:</b> Cuida da parte financeira do plano de saúde, tem acesso a situações financeiras de proprietários, prestadora de serviço e pagamentos a ser realizado pela gerenciadora de plano de saúde.</p>
Prestador de Serviço	<p>Esse ator gerencia o módulo prestador de serviço que possui três níveis de acesso:</p> <p><b>Gerente – Prestserv:</b> O ator Gerência a realização de atendimentos, consultas e inclusão de serviços realizados aos animais na prestadora de serviço.</p> <p><b>Financeiro – Prestserv:</b> Possui acesso à parte financeira da prestadora de serviço em que o mesmo acessa aos relatórios financeiros da prestadora de serviço junto ao plano de saúde.</p> <p><b>Administrador - Prestserv:</b> Possui acesso irrestrito ao gerenciamento da prestadora de serviço, tem permissão para gerenciar os usuários da prestadora de serviço.</p>
Proprietário	Acesso somente a seu cadastro. Pode consultar a sua situação junto ao plano, a situação de cada um dos seus animais e adicionar informações do seu animal para futuras consultas.
Veterinário	Realiza as inclusões de serviços realizados no paciente.

Quadro 3 – Descrição dos atores do sistema.

Fonte: Elaborado pelos autores (2007)

### 3.1.5 Identificação de Requisitos Funcionais (ICONIX)

“Requisito funcional é um processo de desenvolvimento de software antes da solução de tecnologia a ser aplicada. Cada projeto deve ter suas fases de desenvolvimento adequadas

às necessidades de tratamento da informação, voltadas para o resultado do produto contratado (Zanlorenci et al., 2003).”

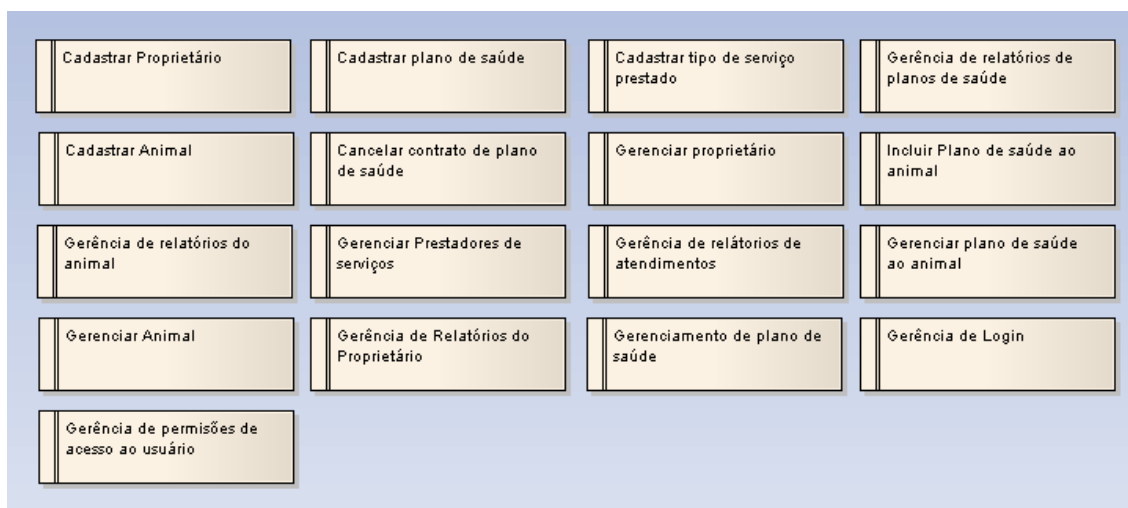


Figura 17 – Requisitos funcionais.

Fonte: Elaborado pelos autores (2007)

Nome	Descrição
Cadastrar Animal	O requisito é responsável pelo cadastro do animal que se beneficiará do plano de saúde animal.
Cadastrar Proprietário	Nesse requisito funcional, será cadastrado o proprietário do animal.
Cadastrar plano de saúde	Para esse requisito, serão cadastrados os planos de saúde existentes no sistema.
Cancelar contrato de plano de saúde	Nesse requisito, será definido o cancelamento de planos de saúde do proprietário do animal.
Gerência de Login	Nesse requisito, serão definidas as árvores de permissões de acesso ao sistema.
Gerência de Pagamentos do Proprietário	Nesse requisito, será realizado o Gerenciamento de pagamento dos planos de saúde dos proprietários.
Gerência de permissões de acesso ao usuário	Nesse requisito, serão definidos os tipos de permissões do sistema ao usuário.
Gerência de relatórios de	Nesse requisito, serão disponibilizados os relatórios de planos

planos de saúde	de saúde.
Gerência de relatórios do animal	Nesse requisito funcional, serão disponibilizados relatórios de controle animal.
Gerência de relatórios de atendimentos	Nesse requisito, serão disponibilizados os relatórios do grupo de atendimentos.
Gerência tipo de serviço prestado	Nesse requisito, serão disponibilizados os relatórios de atendimentos que a gerenciadora de planos de saúde cobre.
Gerenciar Animal	Nesse requisito, serão disponibilizadas as operações de inclusão, exclusão e alteração do animal.
Gerenciar Prestadores de serviços	Nesse requisito, será disponibilizado o gerenciamento de clínicas veterinárias para atendimento dos animais de seus proprietários que estão inclusos no plano de saúde.
Gerenciar plano de saúde	Nesse requisito, serão definidos os termos de inclusão alteração e exclusão de plano de saúde.
Gerenciar plano de saúde ao animal	Nesse requisito, será definida a inclusão exclusão e alteração do plano de saúde ao animal.
Gerenciar proprietário	Nesse requisito, poderá ser realizada a inclusão, exclusão e alteração do proprietário do animal.
Alocação do Plano de saúde ao animal	Nesse requisito funcional, o plano de saúde é alocado ao animal já que um proprietário pode ter vários planos de saúde sendo que cada animal possui um plano de saúde.

Quadro 4 – Descrição dos requisitos funcionais.  
 Fonte: Elaborado pelos autores (2007)

### 3.1.6 Identificação de Requisitos Não Funcionais (ICONIX)

“Requisitos não funcionais, ao contrário dos funcionais, não expressam nenhuma função (transformação) a ser implementada em um sistema de informações; eles expressam condições de comportamento e restrições que devem prevalecer, (CYSNEIROS, 2007)”.

RNF01 - Será desenvolvido e uma linguagem de programação Java (JSF) voltado para web	RNF06 - Telas com resoluções de 800 x 600	RNF12 - Pessoa não autorizada não pode ter acesso aos requisitos, seja para consulta e/ou escrita. Mesmo que uma pessoa possua autorização, pode ser que em alguns casos, requisitos ainda incipientes (não-públicos) não possam ser acompanhados por todos os
RNF02 - Estará sendo armazenado em um banco de dados MySQL	RNF07 - O sistema deverá apresentar "look and feel" considerado "inovador", "moderno", "efetivo" e, sobretudo, "profissional"	RNF14 - Pessoa, mesmo autorizada com poder de escrita, não necessariamente tem poder de decisão e, dessa forma, pode até editar requisitos, que ficarão pendentes. Deve-se ter a garantia de que após a aprovação de requisitos, não será permitida, em hipóteses
RNF03 - O sistema deverá apresentar tempo de resposta inferior a um (1) segundo para 90 % das funções que oferece.	RNF08 - O sistema deverá oferecer conjunto de recursos para visualização dos requisitos. Isto é, particularmente mais importante quanto mais complexa e volumosa é a especificação.	RNF10 - O sistema deverá oferecer interface de programação para clientes (software) que tenham interesse em manipular uma especificação de requisitos de software.
RNF04 - O sistema será feito apenas na língua portuguesa.	RNF09 - O sistema poderá ser utilizado em qualquer plataforma que ofereça suporte para Java SE 5.0 ou superior.	RNF05 - O sistema deverá oferecer mecanismos que contemplem usuários "avançados", isto é, oferecer recursos facilitadores (teclas de atalho, personalização de atividades,...) para o usuário com maior vivência na ferramenta.

Figura 18 – Requisitos não funcionais.

Fonte: Elaborado pelos autores (2007)

### 3.1.7 Jogo de Cartas com o Usuário – Requisitos Funcionais (DCU)

Segundo Ribeiro (1998), jogo de cartas consiste em uma técnica, na visão do usuário, sobre a organização das funcionalidades do sistema. Essa técnica é utilizada da seguinte maneira: são realizadas cartas com as tarefas do sistema, e o usuário seleciona aleatoriamente as tarefas mais importantes para o mesmo. Desta forma, conseguimos definir as tarefas de prioridades altas.

O jogo de cartas foi utilizado para identificação de menus, seqüencialização da navegação entre tarefas, validação de requisitos e validação de permissões de acesso e identificação dos requisitos funcionais.

A imagem, a seguir, mostra o cliente aplicando o jogo de cartas. Nesse momento, retratado pela foto, ele está começando a definir qual tarefa pertence a um determinado grupo de tarefas, dessa forma, organizando a seqüência de interação das tarefas.



Figura 19 – Aplicação do Jogo de cartas (1).

Fonte: Elaborado pelos autores (2007)

Após algumas etapas, o cliente já começou a definir o acesso para cada usuário. Ele definiu somente três atores do uso do sistema, informando que a atendente do plano de saúde e a atendente do prestador de serviço terão os mesmos acessos tais como a atendente poderá usufruir as operações de administrador, e a atendente do prestador de serviço poderá usufruir os acessos do prestserv.



Figura 20 – Aplicação do Jogo de cartas (2).

Fonte: Elaborado pelos autores (2007)

Com a aplicação do jogo de cartas, o cliente definiu a interface das telas. Os desenvolvedores questionaram sobre a necessidade de botões de acesso rápido, pois, no jogo de cartas, ele demonstrou desinteresse por tal mecanismo. Segundo o cliente, os menus estariam agrupados de forma coerente e fácil, sendo desnecessários, como mostra a figura, a seguir, sendo desnecessários botões de acesso rápido.



Figura 21 – Aplicação do Jogo de cartas (3).

Fonte: Elaborado pelos autores (2007)

Com a aplicação do jogo de cartas, pode-se definir como será realizada a construção de pacotes do sistema e ter um melhor entendimento de cada tarefa que será realizada no plano de saúde, como mostra a tabela a seguir.

Nível de acesso	Nome da tarefa
<b>PACOTE</b>	<b>PLANO DE SAÚDE</b>
<b>Sub-pacote</b>	<b>Relatórios.</b>
	**Nesse pacote será aberta uma nova Janela com os tipos de relatórios para o administrador e os seguintes relatórios serão mostrados**
1	• Prestador de Serviço.
2	• Serviços Prestados.
3	• Paciente/Proprietário.
4	• Proprietários Cadastrados.
5	• Tipos de Serviços.

6	<ul style="list-style-type: none"> <li>Plano de Saúde.</li> </ul>
<b>Sub-pacote</b>	<b>Relatórios Financeiros.</b>
	**Nesse pacote será aberta uma nova Janela com os tipos de relatórios para o administrador e os seguintes relatórios serão mostrados**
1	<ul style="list-style-type: none"> <li>Relatório de pagamentos a serem realizados.</li> </ul>
2	<ul style="list-style-type: none"> <li>Relatórios de reembolso por serviços prestados.</li> </ul>
3	<ul style="list-style-type: none"> <li>Relatório de situação financeira do Prestador de Serviço.</li> </ul>
4	<ul style="list-style-type: none"> <li>Relatório de balanço geral no período.</li> </ul>
5	<ul style="list-style-type: none"> <li>Relatório de proprietários inadimplentes.</li> </ul>
<b>Sub-pacote</b>	<b>Proprietário/Paciente.</b>
1	<ul style="list-style-type: none"> <li>Autorizar plano de saúde do paciente.</li> </ul>
2	<ul style="list-style-type: none"> <li>Ativar plano de saúde do paciente.</li> </ul>
3	<ul style="list-style-type: none"> <li>Desativar plano de saúde do paciente.</li> </ul>
<b>Sub-pacote</b>	<b>Prestador de serviço.</b>
1	<ul style="list-style-type: none"> <li>Autorizar prestador de serviço.</li> </ul>
2	<ul style="list-style-type: none"> <li>Alterar dados cadastrais.</li> </ul>
3	<ul style="list-style-type: none"> <li>Ativar prestador de serviço.</li> </ul>
4	<ul style="list-style-type: none"> <li>Desativar prestador de serviço.</li> <li>Incluir autorização de serviços.</li> </ul>
<b>Sub-pacote</b>	<b>Tipo de Serviço Prestado.</b>
1	<ul style="list-style-type: none"> <li>Cadastro de tipo de serviço prestado.</li> </ul>
2	<ul style="list-style-type: none"> <li>Alterar dados de tipo de serviço prestado.</li> </ul>
3	<ul style="list-style-type: none"> <li>Excluir tipo de serviço prestado.</li> </ul>
<b>Sub-pacote</b>	<b>Tipo de Plano de Saúde.</b>
1	<ul style="list-style-type: none"> <li>Incluir novo tipo de plano de saúde.</li> </ul>
2	<ul style="list-style-type: none"> <li>Ativar tipo de plano de saúde.</li> </ul>
3	<ul style="list-style-type: none"> <li>Desativar tipo de plano de saúde.</li> </ul>
<b>Sub-pacote</b>	<b>Contas de Usuário.</b>
1	<ul style="list-style-type: none"> <li>Cadastro de acesso de usuário.</li> </ul>
2	<ul style="list-style-type: none"> <li>Alterar permissão de acesso do usuário.</li> </ul>
3	<ul style="list-style-type: none"> <li>Ativar acesso de usuário.</li> </ul>
4	<ul style="list-style-type: none"> <li>Desativar acesso de usuário.</li> </ul>



<b>PACOTE</b>	<b>PRESTADOR DE SERVIÇOS</b>
<b>Sub-pacote 1</b>	<b>Relatórios de prestador de serviço.</b>
	<b>**Nesse pacote será aberta uma nova Janela com os tipos de relatórios para o prestador de serviço e os seguintes relatórios serão mostrados**</b>
1	<ul style="list-style-type: none"> <li>Relatório de serviços prestados junto ao animal.</li> </ul>
2	<ul style="list-style-type: none"> <li>Relatório dos comentários adicionados.</li> </ul>
3	<ul style="list-style-type: none"> <li>Relatório dos comentários adicionados por proprietário ao animal.</li> </ul>
<b>Sub-pacote 2</b>	<b>Gerência de realização de serviços.</b>
1	<ul style="list-style-type: none"> <li>Cadastrar serviço prestado ao animal.</li> </ul>
2	<ul style="list-style-type: none"> <li>Cadastrar comentários relacionados ao animal.</li> </ul>
<b>Sub-pacote 3</b>	<b>Relatórios de financeiro do prestador de serviço.</b>
	<b>**Nesse pacote será aberta uma nova Janela com os tipos de relatórios para o prestador de serviço, e os seguintes relatórios serão mostrados**</b>
1	<ul style="list-style-type: none"> <li>Relatório de reembolso pendente junto ao plano de saúde.</li> </ul>
2	<ul style="list-style-type: none"> <li>Relatório de serviços prestados realizado no período.</li> </ul>
3	<ul style="list-style-type: none"> <li>Relatório de reembolso recebido no período.</li> </ul>
4	<ul style="list-style-type: none"> <li>Relatório de valor de reembolso por cada tipo de serviço prestado.</li> </ul>
<b>Sub-pacote 4</b>	<b>Contas de usuário do prestador de serviço.</b>
1	<ul style="list-style-type: none"> <li>Cadastrar acesso de usuário do prestador de serviço.</li> </ul>
2	<ul style="list-style-type: none"> <li>Ativar acesso de usuário do prestador de serviço.</li> </ul>
3	<ul style="list-style-type: none"> <li>Desativar acesso de usuário do prestador de serviço.</li> </ul>
<b>Sub-pacote 5</b>	<b>Cadastro de prestador de serviço.</b>
1	<ul style="list-style-type: none"> <li>Cadastrar prestador de serviço.</li> </ul>
2	<ul style="list-style-type: none"> <li>Cadastrar serviços que são prestados pelo prestador de serviço.</li> </ul>
<b>PACOTE</b>	<b>PROPRIETÁRIO</b>
<b>Sub-pacote 1</b>	<b>Relatórios do proprietário.</b>
	<b>**Nesse pacote será aberta uma nova Janela com os tipos de relatórios para o proprietário, e os seguintes relatórios serão mostrados**</b>
1	<ul style="list-style-type: none"> <li>Relatório de situação junto ao plano.</li> </ul>
2	<ul style="list-style-type: none"> <li>Relatório de serviços prestados junto ao animal.</li> </ul>
3	<ul style="list-style-type: none"> <li>Relatório de comentários relacionados ao animal.</li> </ul>
<b>Sub-pacote 2</b>	<b>Gerência de Paciente.</b>

1	<ul style="list-style-type: none"> <li>• Cadastrar animal/plano de saúde.</li> </ul>
	*Após o cadastro, as seguintes tarefas poderão ser realizadas desse pacote.*
2	<ul style="list-style-type: none"> <li>• Emitir Boleto Bancário.</li> </ul>
	*Após liberação do administrador, as seguintes tarefas poderão ser realizadas.*
3	<ul style="list-style-type: none"> <li>• Cadastrar comentários relacionados ao animal.</li> </ul>
4	<ul style="list-style-type: none"> <li>• Alterar dados do animal.</li> </ul>
<b>Sub-pacote 3</b>	<b>Cadastro de Proprietário.</b>
1	<ul style="list-style-type: none"> <li>• Cadastrar proprietário.</li> </ul>
	*Após liberação do administrador, as seguintes tarefas poderão ser realizadas.*
2	<ul style="list-style-type: none"> <li>• Alterar dados do proprietário.</li> </ul>

Quadro 5 – Resultado final do jogo de cartas.

Fonte: Elaborado pelos autores (2007)

Após a realização do jogo de cartas, notou-se que o cliente havia alterado vários requisitos funcionais e adicionado outros que não haviam sido especificados. Notando-se isso, os desenvolvedores realizaram essa alteração e realizaram sua aplicação, como segue abaixo.

RF01 - Autorizar plano de saúde do animal	RF13 - Ativar acesso de usuário	RF25 - Cadastrar animal/plano de saúde.	RF37 - Relatório de pagamentos a serem realizados para Prestadores de serviço	RF49 - Relatório de reembolso recebido no período
RF02 - Autorizar prestador de serviço	RF14 - Desativar plano de saúde do animal	RF26 - Alterar dados do proprietário	RF38 - Relatório de prestador de serviço de situação financeira.	RF50 - Relatório de comentários relacionados ao animal
RF03 - Incluir novo tipo de plano de saúde	RF15 - Desativar prestador de serviço	RF27 - Relatório de serviços prestados junto ao animal	RF39 - Relatórios de reembolso por serviços prestados	RF51 - Imprimir boleto bancário
RF04 - Desativar tipo de plano de saúde	RF16 - Alterar dados do animal	RF28 - Relatório de situação junto ao plano	RF40 - Relatório de balanço geral no período	
RF05 - Ativar tipo de plano de saúde	RF17 - Cadastrar prestador de serviço	RF29 - Backup	RF41 - Desativar comentário adicionado por prestador de serviço ou proprietário	
RF06 - Cadastrar Tipo de serviço	RF18 - Cadastrar serviços prestados pelo prestador de serviço	RF30 - Relatório de Empresas Credenciadas	RF42 - Cadastrar acesso de usuário do prestador de serviço	
RF07 - Alteração de tipo de serviço prestado	RF19 - Cadastrar comentários relacionados ao animal	RF31 - Relatório de serviços prestados pelos prestadores de serviço	RF43 - Ativar acesso de usuário do prestador de serviço	
RF08 - Desativar tipo de serviço prestado	RF20 - Cadastro de serviço prestado ao animal	RF32 - Relatório de tipos de serviços.	RF44 - Desativar acesso de usuário do prestador de serviço	
RF09 - Ativar tipo de serviço prestado	RF21 - Alterar dados cadastrais do prestador de serviço	RF33 - Relatório de proprietários cadastrados no plano de saúde	RF45 - Relatório dos comentários adicionados por prestador de serviço ao animal	
RF10 - Cadastro de acesso de usuário	RF22 - Relatório de serviços prestados junto ao animal	RF34 - Relatório de Animal/Proprietário	RF46 - Relatório dos comentários adicionados por proprietário ao animal	
RF11 - Alterar permissão de acesso do usuário	RF23 - Relatório de serviços prestados realizado no período	RF35 - Relatório de plano de saúde	RF47 - Relatório de valor de reembolso por cada tipo de serviço prestado	
RF12 - Desativar acesso de usuário	RF24 - Cadastrar proprietário	RF36 - Relatório de proprietários inadimplentes	RF48 - Relatório de reembolso pendente junto ao plano de saúde	

Figura 22 – Definição dos novos requisitos funcionais, após a realização do jogo de cartas.

Fonte: Elaborado pelos autores (2007)

### 3.1.7.1 Pacote plano de saúde

Nome	Descrição
RF01 - Autorizar plano de saúde do animal.	Permitir que o animal passe a ser atendido pela rede credenciada do plano de saúde, dentro das normas contratuais.
RF02 - Autorizar prestador de serviço.	Permitir que o novo prestador de serviço faça parte da rede de atendimento do plano de saúde, dentro das normas contratuais definidas pelo plano de saúde.
RF03 - Incluir novo tipo de plano de saúde.	Tornar disponível no sistema e no contrato mais um pacote de serviços a ser vendido aos proprietários e que difere dos outros, em preço, tipo de serviços prestados, datas de carências, etc.
RF04 - Desativar tipo de plano de saúde.	Desativar do sistema o pacote de serviços do plano de saúde em que o proprietário não possa mais realizar nenhum contrato pelo plano que foi desativado.

RF05 - Ativar tipo de plano de saúde.	Ativar um pacote de plano de saúde em que o proprietário possa realizar uma inclusão de plano de saúde para o animal que será incluído no plano de saúde.
RF06 - Cadastrar Tipo de serviço.	Incluir um serviço que até então não era oferecido pela empresa em nenhum dos seus tipos de planos. Ex. translado, cirurgia de catarata, biópsia de pele.
RF07 - Alteração de tipo de serviço prestado.	Alterar parâmetros dentro de cada tipo de serviço, como por exemplo, valor a ser remetido ao prestador de serviço compensando o serviço prestado e data carência.
RF08 - Desativar tipo de serviço prestado.	Desativar do sistema um tipo de serviço que não é mais oferecido pela empresa em nenhum dos seus tipos de plano
RF09 - Ativar tipo de serviço prestado.	Ativar o sistema um tipo de serviço que era prestado para uso nos pacotes de planos de saúde e inclusão de serviços realizados pela prestadora de serviço.
RF10 - Cadastro de acesso de usuário.	Incluir no sistema um usuário e definir seus parâmetros de uso do sistema. Onde o mesmo pode ser: [Administrador – Admin   Gerente - Admin   Financeiro – Admin].
RF11 - Alterar permissão de acesso do usuário.	Definir que operações o usuário pode realizar no sistema.
RF12 - Desativar acesso de usuário.	Desativar o login e senha existente, não permitindo mais seu acesso ao sistema.
RF13 - Ativar acesso de usuário.	Ativar o login e senha existente, permitindo novamente seu acesso ao sistema.
RF14 - Desativar plano de saúde do animal.	Serão desativados os benefícios que o animal possui junto ao plano de saúde, isso poderá ocorrer automaticamente por falta de pagamento ou quando o administrador desativa o plano de saúde do animal.
RF15 - Desativar prestador de serviço.	Desativar do sistema o acesso do credenciado junto ao plano de saúde, não podendo mais realizar atendimentos pelo plano de saúde.

RF21 - Alterar dados cadastrais do prestador de serviço.	O administrador pode alterar dados da prestadora de serviço solicitados pela própria prestadora de serviço.
RF29 – Backup.	O sistema deverá gerar um backup dos dados a cada três dias salvando seus dados em arquivos.
RF30 - Relatório de Empresas Credenciadas.	Relatório das empresas cadastradas no plano de saúde onde mostra os dados das empresas credenciadas e a sua situação junto ao mesmo.
RF31 - Relatório de serviços prestados pelos prestadores de serviço.	Nesse relatório, serão apresentados ao Administrador do sistema os serviços realizados pelas credenciadas ou por somente uma credenciada.
RF32 - Relatório de tipos de serviços.	Em que se mostra a descrição de todos os tipos de serviço cadastrados no plano de saúde, junto com seu status.
RF33 - Relatório de proprietários cadastrados no plano de saúde.	Apresenta todos os proprietários cadastrados no plano de saúde.
RF34 - Relatório de Animal/Proprietário.	Mostra um relatório com seus animais e proprietários cadastrados no plano de saúde.
RF35 - Relatório de plano de saúde.	Nesse relatório, são mostrados os dados do pacote de planos de saúde juntamente com seus dados e status do mesmo.
RF36 - Relatório de proprietários inadimplentes.	Mostra os proprietários que tiveram seu plano de saúde cancelado por falta de pagamento ou somente por simples atraso.
RF37 - Relatório de pagamentos a serem realizados para Prestadores de serviço.	Esse relatório mostra os valores dos pagamentos que devem ser realizados pela prestadora de serviço.
RF38 - Relatório de prestador de serviço de situação financeira	Lista todos os prestadores de serviço e o valor a ser reembolsado pelo plano para cada um deles.

RF39 - Relatórios de reembolso por serviços prestados.	Lista os serviços prestados por cada prestador de serviço, em determinado período, o valor de cada serviço prestado e total de reembolso ao prestador de serviço.
RF40 - Relatório de balanço geral no período.	Lista todas as entradas financeiras no período, a partir de mensalidades de proprietários com animais cadastrados nos planos; e todas as saídas no período, na forma de reembolso aos prestadores de serviços; e os devidos valores parciais e totais.
RF41 - Desativar comentário adicionado por prestador de serviço ou proprietário.	Permite tornar indisponível, no relatório do animal determinado, comentário indevido adicionado por um prestador de serviço ou por um proprietário.

Quadro 6 – Descrição dos requisitos funcionais do pacote plano de saúde.

Fonte: Elaborado pelos autores (2007)

### 3.1.7.2 Pacote Prestador de Serviço.

Nome	Descrição
RF17 - Cadastrar prestador de serviço.	Cadastrar no sistema um novo credenciado, que poderá prestar serviços aos animais do plano e ser reembolsado pelo plano de saúde.
RF18 - Cadastrar serviços prestados pelo prestador de serviço.	Selecionar novo serviço prestado por determinado prestador de serviço dentro da cobertura do plano. Esse serviço, obrigatoriamente, deve ter sido cadastrado anteriormente nos “tipos de serviços”.
RF19 - Cadastrar comentários relacionados ao animal.	O proprietário ou prestadora de serviço poderá inserir comentários com relação a seu animal, que possam ser importantes ao médico veterinário em atendimentos futuros.
RF20 - Cadastro de serviço prestado ao animal.	Registro no sistema de prestação de determinado tipo de serviço por determinado prestador de serviço a determinado animal. Registrado pelo prestador de serviços. Fica armazenado no histórico do animal.
RF22 - Relatório de serviços prestados junto ao animal.	Relatório com todos os dados anteriores de inclusão de serviços prestados de um determinado animal, por prestadores de serviço credenciados no plano.

RF23 - Relatório de serviços prestados realizados no período.	Relatório com dados de inclusão de serviços prestados por esse prestador de serviços, a todos os animais do plano, dentro de determinado período. Incluindo dados financeiros.
RF42 - Cadastrar acesso de usuário do prestador de serviço.	Permissão que o prestador de serviço tem de criar um novo login de usuário no nível de Gerente - Prestserv, Financeiro - Prestserv ou Administrador - Prestserv da empresa com suas respectivas permissões de acesso.
RF43 - Ativar acesso de usuário do prestador de serviço.	Ativação do login de usuário no nível Gerente - Prestserv, Financeiro - Prestserv ou Administrador - Prestserv da empresa, que havia criado anteriormente.
RF44 - Desativar acesso de usuário do prestador de serviço.	Desativação de (tornar acesso indisponível) um login de usuário no nível de Gerente - Prestserv, Financeiro - Prestserv ou Administrador - Prestserv da empresa.
RF45 - Relatório dos comentários adicionados por prestador de serviço ao animal.	Lista informações adicionadas, anteriormente, por qualquer prestador de serviço que tenha atendido determinado animal pelo plano.
RF46 - Relatório dos comentários adicionados por proprietário ao animal.	Lista informações adicionadas, anteriormente, pelo proprietário do animal cadastrado no plano.
RF47 - Relatório de valor de reembolso por cada tipo de serviço prestado.	Enumera para o prestador de serviços o valor de reembolso de cada tipo de serviço oferecido pelo mesmo.
RF48 - Relatório de reembolso pendente junto ao plano de saúde.	Enumera para o prestador de serviços o valor parcial e total de serviços que ainda não lhe foi reembolsado pelo plano de saúde.
RF49 - Relatório de reembolso recebido no período	Enumera para o prestador de serviços o valor parcial e total de serviços que lhe foram reembolsados pelo plano de saúde em determinado período.

Quadro 7 – Descrição dos requisitos funcionais do pacote prestadora de serviço.

Fonte: Elaborado pelos autores (2007)

### 3.1.7.3 Pacote Proprietário

Nome	Descrição
RF24 - Cadastrar proprietário.	Cadastrar no sistema um novo proprietário cujos animais poderão ser, posteriormente, cadastrados e inscritos em um plano.
RF25 - Cadastrar animal/plano de saúde.	Cadastrar novo animal junto ao plano de saúde, que, após autorização, será atendido pelo plano na rede credenciada.
RF16 - Alterar dados do animal.	O proprietário pode realizar alterações do animal, tais como peso, altura, etc.
RF26 - Alterar dados do proprietário.	Alterar, no próprio cadastro do proprietário, dados de contato, endereço, etc.
RF27 - Relatório de serviços prestados junto ao animal.	Relatório com todos os dados anteriores de serviços prestados no animal por prestadores de serviço.
RF28 - Relatório de situação junto ao plano	Relatório com dados relacionados ao proprietário e seus animais, tratando de finanças, datas carências para os serviços prestados ao animal, pendências, etc.
RF50 - Relatório de comentários relacionados ao animal	Lista os comentários extras adicionados, anteriormente, pelo proprietário do animal, que possam interessar ao veterinário em atendimentos futuros.
RF51 - Imprimir boleto bancário	O proprietário pode realizar a impressão do boleto para que o mesmo possa efetuar o pagamento da fatura do mês do plano de saúde.

Quadro 8 – Descrição dos requisitos funcionais do pacote proprietário.

Fonte: Elaborado pelos autores (2007)

### 3.1.8 Prototipação das Interfaces (ICONIX+DCU)

O uso da prototipação apóia o desenvolvedor na validação da solução do projeto e no refinamento de requisitos. Os protótipos foram utilizados na identificação da estrutura da interface, assim como no seu conteúdo.



### 3.1.8.1 Estrutura da Tela

Foram estudados para a estrutura três modelos. O primeiro modelo apresenta a tela estruturada em cinco áreas, área de identificação da empresa, área de menus icônicos, área de menu textual, área de entrada de dados e área de mensagens e informações, conforme figura abaixo:

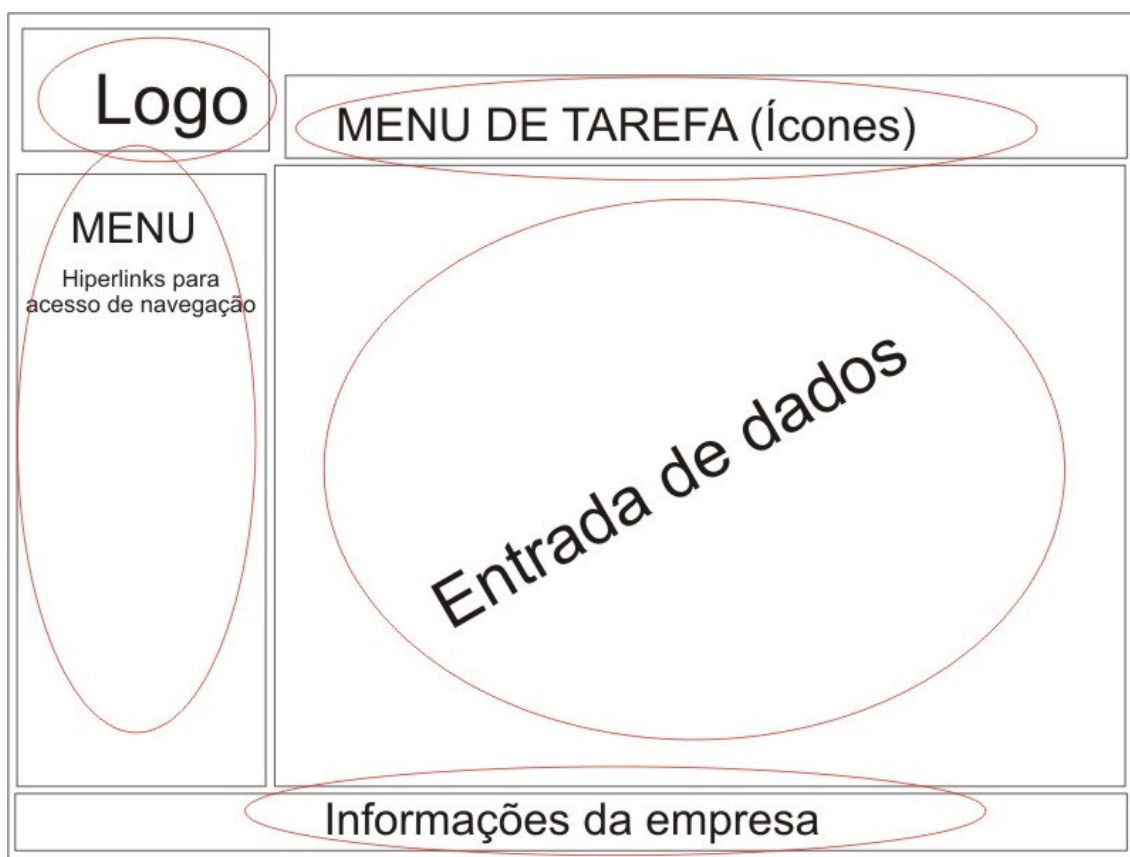


Figura 23 – Protótipo de tela de Identificação do Modelo 01.

Fonte: Elaborado pelos autores (2007)

No segundo modelo, definiram-se quatro áreas, a área de identificação da empresa, área de menu textual, área de entrada de dados e de botões, conforme figura abaixo:

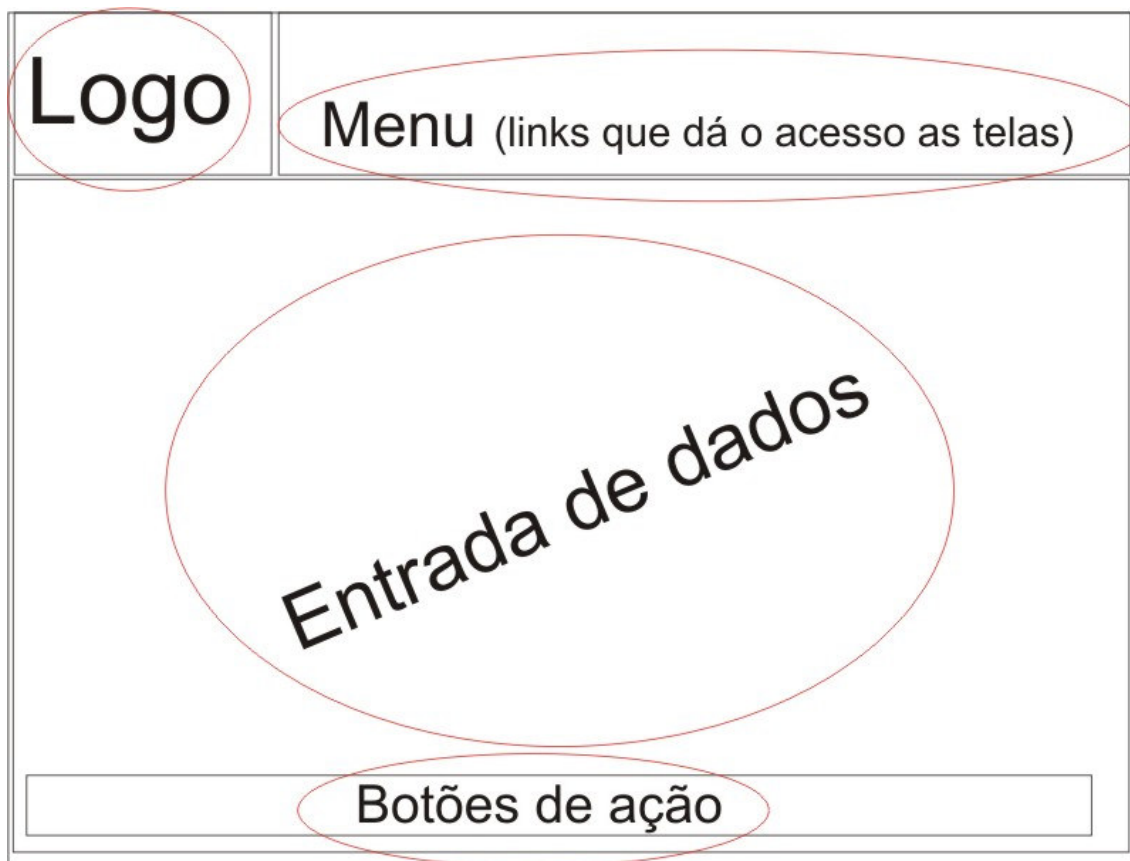


Figura 24 – Protótipo de tela de Identificação do Modelo 02.

Fonte: Elaborado pelos autores (2007)

No terceiro modelo, foram definidas cinco áreas, estruturadas em cinco áreas, área de identificação da empresa, área de menus hiperlinks, área de identificação de acesso, área de entrada de dados e área de botões, conforme figura abaixo:



Figura 25 – Protótipo de tela de Identificação do Modelo 03  
 Fonte: Elaborado pelos autores (2007)

No decorrer do desenvolvimento, houve a necessidade de se criar um quarto modelo com cinco áreas definidas, estruturadas em cinco áreas, área de menu de tarefa (os módulos), área de sub-menus de tarefas, área de filtro para pesquisa de dados, área de entrada e saída de dados e área de botões, conforme figura abaixo:



Figura 26 - Protótipo de tela de Identificação do Modelo 04  
Fonte: Elaborado pelos autores (2008)

#### 3.1.8.2 Definição Contextual da Tela

Na definição contextual, foram definidas fontes e cores usadas em rótulos, menus, cabeçalhos da tela.

No primeiro modelo, usou-se o requisito cadastrar os proprietários com melhor visualização de resolução de tela 800 x 600 pixels. A fonte de letra utilizada é Arial. Utilizou-se o fundo cinza com uma pigmentação de 50% da cor preta. O cadastro está subdividido por dados pessoais, dados de endereçamento e dados de contato.

**Logo**

**MENU DE TAREFA**

**Cadastros**

- \* Cadastro Proprietário
- \* Cadastro Animal

**Consultas**

- \* Vacinas
- \* Exames

**Ajuda**

- \* Funcionamento
- \* Contato

**Cadastro Proprietário**

**Dados Pessoais**

Nome:

RG:  CPF:

Estado Civil:  ☒ Solteiro(a) ☐ Casado(a) ☐ Separado(a) ☐ Viúvo(a) Data de Nascimento:

**Dados de Endereçamento**

Endereço:

Número:  Bairro:

Cidade:

Complemento:  UF:  ☒ SC - Santa Catarina ☐ PR - Paraná ☐ RJ - Rio de Janeiro ☐ SC - Santa Catarina ☐ SP - São Paulo ☐ TO - Tocantins CEP:

**Dados de Contato**

Fone Residencial:   -

Fone Comercial:   -

Celular:   -

E-mail:

**RODAPÉ - ENDEREÇO**

Figura 27 – Protótipo de tela Modelo 01.

Fonte: Elaborado pelos autores (2007).

No segundo modelo, utilizou-se o requisito cadastrar dados do animal. Essa tela apresenta o logo da empresa no topo central. Esse protótipo tem como objetivo cadastrar o animal do proprietário. É apresentado em uma tela de fundo branco.

**LOGO**

Gerenciamento de Cadastro Consultas Ajuda

Cadastro Proprietário

**Cadastro Animal**

Fechar

### Dados Animal

Nome do Animal:

RG do Animal:

Espécie: 

- Canino
- Felino
- Aves
- Roedores
- Répteis/Anfíbios
- Primatas
- Aquários/Peixes

Data de Nascimento:  /  dd/mm/aaaa

Características físicas:

Raça:

Pelagem: ☒ Curta ☐ Longa

Peso:  Altura:  Plano:

**Cadastrar Animal** **Limpar** **Cancelar**

Figura 28 – Protótipo de tela Modelo 02.

Fonte: Elaborado pelos autores (2007).

No terceiro modelo, utilizou-se o requisito cadastro prestador de serviço. Nessa tela, dividida em três *frames*, o logo da empresa está posicionado à esquerda no topo. Na lateral esquerda, temos o acesso a menu com os *links* de acesso para navegação. As fontes utilizadas são Arial para uma melhor visibilidade. Essa tela foi dimensionada para melhor visibilidade em 800 x 600 pixels. O acesso ao menu é extremamente simples, de forma legível e indicação de passos em cada tarefa executada. Sua cor é cinza e de fundo branco.

**LOGO**

**MENU**

- Cadastros**
  - \* Cadastro Proprietário
  - \* Cadastro Animal
  - \* Cadastro Prestadora de Serviço
- Consultas**
  - \* Vacinas
  - \* Exames
- Ajuda**
  - \* Funcionamento
  - \* Contato

**Cadastro do Prestador de Serviço**

**Dados do Prestador de Serviço**

Tipo de Cadastro:  ▼

Razão Social:

Nome Fantasia:

CPF/CNPJ:  Inscrição Estadual:

**Dados de Endereçamento**

Endereço:

Número:  Bairro:

Cidade:

Complemento:  UF:  ▼ CEP:  🔍

**Dados de Contato**

Fone Residencial: ()  -

Fone Comercial: ()  -

Fone Fax: ()  -

Celular: ()  -

E-mail:

Site:

Figura 29 – Protótipo de tela Modelo 03.  
 Fonte: Elaborado pelos autores (2007).

### 3.1.9 Validação das Interfaces com Usuário Final (DCU)

As estruturas e cores foram apresentadas ao usuário final. A estrutura da interface escolhida pelo usuário foi a do terceiro modelo, mas com modificações.

**LOGO**

**Cadastro do Animal**

**Dados Animal**

Nome:  Data de Nasc.:  [dd/mm/aaaa]

Espécie: ☐ Canino ☐ Felino ☒ Outra

Raça:  Selezione  
Akita  
Boxer  
Pitbull

Sexo: ☒ Macho ☐ Fêmea Peso:  Kg Altura:  0 cm a 10 cm

Porte:  Selezione  
Pequeno  
Médio  
Grande

Pelagem: ☒ Curta ☐ Longa

Cor:  Selezione  
Amarelo  
Cinza  
Preto

Características físicas particulares:

**Dados Pessoais**  
\* Alterar dados pessoais  
\* Alterar senha

**Dados do Animal**  
\* Cadastrar Paciente  
\* Alterar dados do Paciente

**Consultas**  
\* Serviços prestados ao animal  
\* Situação junto ao plano

**CADASTRAR** **LIMPAR** **CANCELAR**

Figura 30 – Protótipo de tela Aprovado Pelo Cliente.  
Fonte: Elaborado pelos autores (2007).

O fundo da tela foi uma das modificações. Optou-se pela cor de verde claro para o fundo por causa da logomarca da empresa em tonalidade verde. As fontes continuaram como Arial de cor preta. Sobre o tamanho da tela, foi decidida como no anterior 800 x 600 pixels. Os dados cadastrais do animal tiveram algumas alterações, são elas:

- Retirado: RG do animal.
- Incluído: Sexo do animal, Porte do animal e Cor do animal.
- Alterado: Peso: No caso do peso, foi retirado a caixa de escolha (ComboBox) e colocado a caixa de texto para o usuário informar.

Observou-se, no entanto, a “vontade” do usuário final em possuir uma tela mais “moderna”. Foi, então, proposto um novo *layout* de cores, baseado no estudo anterior.



### 3.1.9.1 Segunda Validação com o Usuário Final

A tela apresentada, abaixo, foi apresentada ao usuário final para avaliação:

O protótipo da tela apresenta uma interface com uma barra superior contendo quatro botões de menu: 'PLANO DE SAÚDE' (com ícone de cruz verde), 'PRESTADOR DE SERVIÇOS' (com ícone de animais), 'PROPRIETÁRIO PACIENTE' (com ícone de pessoas) e 'FECHAR' (com ícone de X vermelho). Abaixo, há uma barra com três sub-menus: 'TIPO DE SERVIÇO' (com ícone de pessoa), 'TIPO DE PLANO' (com ícone de lâmpada) e 'FECHAR SUB-MENU' (com ícone de X vermelho).

Na área principal, há campos para 'Código' e 'Nome', seguidos por um checkbox 'Ativos'. À direita, há botões 'Limpar' e 'Filtrar'. Abaixo, há uma tabela com as seguintes colunas: 'Id', 'Nome', 'Descrição', 'Carência' e 'Valor'. A tabela possui 10 linhas vazias para dados. Na base da tela, há botões 'Alterar' e 'Cadastrar'.

Id	Nome	Descrição	Carência	Valor

Figura 31 – Novo Protótipo de tela Aprovado Pelo Cliente.  
Fonte: Elaborado pelos autores (2008).

O objetivo dessa nova tela é de ser um módulo simples e eficiente para efetuar as operações necessárias. Os menus são módulos que estão divididos na parte superior do sistema, logo após, vêm os sub-menus que executarão as telas presenciais. A cor verde escura foi escolhida pelo usuário final. As tarefas foram selecionadas de acordo com o jogo de cartas.

A tela foi modificada por inteiro para que houvesse comunicação com os dados tanto de entrada como de saída. Os botões de ação ficaram na parte inferior à direita.

### 3.1.10 Definição dos Casos de Uso (ICONIX)

De acordo com Blaha et al. (2006, p.137), caso de uso é uma funcionalidade que faz parte do sistema que pode fornecer interação com os atores.

Na definição dos casos de uso, foi definido que o sistema será dividido em dois pacotes, o pacote plano de saúde, prestadora de serviço e proprietário que será mostrado na figura a seguir.

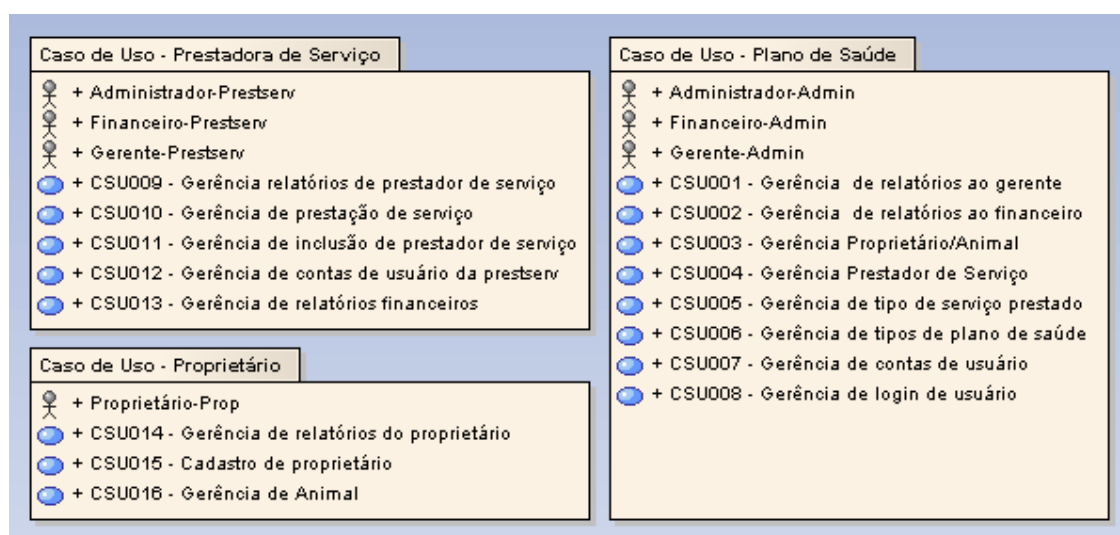


Figura 32 – Definição dos pacotes de Casos de Uso.

Fonte: Elaborado pelos autores (2007).

### 3.1.11 Casos de Uso – Pacote Plano de Saúde (ICONIX)

A seguir, é apresentado o diagrama de caso de uso do pacote Plano de saúde. Nesses, os atores são: Administrador (Administrador - admin, Gerente - admin e Financeiro – admin).

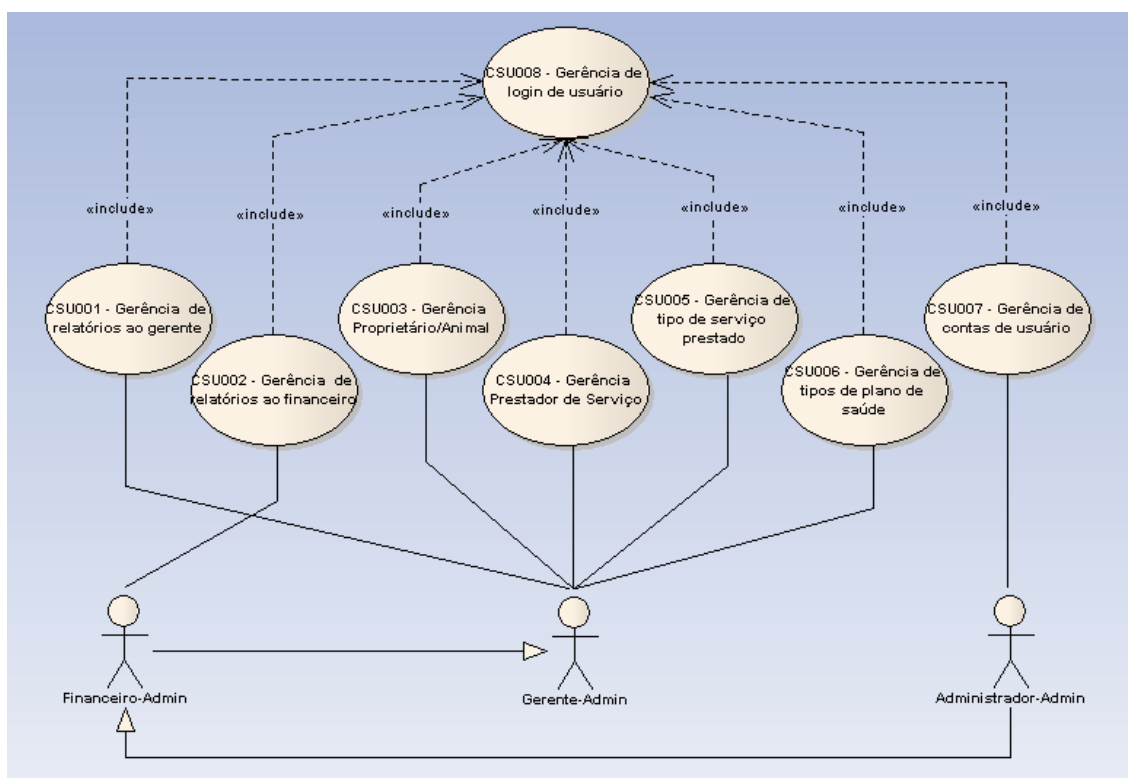


Figura 33 – Diagrama de Casos de Uso – Pacote plano de saúde.

Fonte: Elaborado pelos autores (2007)

A seguir, segue a descrição dos casos de uso;

Nome	Descrição dos casos de uso
CSU001 – Gerência de relatórios ao gerente.	Caso de uso que mostra os relatórios do plano de saúde.
CSU002 – Gerência de relatórios ao financeiro.	Nesse caso de uso, serão descritos os vários tipos de relatórios financeiros do plano de saúde.
CSU003 – Gerência Proprietário/Animal.	Nesse caso de uso, será realizada a gerência dos proprietários e animais que estão cadastrados no plano de saúde.
CSU004 – Gerência Prestador de Serviço.	Nesse caso de uso, será realizada a gerência da prestadora de serviços.
CSU005 - Gerência de tipo de serviço prestado.	Nesse caso de uso, será realizada a gerência dos tipos de serviços prestados pelo plano de saúde.

CSU006 - Gerência de tipos de plano de saúde.	Nesse caso de uso, será realizada a gerência dos tipos de planos de saúde, para a realização de inclusão de animais nos mesmos.
CSU007 - Gerência de contas de usuário.	Nesse caso de uso, será realizada a gerência de contas dos usuários do plano de saúde.
CSU008 - Gerência de login de usuário.	Nesse caso de uso, será realizada a gerência de acesso de usuários do sistema.

Quadro 9 – Descrição dos casos de uso do pacote plano de saúde.

Fonte: Elaborado pelos autores (2007).

### 3.1.12 Casos de Uso – Pacote Prestador de Serviço (ICONIX)

No pacote prestador de serviço, fazem parte os casos em que os atores Administrador - Prestserv, Gerente - Prestserv e Financeiro - Prestserv da Prestadora de serviço poderão gerenciar o sistema e, a seguir, serão mostrados os casos de uso que fazem parte desse pacote:

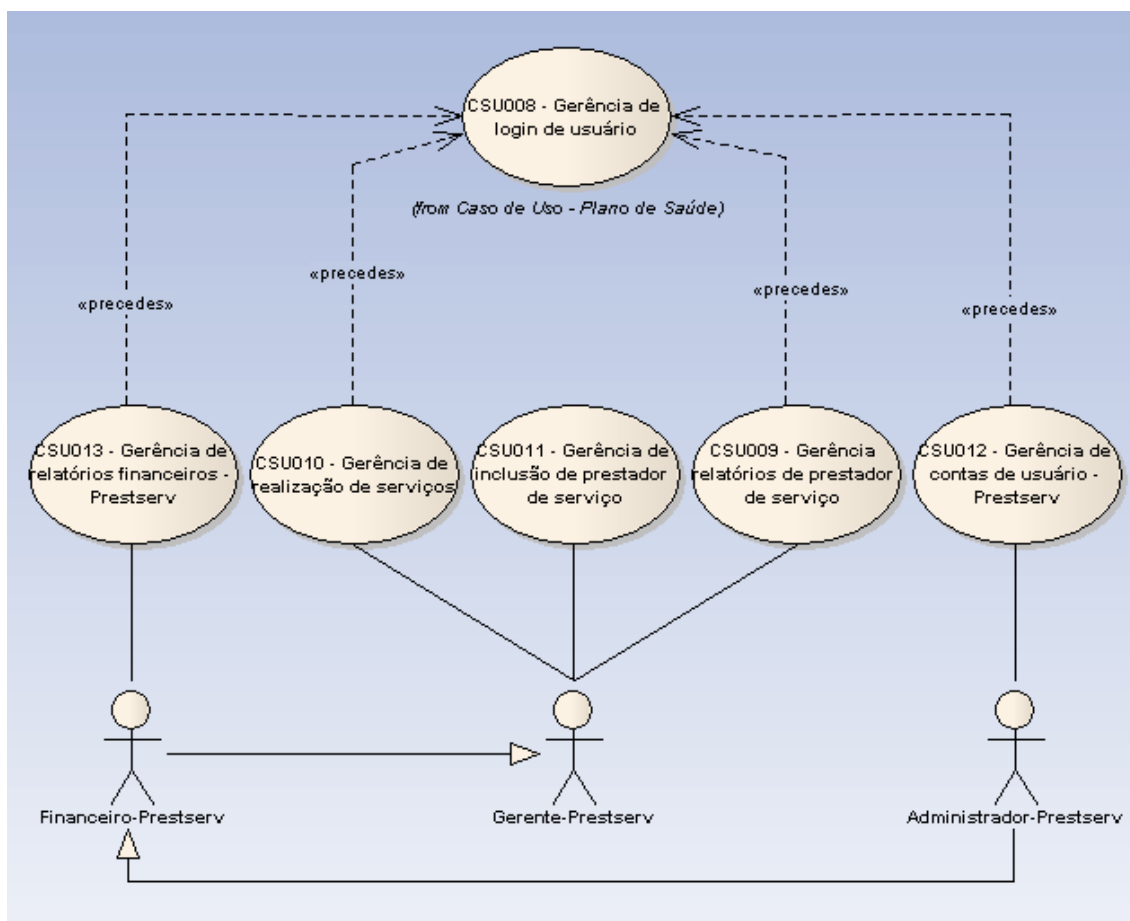


Figura 34 – Diagrama de Casos de Uso – Pacote prestadora de serviço.

Fonte: Elaborado pelos autores (2007)

Nesse pacote de caso de uso, serão definidas as tarefas do administrador:

<b>Nome</b>	<b>Descrição dos casos de uso</b>
CSU009 - Gerência relatórios de prestador de serviços	Esse pacote se refere aos relatórios que serão emitidos pelo prestador de serviço com o intuito para a gerência do plano de saúde na prestadora de serviço.
CSU010 - Gerência de prestação de serviço	Nesse pacote, será realizado a inclusão de serviços realizados nos animais pela prestadora de serviço.
CSU011 - Gerência de inclusão de prestador de serviço.	Esse pacote refere-se à gerência de inclusão do prestador de serviço no plano de saúde, para a realização de atendimentos nos animais domésticos.
CSU012 - Gerência de contas de usuário da Prestserv.	Nesse pacote, será realizada a gerência de contas de usuário da prestadora de serviço.
CSU013 - Gerência de relatórios financeiros	Nesse caso de uso, será realizada a gerência dos relatórios da parte financeira da prestadora de serviço.

Quadro 10 – Descrição dos casos de uso do pacote prestadora de serviço.

Fonte: Elaborado pelos autores (2007).

### 3.1.13 Casos de Uso – Pacote Proprietário (ICONIX)

Os casos de uso do pacote proprietário em que o ator principal é definido a seguir:

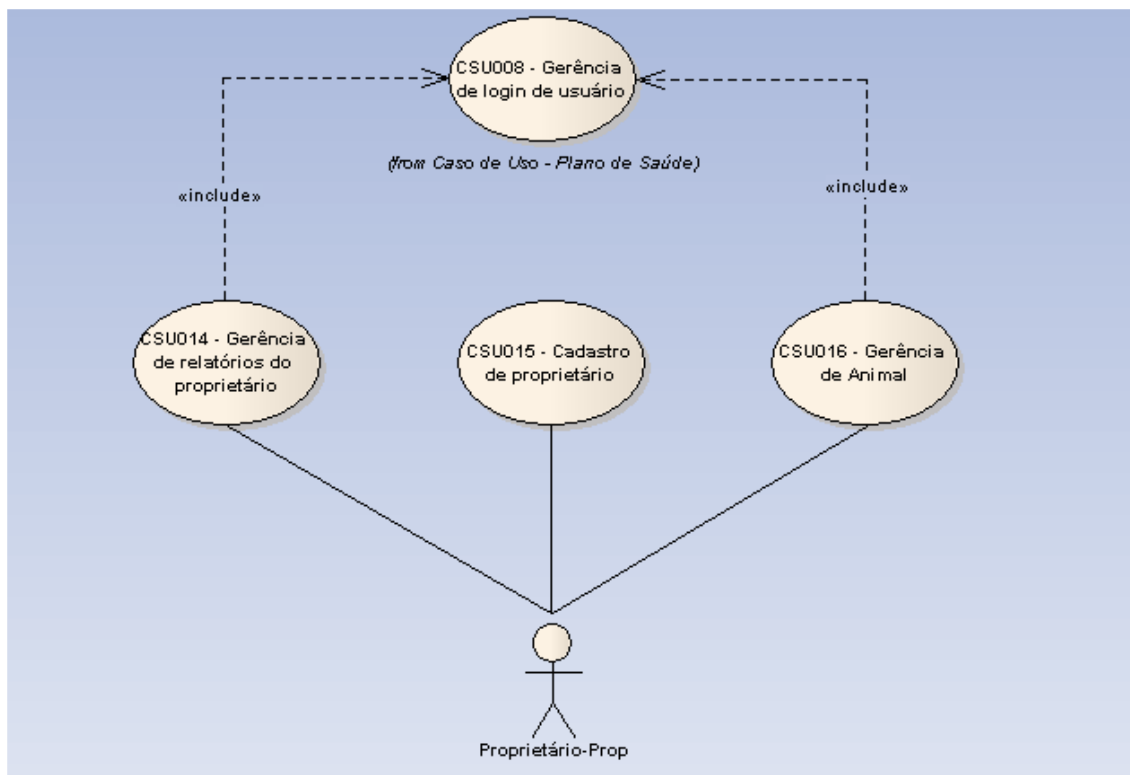


Figura 35 – Diagrama de Casos de Uso – Pacote proprietário.

Fonte: Elaborado pelos autores (2007).

Nome	Descrição dos casos de uso
CSU014 - Gerência de relatórios do proprietário.	Nesse caso de uso, serão documentados os relatórios de uso do proprietário.
CSU015 - Cadastro de proprietário.	Nesse caso de uso, o proprietário irá realizar a inclusão do mesmo no sistema para que ele possa incluir os animais de estimação no plano de saúde.
CSU016 - Gerência de Animal.	Nesse caso de uso, o proprietário irá gerenciar dados do animal e emissão de boletos bancários.

Quadro 11 – Descrição dos casos de uso do pacote proprietário.

Fonte: Elaborado pelos autores (2007).

### 3.1.14 Casos de Uso – Pacote Plano de Saúde (ICONIX)

A documentação dos casos de uso do pacote plano de saúde é apresentado a seguir:

<b>Gerência de relatórios ao gerente</b>	
<b>Identificador:</b>	CSU001
<b>Descrição:</b>	Nesse serão mostrados os relatórios do plano de saúde na parte de gerenciamento do plano de saúde.
<b>Ator Primário:</b>	Gerente – Admin / Administrador – Admin
<b>Ator Secundário:</b>	- -
<b>Pré-condição:</b>	O Usuário com permissão de Gerente – Admin ou Administrador – Admin estar logado no Sistema.
<b>Fluxo de Tarefas:</b>	1 - Relatório de prestador de serviço. 2 - Relatórios de Animal/Proprietário 3 - Relatório de proprietários cadastrados no plano de saúde. 4 - Relatório de tipos de serviços. 5 - Relatórios de plano de saúde
<b>Pós-condição:</b>	Todos os relatórios podem ser gerados com condições de serem exportados para PDF.

Quadro 12 – Documentação do Caso de Uso - Gerência de relatórios ao gerente.

Fonte: Elaborado pelos autores (2007).

<b>Gerência de relatórios ao financeiro</b>	
<b>Identificador:</b>	CSU002
<b>Descrição:</b>	Nesse caso de uso, serão descritos os vários tipos de relatórios da parte financeira do plano de saúde.
<b>Ator Primário:</b>	Financeiro – Admin / Administrador – Admin
<b>Ator Secundário:</b>	- - -
<b>Pré-condição:</b>	O Usuário com permissão de Financeiro – Admin ou Administrador – Admin estar logado no Sistema.
<b>Fluxo de Tarefas:</b>	1 - Relatório de pagamentos a serem realizados para prestadores de serviço 2 - Relatórios de reembolso por serviços prestados. 3 - Relatório de prestador de serviço de situação financeira.

	4 - Relatório de balanço geral no período. 5 - Relatórios de proprietários inadimplentes.
<b>Pós-condição:</b>	Todos os relatórios podem ser gerados com condições de serem exportados para PDF.

Quadro 13 – Documentação do Caso de Uso - Gerência de relatórios ao financeiro.

Fonte: Elaborado pelos autores (2007)

<b>Gerência Proprietário/Animal</b>	
<b>Identificador:</b>	CSU003
<b>Descrição:</b>	Nesse caso de uso, será realizada a gerencia dos proprietários e animais que estão cadastrados no plano de saúde.
<b>Ator Primário:</b>	Gerente – Admin / Administrador – Admin
<b>Ator Secundário:</b>	- -
<b>Pré-condição:</b>	O Usuário com permissão de Gerente – Admin ou Administrador – Admin estar logado no Sistema.
<b>Fluxo de Tarefas:</b>	1 - Autorizar plano de saúde do animal 2 - Ativar plano de saúde do animal 3 - Desativar plano de saúde do animal 4 - Desativar comentário adicionado pelo prestador de serviço ou proprietário.
<b>Pós-condição:</b>	1 – Plano de saúde autorizado, após, realiza impressão de contrato. 2 - Plano de saúde é reativado novamente para que o proprietário possa realizar os atendimentos na prestadora de serviço pelo plano de saúde. 3 - O plano de saúde do animal é bloqueado em que o mesmo não poderá realizar consultas junto ao plano de saúde. 4 – Em que o mesmo é desativado da tabela os comentários realizados pelo proprietário ou prestador de serviço, para que não apareça mais nas consultas realizadas pelo prestador ou proprietário.

Quadro 14 – Documentação do Caso de Uso - Gerência Proprietário/Animal.

Fonte: Elaborado pelos autores (2007).



<b>Gerência Prestador de Serviço</b>	
<b>Identificador:</b>	CSU004
<b>Descrição:</b>	Nesse caso de uso, será realizada a gerência da prestadora de serviço.
<b>Ator Primário:</b>	Gerente – Admin / Administrador – Admin
<b>Ator Secundário:</b>	- -
<b>Pré-condição:</b>	O Usuário com permissão de Gerente – Admin ou Administrador – Admin estar logado no Sistema.
<b>Fluxo de Tarefas:</b>	1 - Autorizar prestador de serviço 2 - Alterar dados cadastrais do prestador de serviço 3 - Ativar prestador de serviço 4 - Desativar prestador de serviço
<b>Pós-condição:</b>	1 – Após a realização dessa tarefa, o administrador poderá autorizar o prestador de serviço ou não, caso o mesmo autorize o prestador de serviço, já, poderá realizar prestação de serviço pelo plano de saúde. 2 – Na realização dessa tarefa, o administrador ou gerente poderá alterar os dados cadastrais do prestador de serviço. 3 – Na realização dessa tarefa, o mesmo poderá ativar o prestador de serviço para que o mesmo possa realizar prestação de pelo plano de saúde. 4 - Na realização dessa tarefa, o mesmo poderá desativar o prestador de serviço para que o mesmo não possa realizar prestação de pelo plano de saúde.

Quadro 15 – Documentação do Caso de Uso - Gerencia Prestador de Serviço.

Fonte: Elaborado pelos autores (2007).

<b>Gerência de tipo de serviço prestado</b>	
<b>Identificador:</b>	CSU005
<b>Descrição:</b>	Nesse caso de uso, será realizada a gerência dos tipos de serviços prestados do plano de saúde.
<b>Ator Primário:</b>	Gerente – Admin / Administrador – Admin
<b>Ator Secundário:</b>	- -

<b>Pré-condição:</b>	O Usuário com permissão de Gerente – Admin ou Administrador – Admin estar logado no Sistema.
<b>Fluxo de Tarefas:</b>	1 - Cadastro de tipo de serviço prestado 2 - Gerência tipo serviço prestado 2.1 - Alterar dados de tipo de serviço prestado 2.2 - Ativar tipo de serviço prestado 2.3 - Desativar tipo de serviço prestado
<b>Pós-condição:</b>	1 – Nessa tarefa, será realizado o cadastro de tipos de serviço prestado em que o mesmo, após o cadastro, estará em estado de desativado, assim, não podendo ser incluído no serviço prestado, quando o prestador de serviço for incluir o serviço prestado no momento do cadastro ou alteração. 2.1 – Nessa tarefa, o gerente ou administrador poderá realizar as alterações dos dados dos tipos de serviço prestado. 2.2 – Nessa tarefa, após a execução do mesmo, o prestador de serviço poderá incluir nos serviços prestados pelo prestador de serviço o tipo de serviço prestado. 2.3 – Nessa tarefa, após a realização do mesmo, o tipo de serviço prestado não poderá mais ser incluído nos serviços prestados pelo prestador de serviço.

Quadro 16 – Documentação do Caso de Uso - Gerência de tipo de serviço prestado.

Fonte: Elaborado pelos autores (2007).

<b>Gerência de tipos de plano de saúde</b>	
<b>Identificador:</b>	CSU006
<b>Descrição:</b>	Nesse caso de uso, será realizada a gerência dos tipos de planos de saúde, para a realização de inclusão de animais nos mesmos.
<b>Ator Primário:</b>	Gerente – Admin / Administrador – Admin
<b>Ator Secundário:</b>	- -
<b>Pré-condição:</b>	O Usuário com permissão de Gerente – Admin ou Administrador – Admin estar logado no Sistema.
<b>Fluxo de Tarefas:</b>	1 - Incluir novo tipo de plano de saúde 2 - Ativar tipo de plano de saúde

	3 - Desativar tipo de plano de saúde
<b>Pós-condição:</b>	<p>1 – Nessa tarefa, o usuário poderá incluir um novo plano de saúde para que o proprietário possa incluir o seu animal no plano de saúde, no cadastro de plano de saúde, ele será cadastrado em estado de desativado.</p> <p>2 – Nessa tarefa, o usuário poderá ativar o plano de saúde que está em estado de desativado, após a ativação o mesmo já estará liberado para o cadastro de animal.</p> <p>3 - Nessa tarefa, o usuário poderá desativar o plano de saúde para o cadastro de animal no plano de saúde. Em que o mesmo, após a realização dessa tarefa, o proprietário não poderá mais cadastrar o animal nesse plano de saúde.</p>

Quadro 17 – Documentação do Caso de Uso - Gerência de tipos de plano de saúde.

Fonte: Elaborado pelos autores (2007).

<b>Gerência de contas de usuário</b>	
<b>Identificador:</b>	CSU007
<b>Descrição:</b>	Nesse caso de uso, será realizada a gerência de contas dos usuários do plano de saúde.
<b>Ator Primário:</b>	Administrador – Admin
<b>Ator Secundário:</b>	- -
<b>Pré-condição:</b>	O Usuário com permissão de Administrador – Admin estar logado no Sistema.
<b>Fluxo Tarefas:</b>	<p>1 - Cadastro de acesso de usuário</p> <p>2 - Alterar permissão de acesso do usuário</p> <p>3 - Ativar acesso de usuário</p> <p>4 - Desativar acesso de usuário</p>
<b>Pós-condição:</b>	<p>1 – Nessa tarefa, são definidas as permissões de acesso do usuário em que o mesmo poderá ser definido em detalhes em nível de tarefas ou mesmo em nível de atores, após a realização do mesmo usuário cadastrado se encontre em estado de desativado.</p> <p>2 – Nessa tarefa, o administrador irá alterar as permissões de acesso do usuário, podendo ser definida por atores ou em nível</p>

	<p>de tarefas.</p> <p>3 – Nessa tarefa, o administrador poderá ativar a permissão de acesso do usuário, podendo, assim, o mesmo, após a execução do mesmo~, a utilização do sistema.</p> <p>4 – Nessa tarefa, o administrador poderá desativar a permissão de acesso do usuário, podendo, assim, o mesmo, após a execução do mesmo em que o usuário não podendo acessar mais o sistema para gerenciamento.</p>
--	--

Quadro 18 – Documentação do Caso de Uso - Gerência de contas de usuário.

Fonte: Elaborado pelos autores (2007).

Gerência de login de usuário	
<b>Identificador:</b>	CSU008
<b>Descrição:</b>	Nesse caso de uso, será realizada a gerência de acesso de usuários do sistema.
<b>Ator Primário:</b>	Gerente – Admin / Administrador – Admin / Financeiro – Admin
<b>Ator Secundário:</b>	- -
<b>Pré-condição:</b>	O Usuário com permissão de Administrador – Admin estar logado no Sistema.
<b>Fluxo de tarefas:</b>	<p>1 – Carrega árvore de Permissão de acesso</p> <p>2 – Verifica permissão de acesso</p>
<b>Pós-condição:</b>	<p>1 – Nessa tarefa, quando o usuário se loga no sistema, o sistema carrega um objeto com as definições de permissão de acesso que fica na memória RAM para verificações de acesso quando o usuário for tentar acessar algo.</p> <p>2 – Nessa tarefa, o sistema verifica se o usuário tem permissão de acesso, caso o usuário tenha permissão retorna um <i>boolean</i> de verdadeiro caso contrário falso.</p>

Quadro 19 – Documentação do Caso de Uso - Gerência de login de usuário.

Fonte: Elaborado pelos autores (2007).

### 3.3.15 Casos de Uso – Pacote Prestador de Serviço (ICONIX)

Neste capítulo será mostrado a documentação dos casos de uso do pacote administrador onde serão mostradas as tarefas de cada caso de uso;

<b>Gerência relatórios de prestador de serviço</b>	
<b>Identificador:</b>	CSU009
<b>Descrição:</b>	Esse pacote se refere aos relatórios que serão emitidos pelo prestador de serviço com o intuito para a gerência do plano de saúde na prestadora de serviço.
<b>Ator Primário:</b>	Gerente - Prestserv / Financeiro – Prestserv / Administrador – Prestserv
<b>Ator Secundário:</b>	- -
<b>Pré-condição:</b>	O Usuário com permissão de Gerente – Prestserv ou Financeiro – Prestserv estar logado no Sistema.
<b>Fluxo de tarefas:</b>	1 - Relatório de serviços prestados junto ao animal 2 - Relatório dos comentários adicionados por prestador de serviço ao animal. 3 - Relatório dos comentários adicionados por proprietário ao animal.
<b>Pós-condição:</b>	1 – Nessa tarefa, são mostrados dados dos serviços que foram realizados no animal pelos prestadores de serviço, data medicações dadas entre outras. 2 – Nessa tarefa, são mostrados os comentários adicionados pelos prestadores de serviço sobre o animal durante a realização das consultas. 3 – Nessa tarefa, são mostrados os comentários adicionados pelos proprietários sobre seu animal para que os veterinários possam ver os comentários adicionados.

Quadro 20 – Documentação do Caso de Uso - Gerência relatórios de prestador de serviço.  
Fonte: Elaborado pelos autores (2007).

<b>Gerência de prestação de serviço</b>	
<b>Identificador:</b>	CSU010
<b>Descrição:</b>	Nesse pacote será realizado a inclusão de serviços realizados nos animais pela prestadora de serviço.
<b>Ator Primário:</b>	Gerente - Prestserv / Financeiro – Prestserv / Administrador – Prestserv
<b>Ator Secundário:</b>	- -
<b>Pré-condição:</b>	O Usuário com permissão de Gerente – Prestserv, Financeiro – Prestserv ou Administrador – Prestserv estar logado no Sistema.
<b>Fluxo de tarefas:</b>	1 - Cadastrar serviço prestado ao animal. 2 - Cadastrar comentários relacionados ao animal
<b>Pós-condição:</b>	1 – Nessa tarefa, será cadastrado o serviço que foi realizado no animal, mas o serviço só poderá ser realizado, caso o prestador de serviço tenha permissão para a realização do mesmo. 2 – Nessa tarefa, o prestador de serviço poderá adicionar comentários sobre o animal.

Quadro 21 – Documentação do Caso de Uso - Gerência de prestação de serviço.

Fonte: Elaborado pelos autores (2007).

<b>Gerência de inclusão de prestador de serviço</b>	
<b>Identificador:</b>	CSU011
<b>Descrição:</b>	Este pacote refere-se à gerência de inclusão do prestador de serviço no plano de saúde para a realização de atendimentos nos animais domésticos.
<b>Ator Primário:</b>	Gerente - Prestserv / Financeiro – Prestserv / Administrador – Prestserv
<b>Ator Secundário:</b>	- -
<b>Pré-condição:</b>	O Usuário, com permissão de Gerente – Prestserv, Financeiro – Prestserv ou Administrador – Prestserv estar logado no Sistema.
<b>Fluxo de tarefas:</b>	1 - Cadastrar prestador de serviço. 2 - Cadastrar serviços que são prestados pelo prestador de

	serviço
<b>Pós-condição:</b>	<p>1 – Nessa tarefa, o prestador realiza o seu cadastro para que o mesmo possa prestar serviço junto a clínica – O prestador de serviço só poderá realizar prestação de serviço junto ao plano, após a autorização do mesmo.</p> <p>2 – Nessa tarefa, o prestador de serviço, após a realização do seu cadastro, adicionará os serviços que ele presta na clínica para que o plano de saúde analise sua solicitação.</p>

Quadro 22 – Documentação do Caso de Uso - Gerência de inclusão de prestador de serviço.

Fonte: Elaborado pelos autores (2007).

<b>Gerência de contas de usuário da Prestserv</b>	
<b>Identificador:</b>	CSU012
<b>Descrição:</b>	Nesse pacote, será realizada a gerência de contas de usuário da prestadora de serviço.
<b>Ator Primário:</b>	Administrador – Prestserv
<b>Ator Secundário:</b>	Gerente - Prestserv / Financeiro – Prestserv
<b>Pré-condição:</b>	O Usuário, com permissão de Administrador – Prestserv estar logado no Sistema.
<b>Fluxo de tarefas:</b>	<p>1 - Cadastrar acesso de usuário do prestador de serviço.</p> <p>2 - Ativar acesso de usuário do prestador de serviço.</p> <p>3 - Desativar acesso de usuário do prestador de serviço.</p>
<b>Pós-condição:</b>	<p>1 - Nessa tarefa, o administrador poderá incluir novos usuários para o gerenciamento do plano de saúde, no mesmo terá três níveis de usuário que ele pode cadastrar.</p> <p>2 - Nessa tarefa, o administrador poderá ativar o usuário para que o mesmo possa acessar ao modulo plano de saúde.</p> <p>3 – Nessa tarefa, o administrador desativa os usuários do sistema no módulo plano de saúde para que o mesmo não possa mais acessar ao sistema, assim, bloqueando o usuário.</p>

Quadro 23 – Documentação do Caso de Uso - Gerência de contas de usuário da Prestserv.

Fonte: Elaborado pelos autores (2007).

<b>Gerência de relatórios financeiros</b>	
<b>Identificador:</b>	CSU013
<b>Descrição:</b>	Nesse caso de uso, será realizada a gerência dos relatórios da parte financeira da prestadora de serviço.
<b>Ator Primário:</b>	Financeiro – Prestserv / Administrador - Prestserv
<b>Ator Secundário:</b>	- -
<b>Pré-condição:</b>	O Usuário com permissão de Gerente – Prestserv, Financeiro – Prestserv ou Administrador – Prestserv estar logado no Sistema.
<b>Fluxo de tarefas:</b>	<p>1 - Relatório de reembolso pendente junto ao plano de saúde.</p> <p>2 - Relatório de serviços prestados realizado no período.</p> <p>3 - Relatório de reembolso recebido no período.</p> <p>4 - Relatório de valor de reembolso por cada tipo de serviço prestado.</p>
<b>Pós-condição:</b>	<p>1 – Nessa tarefa, é gerado para o prestador de serviço o valor parcial e total de serviços que ainda não lhe foi reembolsado pelo plano de saúde.</p> <p>2 – Nessa tarefa, é gerado um relatório com dados de inclusão de serviços prestados por esse prestador de serviços, a todos os animais do plano, dentro de determinado período, incluindo dados financeiros, se foi pago pela prestadora ou não, que não estarão presentes no relatório de serviços prestados junto ao animal.</p> <p>3 - Nessa tarefa, gera-se um relatório para o prestador de serviços com o valor parcial e total de serviços que lhe foram reembolsados pelo plano de saúde em determinado período.</p> <p>4 – Nessa tarefa, gera-se um relatório para o prestador de serviço o valor de reembolso de cada tipo de serviço oferecido pelo mesmo.</p>

Quadro 24 – Documentação do Caso de Uso - Gerência de relatórios financeiros.

Fonte: Elaborado pelos autores (2007)



### 3.3.16 Casos de Uso – Pacote Proprietário (ICONIX)

Neste capítulo, mostrar-se-á a documentação dos casos de uso em que se apresentarão as tarefas de cada caso de uso;

Gerência de relatórios do proprietário	
<b>Identificador:</b>	CSU014
<b>Descrição:</b>	Nesse caso de uso, serão documentados os relatórios de uso do proprietário
<b>Ator Primário:</b>	Proprietário – Prop
<b>Ator Secundário:</b>	- -
<b>Pré-condição:</b>	O Usuário, com permissão de Proprietário – Prop, estar logado no Sistema.
<b>Fluxo de tarefas:</b>	1 - Relatório de situação junto ao plano. 2 - Relatório de serviços prestados junto ao animal. 3 - Relatório de comentários relacionados ao animal.
<b>Pós-condição:</b>	1 - Nessa tarefa, é gerado um relatório com dados relacionados ao proprietário e seus animais, tratando de finanças, datas, carências para os serviços prestados ao animal, pendências, etc. 2 – Nessa tarefa, é gerado um relatório com todos os dados anteriores de serviços prestados no animal por prestadores de serviço. 3 – Nessa tarefa, será gerado um relatório com os comentários extras adicionados anteriormente pelo proprietário do animal, que possam interessar ao veterinário em atendimentos futuros.

Quadro 25 – Documentação do Caso de Uso - Gerência de relatórios do proprietário.  
Fonte: Elaborado pelos autores (2007).

Cadastro de proprietário	
<b>Identificador:</b>	CSU015
<b>Descrição:</b>	Nesse caso de uso, o proprietário irá realizar a inclusão do mesmo no sistema para que ele possa incluir os animais de estimação no plano de saúde.

<b>Ator Primário:</b>	Proprietário – Prop
<b>Ator Secundário:</b>	- -
<b>Pré-condição:</b>	O Usuário, com permissão de Proprietário – Prop, estar logado no Sistema.
<b>Fluxo de tarefas:</b>	1 - Cadastrar proprietário. 2 - Alterar dados do proprietário.
<b>Pós-condição:</b>	1 – Nessa tarefa, o proprietário poderá cadastrar no sistema um novo proprietário cujos animais poderão ser, posteriormente, cadastrados e inscritos em um plano. 2 – Nessa tarefa, o proprietário poderá alterar, no próprio cadastro do proprietário, dados de contato, endereço, etc.

Quadro 26 – Documentação do Caso de Uso - Cadastro de proprietário.

Fonte: Elaborado pelos autores (2007).

Gerencia de Animal	
<b>Identificador:</b>	CSU016
<b>Descrição:</b>	Nesse caso de uso, o proprietário irá gerenciar o que se refere ao animal e à emissão de boletos bancários.
<b>Ator Primário:</b>	Proprietário – Prop
<b>Ator Secundário:</b>	- -
<b>Pré-condição:</b>	O Usuário, com permissão de Proprietário - Prop estar logado no Sistema.
<b>Fluxo de tarefas:</b>	1 - Cadastrar animal/plano de saúde. 2 - Emitir Boleto Bancário. 3 - Cadastrar comentários relacionados ao animal. 4 - Alterar dados do animal.
<b>Pós-condição:</b>	1 – Nessa tarefa, o proprietário poderá cadastrar novo animal junto ao plano de saúde, que, após autorização, será atendido pelo plano na rede credenciada (prestadora de serviço). 2 – Nessa tarefa, o proprietário pode realizar a impressão do boleto para que o mesmo possa efetuar o pagamento da fatura do mês do plano de saúde. 3 – Nessa tarefa, o proprietário ou prestadora de serviço poderá

	<p>inserir comentários de observação própria com relação a seu animal, que possam ser importantes ao medico veterinário em atendimentos futuros.</p> <p>4 – Nessa tarefa, o proprietário pode realizar alterações do animal, tais como, peso, altura, etc.</p>
--	--

Quadro 27 – Documentação do Caso de Uso - Gerencia de Animal.

Fonte: Elaborado pelos autores (2007),

### 3.3.17 Diagramas Complementares do Pacote Plano de Saúde (ICONIX)

Para Blaha (2006, p.145), diagrama de atividade é uma seqüência de etapas que compõem o processo de um fluxo de trabalho. Esse diagrama é semelhante ao diagrama de seqüência, só que esse diagrama dá mais enfoque nas operações.

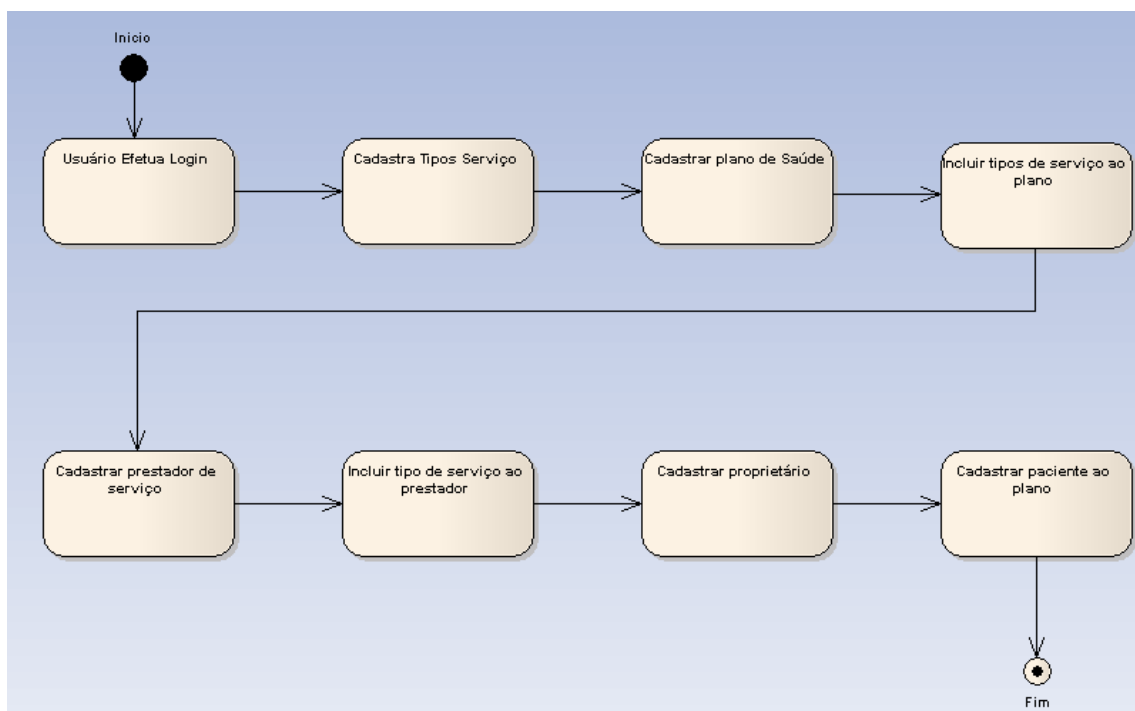


Figura 36 – Diagrama de atividade – plano de saúde.

Fonte: Elaborado pelos autores (2008).

O diagrama de atividade ilustra a execução dos diferentes casos de uso do pacote plano de saúde.

Os diagramas de robustez e diagramas de seqüência serão apresentados por casos de uso.

### 3.3.17.1 Diagrama de Gerência de Tipo de Serviço Prestado

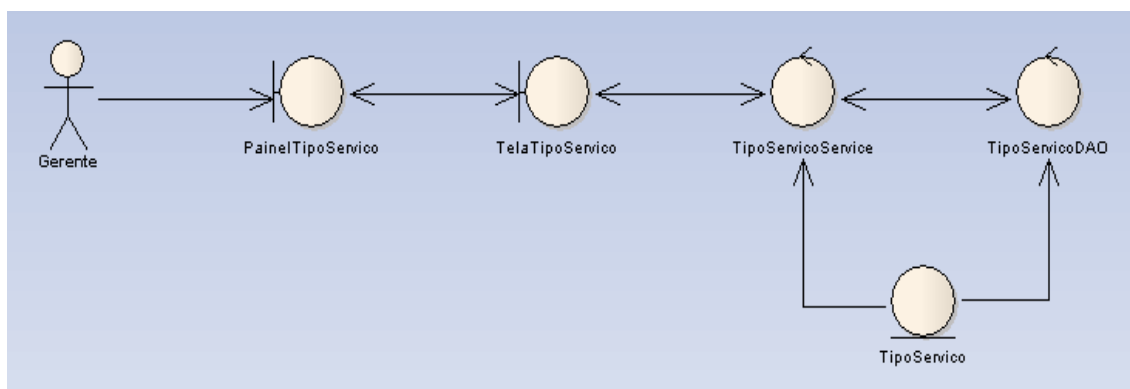


Figura 37 – Diagrama de robustez – Cadastro tipo serviço.

Fonte: Elaborado pelos autores (2008).

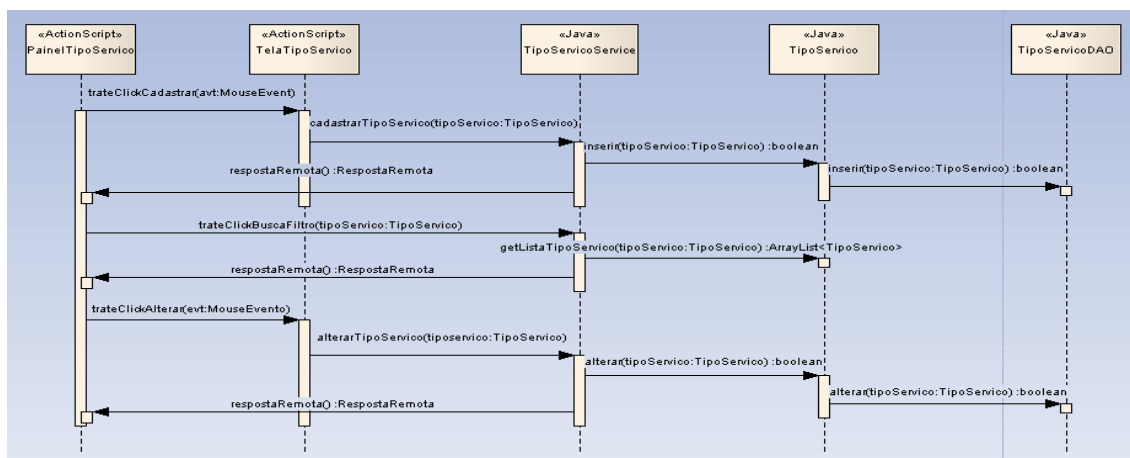


Figura 38 – Diagrama de seqüência – Cadastro tipo serviço.

Fonte: Elaborado pelos autores (2008).

### 3.3.17.2 Diagrama de Gerência de Plano de Saúde.

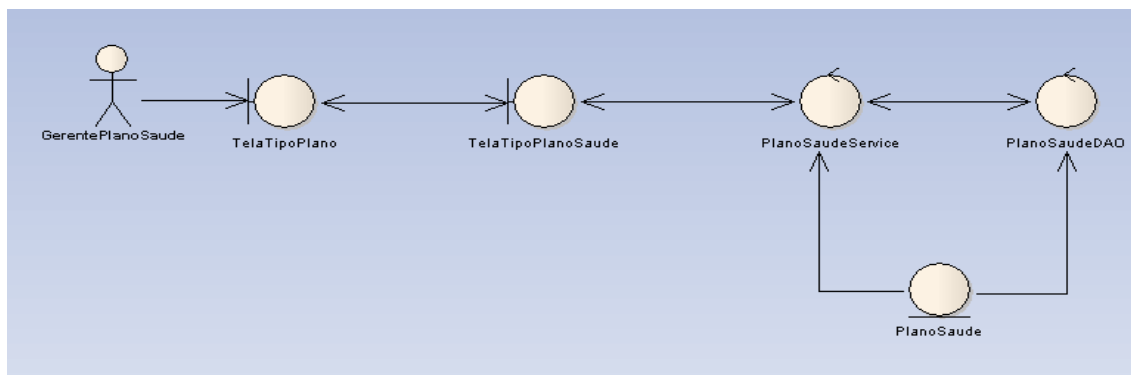


Figura 39 – Diagrama de robustez – inclusão/alteração de plano de saúde.  
Fonte: Elaborado pelos autores (2008).

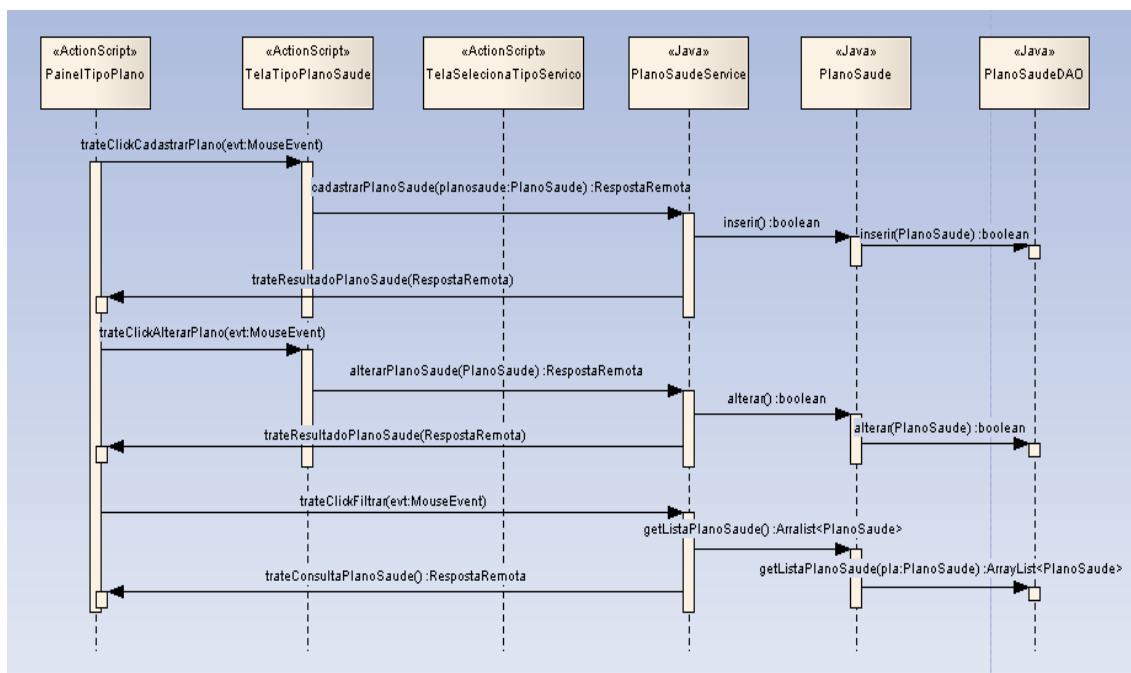


Figura 40 – Diagrama de sequência – inclusão/alteração de plano de saúde.  
Fonte: Elaborado pelos autores (2008).

### 3.3.17.3 Diagrama de Gerência de Tipo Serviço ao Plano.

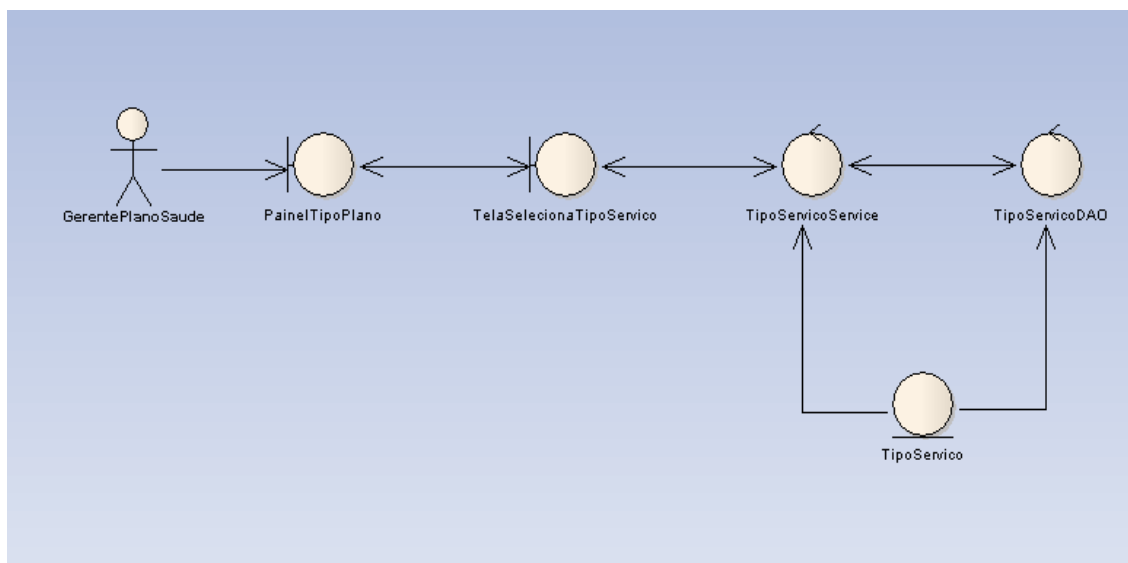


Figura 41 – Diagrama de robustez – inclusão/exclusão tipo serviço no plano de saúde.  
Fonte: Elaborado pelos autores (2008).

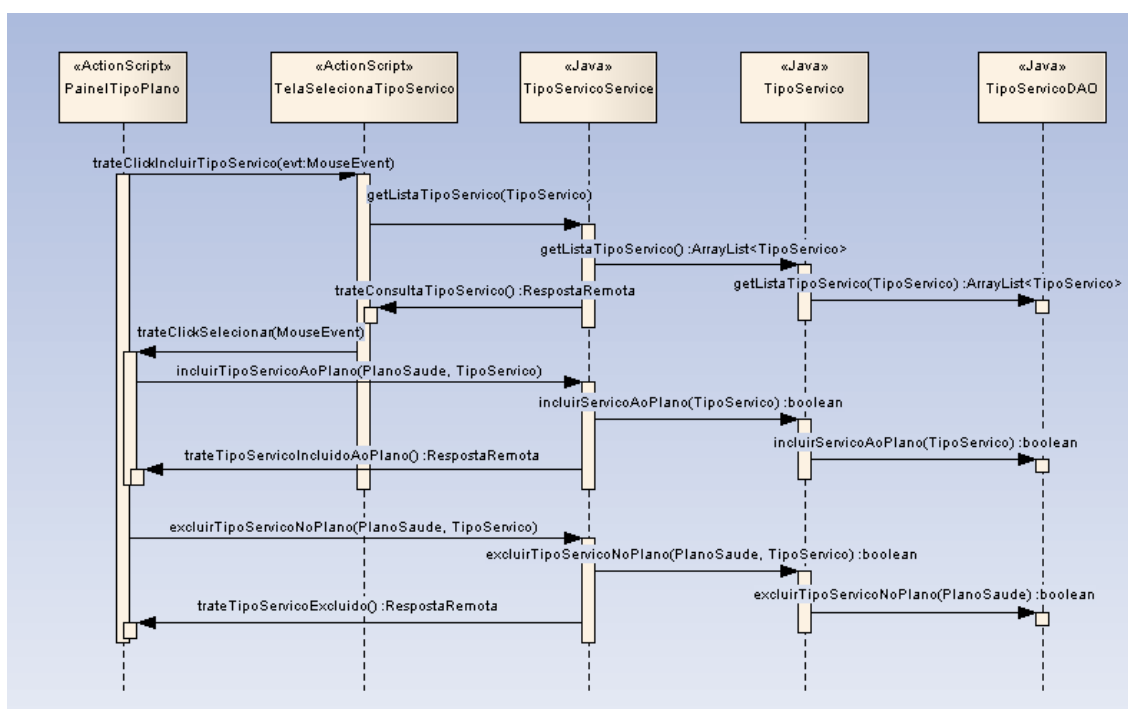


Figura 42 – Diagrama de sequência – inclusão/exclusão tipo serviço no plano de saúde.  
Fonte: Elaborado pelos autores (2008).

### 3.3.17.4 Diagrama de Gerência de Inclusão e Alteração Prestador Serviço.

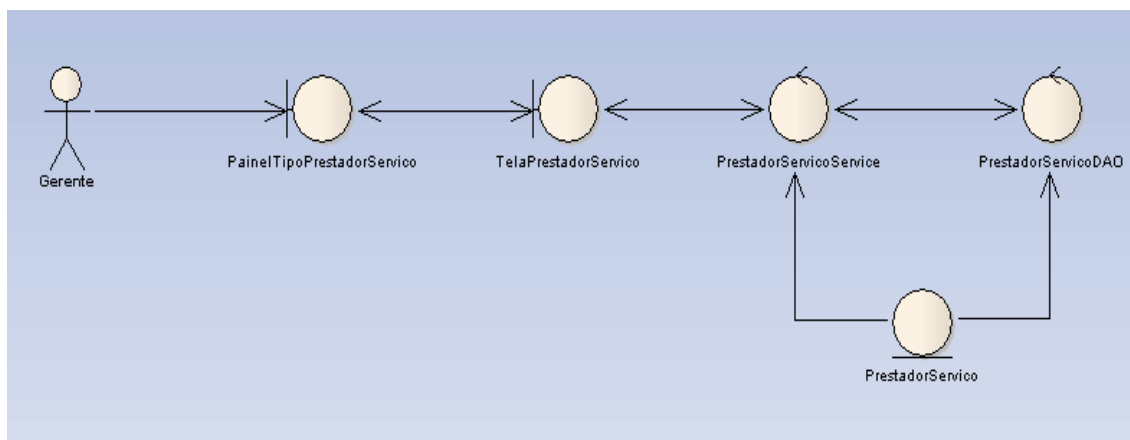


Figura 43 – Diagrama de robustez – inclusão/alteração prestador de serviço.

Fonte: Elaborado pelos autores (2008).

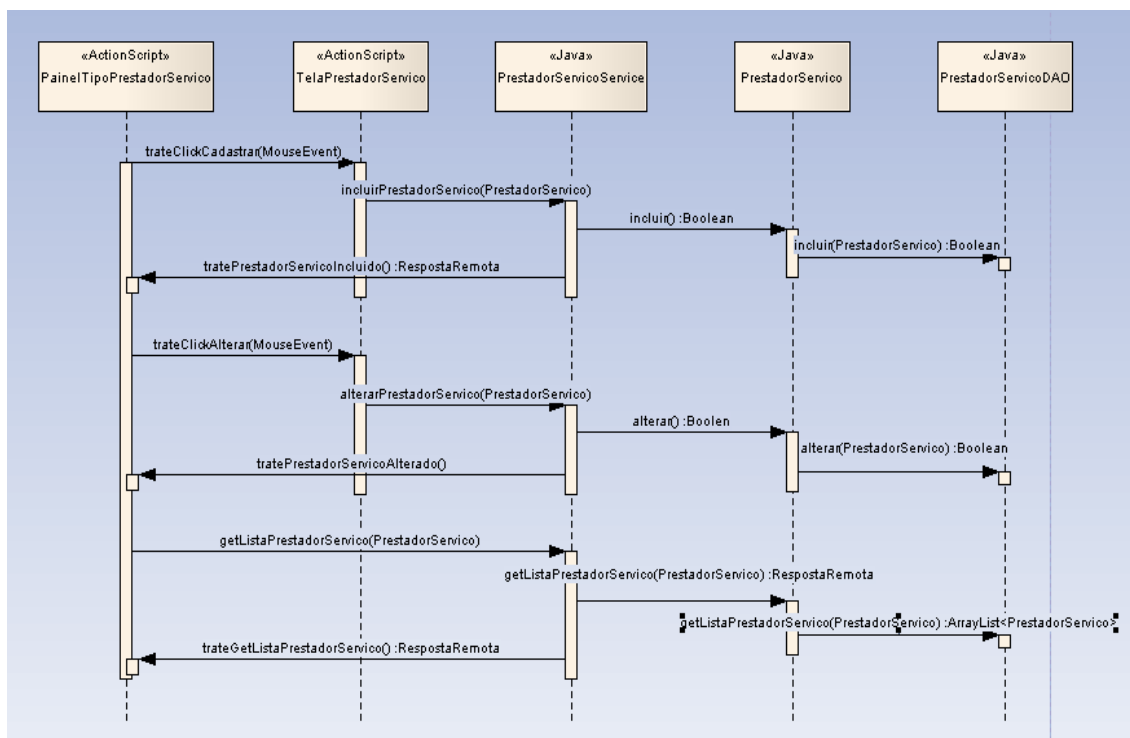


Figura 44 – Diagrama de sequência – inclusão/alteração prestador de serviço.

Fonte: Elaborado pelos autores (2008).

### 3.3.17.5 Diagrama de Gerência de Tipos de Serviço ao Prestador

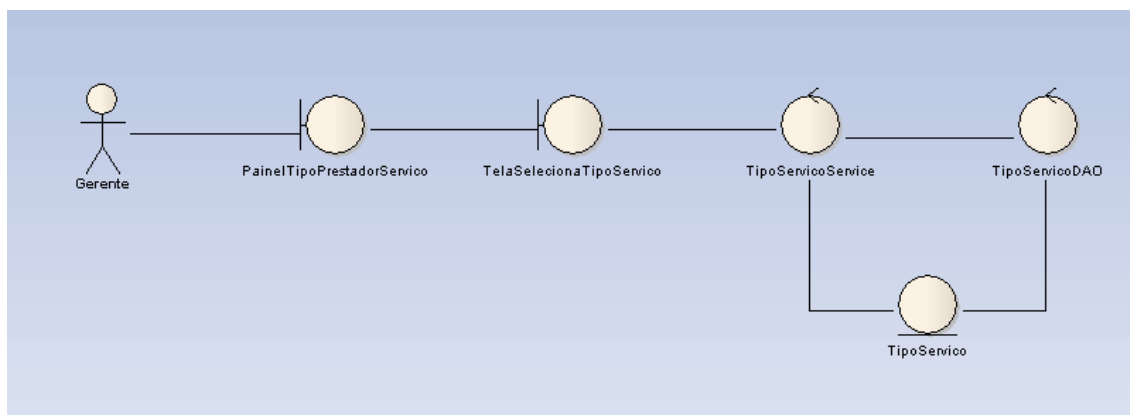


Figura 45 – Diagrama de robustez – Tipos de serviços ao prestador de serviço.  
 Fonte: Elaborado pelos autores (2008)

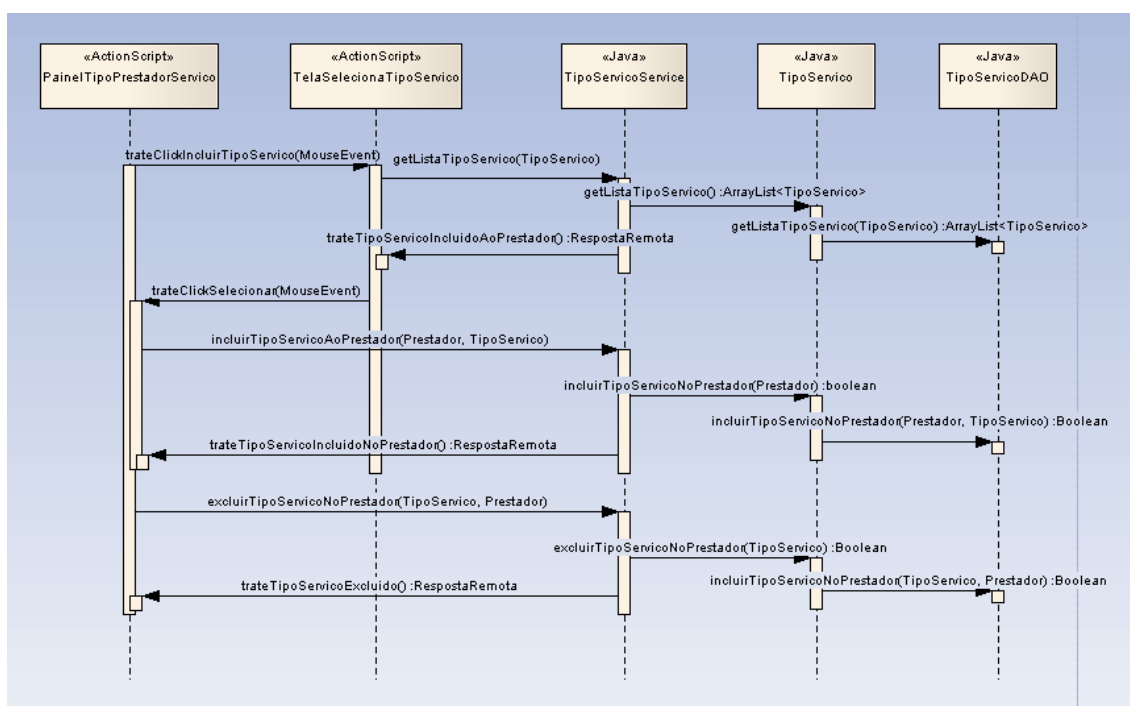


Figura 46 – Diagrama de sequência – Tipos de serviços ao prestador de serviço.  
 Fonte: Elaborado pelos autores (2008)



### 3.3.17.6 Diagrama de Gerência de Cadastro e Alteração de Proprietário

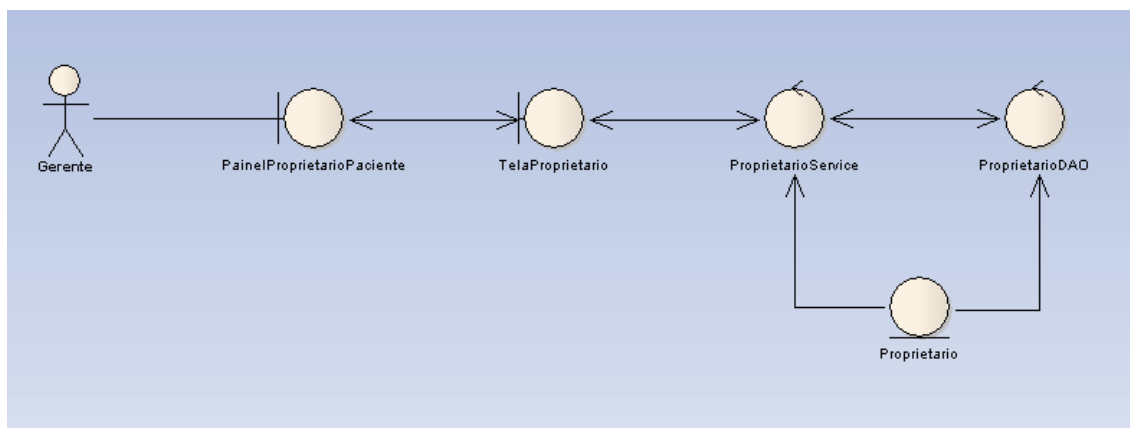


Figura 47 – Diagrama de robustez – cadastro e alteração do proprietário.  
 Fonte: Elaborado pelos autores (2008)

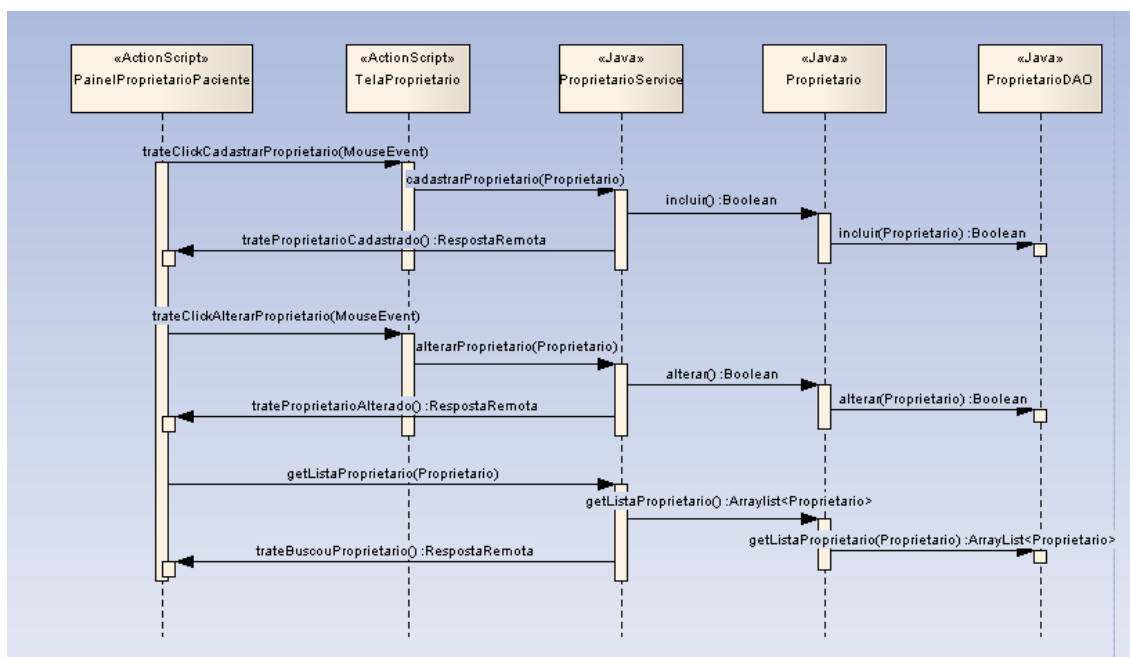


Figura 48 – Diagrama de sequência – cadastro e alteração do proprietário.  
 Fonte: Elaborado pelos autores (2008)

### 3.3.17.7 Diagrama de Gerência de Alteração do Paciente

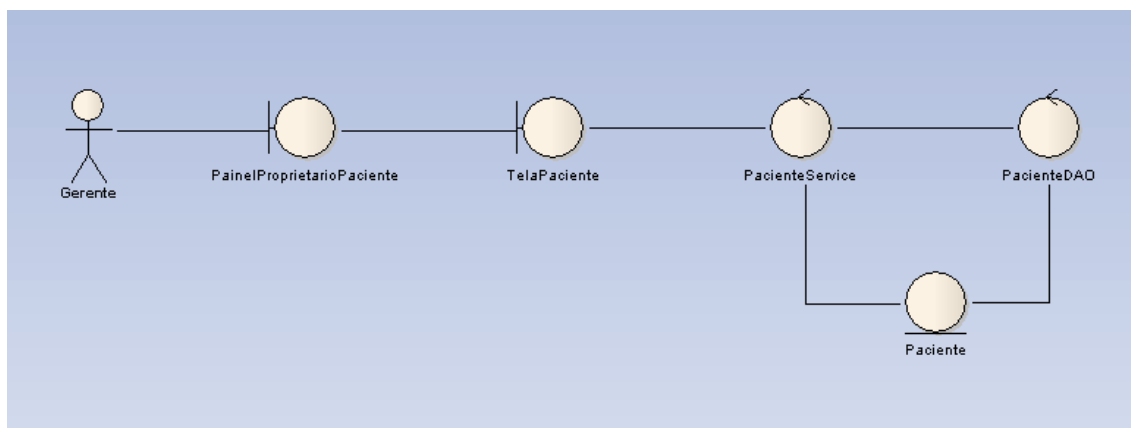


Figura 49 – Diagrama de robustez – cadastro e alteração do paciente.

Fonte: Elaborado pelos autores (2008)

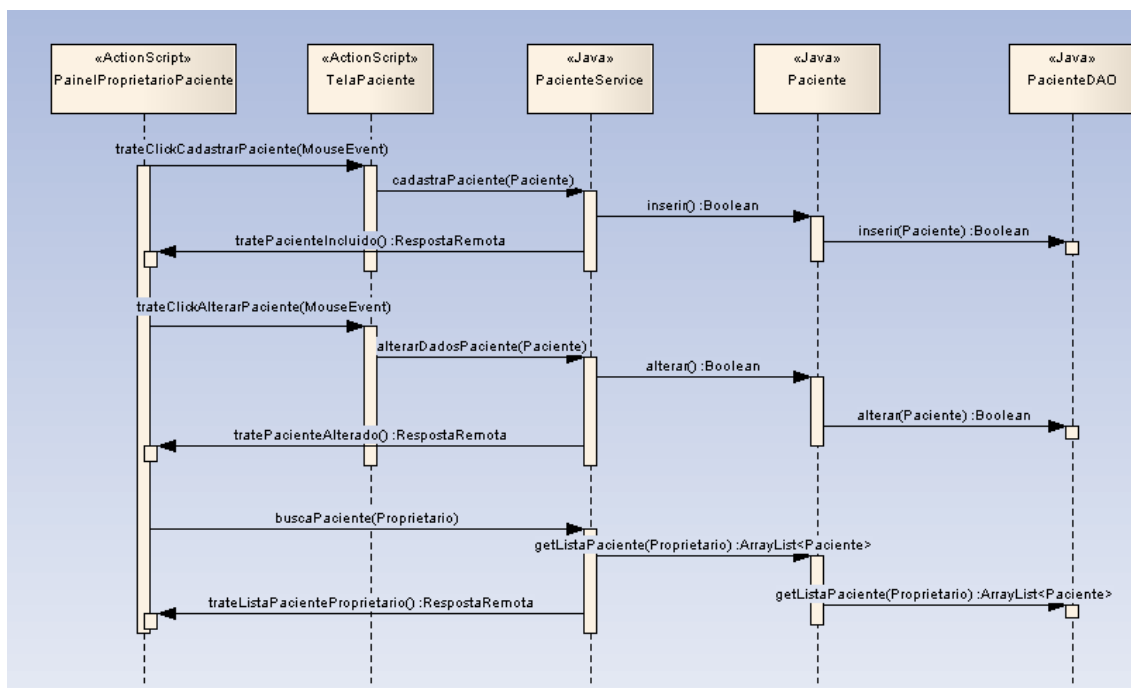


Figura 50 – Diagrama de sequência – cadastro e alteração do paciente.

Fonte: Elaborado pelos autores (2008)

### 3.3.18 Diagrama de Classe Persistente

Segundo Ambler (1978 *apud* ZINDEL 2008), os diagramas de classes são responsáveis pela estrutura formada e suas relações. As relações entre classes podem ser de três tipos. São elas: Associações, agregações e herança. As classes são compostas da seguinte forma: De um nome, seguido pelos atributos e, por último, as operações.

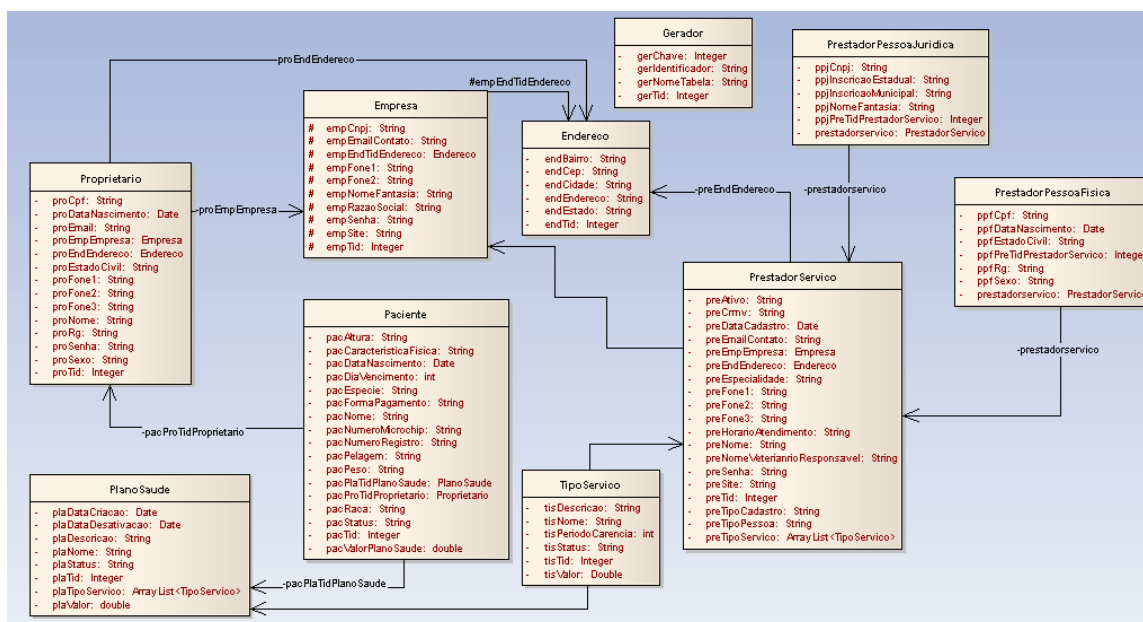


Figura 51 – Diagrama de classe – plano saúde.

Fonte: Elaborado pelos autores (2008).

Esse diagrama nos mostra a persistência das classes com as tabelas do banco de dados e de como elas se relacionam entre si.

### 3.3.19 Considerações Finais

Com a aplicação das metodologias, obteve-se um maior entendimento nos processos do módulo proprietário paciente seus processos e objetos. Observou-se, nesta etapa, a dificuldade encontrada no processo de análise, interpretar as necessidades do cliente que, aparentemente, seria um procedimento rápido, foi refeito, repetidas vezes, até se obter a correta especificação.

## 4. DESENVOLVIMENTO E VALIDAÇÃO

Neste capítulo, serão apresentados os recursos tecnológicos que foram utilizados para a construção do sistema de Plano de Saúde Animal. As ferramentas utilizadas estão descritas focando seu uso no desenvolvimento.

### 4.1 AMBIENTE DE DESENVOLVIMENTO

No ambiente de desenvolvimento, foram utilizadas ferramentas como o banco de dados MySQL, o servidor Apache Tomcat que permite a comunicação na *Web*, a linguagem Java, concomitantemente, com a linguagem ActionScript e MXML, tornando o sistema dinâmico e permitindo ao usuário interagir remotamente com o fluxo de dados entre o sistema e banco de dados.

A seguir, são apontadas as tecnologias que foram utilizadas ao longo do projeto:

- Adobe Flex Builder 3.0 – Ferramenta para utilização de interface gráfica.
- ActionScript – Linguagem de programação que permite validações e interação com o sistema.
- BlazeDs – Ferramenta responsável pela integração remota entre as linguagens de programação ActionScript e Java.
- Java – Linguagem de programação utilizada para manusear a base de dados com a ferramenta FLEX e o banco de dados.
- Corel Draw – Ferramenta para criação de imagens ou ícones para o sistema.
- Corel Photo-Paint- Ferramenta para editar as imagens e aperfeiçoar sua exibição.
- Eclipse 3.0 – Ferramenta utilizada na edição do código fonte da linguagem de programação (Java).

- Servidor Tomcat 6.0 – Servidor utilizado na conexão do sistema via Web.

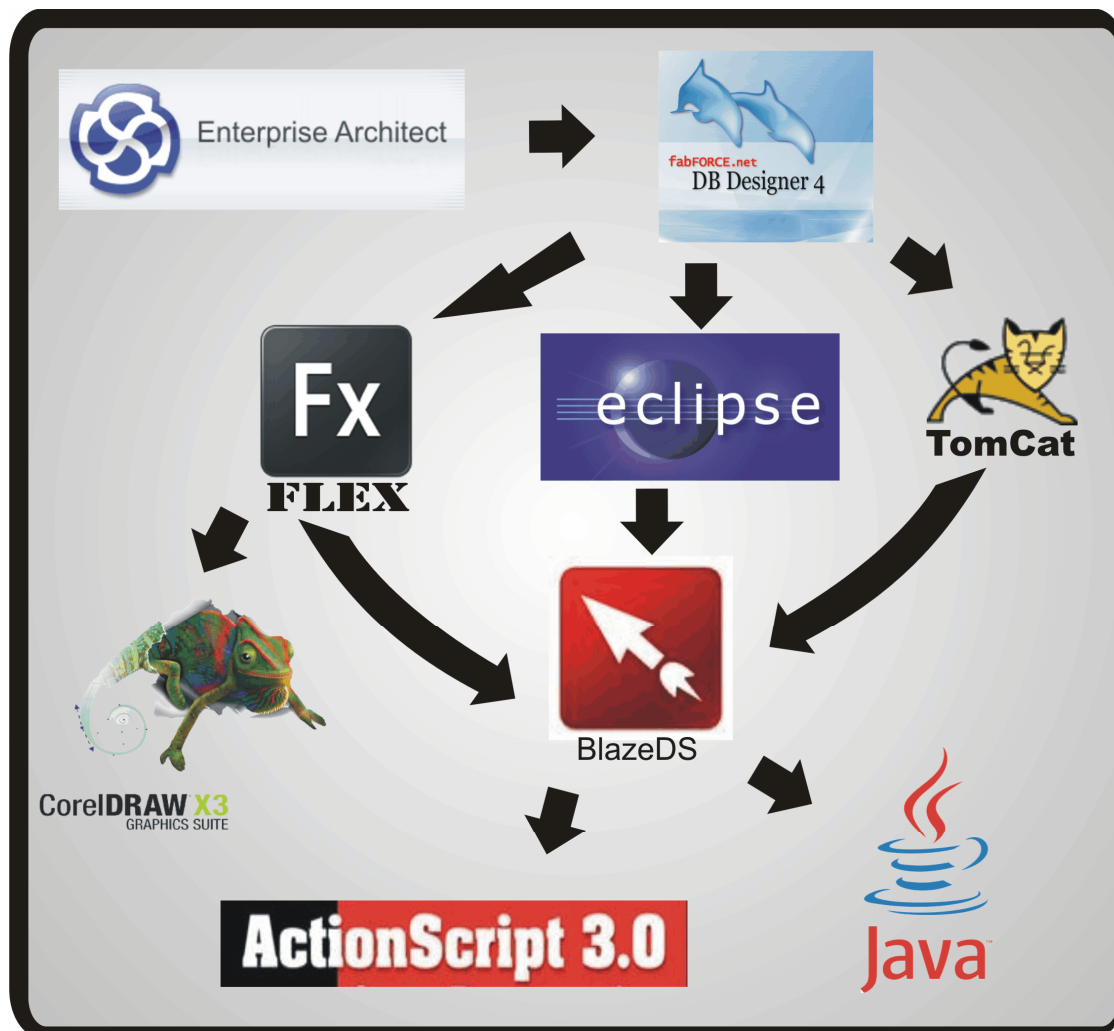


Figura 52 – Ambiente de Desenvolvimento do Sistema.  
Fonte: Elaborado pelos autores (2008).

#### 4.1.1 Enterprise Architect 7.0

De acordo com o Sparx Systems(2008), esta ferramenta Enterprise Architect 7.0 super poderosa, tem o recurso de criar modelos para projetar a arquitetura de um sistema. Toda a base do projeto se baseia nesse software, em que a modelagem é analisada e gerada em algum tipo de documento.

O objetivo do Enterprise Architect é ajudar a projetar o produto de software que será desenvolvido, e ele nos traz vários segmentos como, Caso de usos, Diagramas de seqüência, diagrama de robustez e outros. Toda parte de modelagem é concentrada nessa ferramenta e, depois, é passado aos desenvolvedores, para uma boa clareza e entendimento do que será produzido nesse software.

#### **4.1.2 Adobe Flex Builder 3.0**

De acordo com a Adobe Flex (2008), essa ferramenta é baseada na IDE de desenvolvimento Eclipse. O objetivo da utilização dessa ferramenta foi à criação de novos aplicativos para internet. Seu visual é considerado inovador, oferecendo um novo conceito para seus clientes.

A ferramenta foi escolhida por suas diferentes características, relacionadas, abaixo, conforme (ADOBE FLEX ,2008):

- Aplicação multi-plataforma (Windows, Mac e Linux).
- Linguagem Orientada a Objetos.
- Facilidade no desenvolvimento de Designer.
- Tecnologias de suporte já implementadas na maioria do sistema.
- Acessos facilitados a serviços web XML.

Segundo Pinho et al. (2008), o Adobe Flex tem a tecnologia do Flash, sendo que, atualmente, 95% dos computadores já estão com o Flash player instalado e estão prontos para rodar os aplicativos na *web*.

O Flex é implementado pelas linguagens MXML e ActionScript.

A linguagem MXML<sup>3</sup> é responsável pela estrutura de design, em que você poderá pegar componentes da Adobe ou criar novos componentes para futuros desenvolvimentos. Quando

---

<sup>3</sup> Macromedia Extensible Markup Language

você cria um aplicativo de design, automaticamente, cria um MXML e, quando é compilado ou seja rodado, o aplicativo gera os arquivos HTML e SWF.

Segundo DEHAAN (2004), a linguagem ActionScript consiste em escrever aplicativos orientando objetos, desta maneira ajuda na reutilização de código fonte. Para os desenvolvedores, a linguagem ActionScript é conhecida por AS, já que sua extensão é dessa forma, já que, o AS é baseado em objetos, é uma boa forma de organizar por pacotes, aproximando ao estilo Java.

Um evento no AS é disparado quando um SWF é executado. Ele dispara ações conforme a entrada do teclado ou mouse no qual reproduz suas operações e aplicações, utilizando métodos externos para envio e retorno de informações.

#### **4.1.3 Eclipse 3.0**

Essa ferramenta foi utilizada para facilitar o desenvolvimento de códigos e bibliotecas prontas. O Flex é acoplado ao eclipse e possui o plug-in para a conectividade com o Eclipse, podendo, assim, visualizar a interface e aplicações feitas em Java.

De acordo com Mundo Java (2005), o eclipse é uma IDE de nível popular para os programadores Java. O eclipse possui diversos plugins para sua utilização, dessa forma o programador tem a possibilidade de trabalhar vários tipos de ambiente.

#### **4.1.4 Java**

Segundo Deitel (2001), Java é uma linguagem que atrai pelo fato de ser totalmente orientada a objeto e, também, por seu grande avanço no mercado. Java é uma linguagem de programação que vem evoluindo muito nestes últimos tempos e vem sendo muito utilizada para criação de páginas de *web* em uma forma dinâmica e interativa.

Os programas Java consistem em partes que são chamados de classes e, em seqüência, possuem os métodos que são responsáveis por realizarem tarefas e, após concluídas suas tarefas, buscam-se os dados que foram acionados a esta tarefa.

Deitel (2001) destaca que, ao invés de escrever (codificar) as próprias versões, é recomendado utilizar a biblioteca de classes e métodos. Com essa atitude, ganha-se em portabilidade e, também, em desempenho.

De acordo com Javafree (2008), Java é uma linguagem de programação multi-plataforma, que é desenvolvida para gerar um software executado em um ambiente distribuído.

#### **4.1.5 Apache Tomcat**

Segundo D'Àvila (2008), Tomcat é um servidor que permite a execução de aplicações Java para *Web*. Trata-se de um software livre e de código aberto. O Tomcat possui um projeto de desenvolvimento independente sendo considerado eficiente e robusto, permitindo seu uso em ambiente de produção.

Tecnicamente, o Tomcat abrange as tecnologias Servlet e JSP, incluindo, também, tecnologias de segurança. O servidor tem a capacidade de atuar com web/http autônomo ou, ainda, funciona de forma integrada a um servidor web dedicado, tal como apache httpd ou Microsoft IIS.

#### **4.1.6 DbDesigner**

Segundo Imasters (2008), o DbDesigner é uma ferramenta de código livre que permite a modelagem de banco de dados, especificamente, diagramas ER (Entidade – Relacionamento). Sua principal vantagem é gerar os códigos SQLs.

De acordo com Bonetto (2008), a ferramenta DbDesigner suporta não apenas MySQL como outros bancos de dados, tais como, Oracle, MS SQL Server, SQLite e outros. O DbDesigner é multi-plataforma.

#### **4.1.7 MySQL**

De acordo com Soares (2001), a melhor forma de armazenar e gerenciar dados é através de um gerenciador como o MySQL. Antigamente, poucas empresas utilizavam gerenciadores



por causa do alto custo da despesa, mas isso mudou com a chegada do MySQL por ser uma ferramenta free (gratuita). Outro fato é que o MySQL é uma ferramenta muito fácil de ser utilizada e gerenciada e, ainda, disponibiliza de alguns comandos extras para manipulação dos dados.

De acordo com a Infowester (2008), o MySQL é um SGDB mais popular que existe. É amplamente utilizado na internet, permitindo de foma portátil a gerência para qualquer sistema operacional.

O MySQL possui uma alta compatibilidade com a linguagem de programação Java. Seu processamento é de baixa exigência e é considerado seguro em termos de conectividade.

O MySQL foi criado na Suécia por Allan Larsson, David Axmark e Michael Monty Widenius, e lançada sua primeira versão em 1996.

#### **4.1.8 Corel Draw X3**

O Corel Draw é composto por um conjunto de ferramentas. Neste projeto foram utilizados apenas o próprio Corel Draw e o Corel Photo-Paint, cujas características serão descritas abaixo:

- Corel Draw - Segundo Locasite (2008), Corel Draw é uma ferramenta para se construir desenhos vetoriais. Na criação de imagens, utiliza-se formas de objetos armazenados bem como de equações matemáticas, no caso o vetor, diferenciando das aplicações que utilizam no bitmap.
- Corel Photo-Paint – O Corol Photo-Paint é um editor de imagens. Nessa ferramenta, você pode fazer o melhoramento da imagem e gravar a imagem no formato que desejar.

#### **4.1.9 BlazeDS**

De acordo com a Adobe (2008) , BlazeDS é um servidor baseado em Java em que se permite que os desenvolvedores façam facilmente as comunicações remotas entre o Flex e a linguagem que se utilizaram em tempo real, tornado-se as aplicações ricas na internet.

### **4.2 APRESENTAÇÃO DO SISTEMA**

Na apresentação do sistema, as funcionalidades desenvolvidas para o pacote Plano de Saúde, são:

#### **4.2.1 Tela de Login**

Nessa tela (figura 53), identifica-se o administrador com a entrada de “Usuário” e “Senha”, após feito isso, as opções no topo são liberadas para gerenciar.

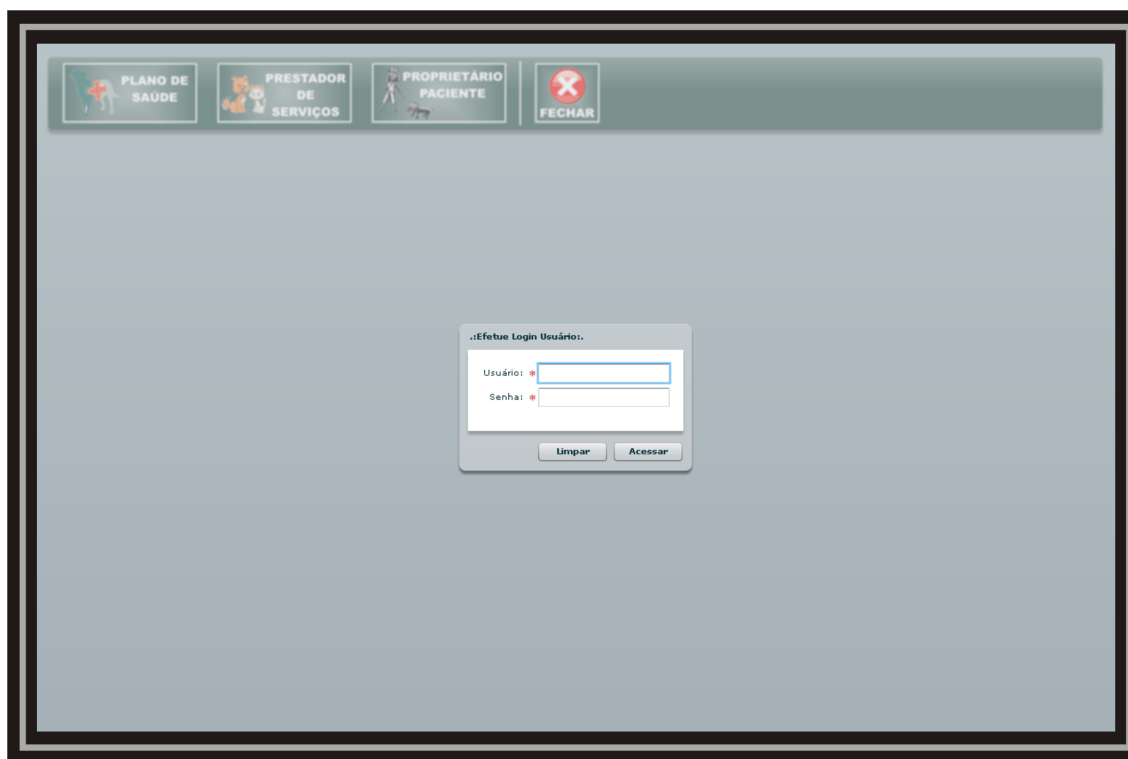


Figura 53 – Tela de Identificação do Administrador.

Fonte: Elaborado pelos autores (2008).

#### 4.2.2 Tela Gerenciamento Tipo de Serviço

Nessa tela (figura 54), vê-se que, ao apertar o bloco Plano de Saúde, abre-se um sub menu abaixo do topo, em seguida, a tela visualizada pertence ao Tipo de Serviço. Observa-se que ela possui um filtro para buscar algum tipo de serviço (do código ou pelo nome). A tela permite a classificação pelo status que o serviço possui, nesse caso, pode-se ter uma consulta apenas de serviços ativos ou vice versa. É possível listar todos os serviços pelo clicar no botão Filtrar.

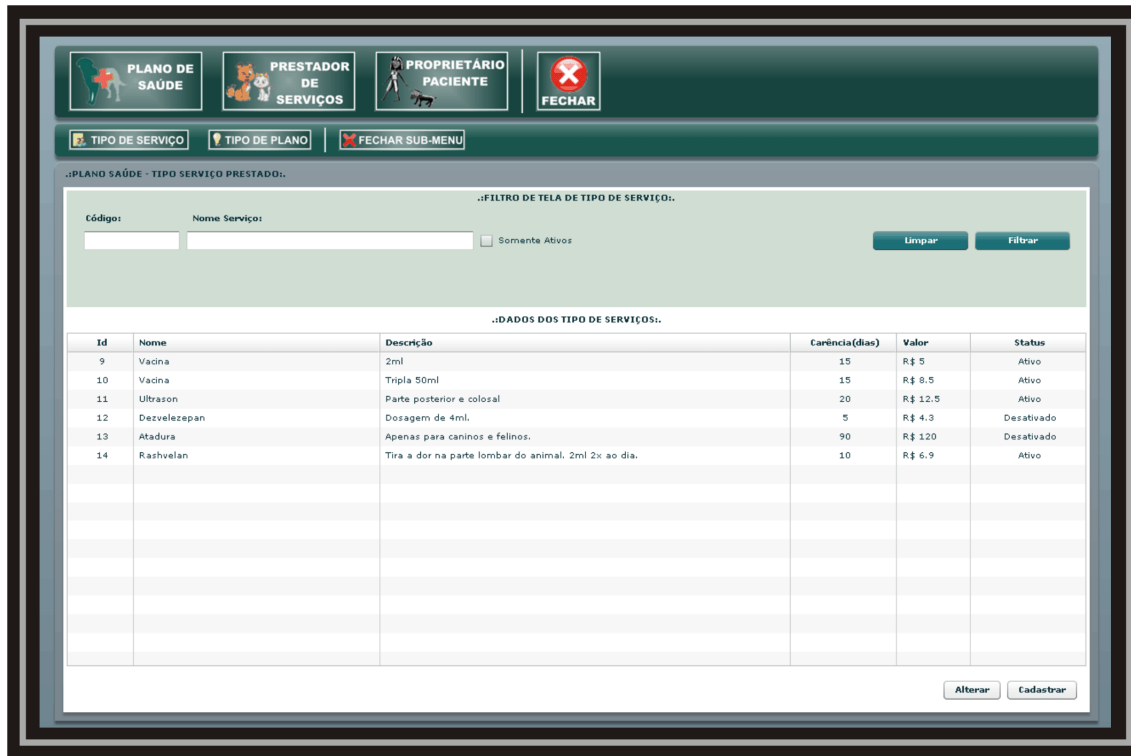


Figura 54 – Tela de Gerenciamento de Tipo de Serviço.

Fonte: Elaborado pelos autores (2008).

Ao clicar, no botão cadastrar, uma nova tela é sobreposta (figura 54), nela é possível cadastrar conforme as informações solicitadas.

..PLANO SAÚDE - TIPO SERVIÇO PRESTADO..

..FILTRO DE TELA DE TIPO DE SERVIÇO..

Nome: \*  
 Descrição: \*  
 Período de carência: \*  
 Valor: \*  
 Situação do Serviço: \* ☒ Ativo

Cancelar Limpar Campos OK

Id	Nome	Duração (dias)	Valor	Status
9	Vadna	15	R\$ 5	Ativo
10	Vadna	15	R\$ 8.5	Ativo
11	Ultrason	20	R\$ 12.5	Ativo
12	Dezvelezepan	5	R\$ 4.3	Desativado
13	Atadura	90	R\$ 120	Desativado
14	Rashvelan	10	R\$ 6.9	Ativo

Alterar Cadastrar

Figura 55 – Tela de cadastro de Serviços.

Fonte: Elaborado pelos autores (2008).

### 4.2.3 Tela Gerenciamento Plano de Saúde

Na tela Tipo de Plano (figura 56), é possível a busca por planos de saúde que estão cadastrados no sistema. Na opção Cadastrar, é solicitado o pedido e informados os dados e, em sequência, atualizados na grade. A opção de alteração só pode ser realizada em planos desativados para futuramente serem ativados. Quando um plano ativo é desativado, é registrado dia de sua ativação e o dia em que foi desativado para o administrador ter um controle sobre os planos.

Plano Saúde    Tipos de serviços do plano

..FILTRO DA TELA DE TIPOS DE PLANOS..

Código:     Nome de Plano:     Seleção Somente : ☐ Ativados

Id	Nome	Descrição	Valor	Data Criação	Data desativação	Situação
4	Plano Master	Liberado para todos os serviços.	550	04/05/2008		Ativo
5	Plano Intermediário	Não cobre operações, e plataforma bucal	150	04/05/2008	28/05/2008	Desativado
6	Plano Avançado	Não cobre operações de risco A e B	230	04/05/2008	28/05/2008	Desativado
7	Plano Bebe	plano minimo	2000	22/05/2008		Ativo

Figura 56 – Tela de Gerenciamento dos Planos de Saúde.

Fonte: Elaborado pelos autores (2008).

O exemplo acima é de plano de saúde selecionado (figura 56). O Administrador pode incluir ou excluir os serviços que se apresentam no plano de saúde.

#### 4.2.4 Tela Gerenciamento Plano de Planos com seus Respectivos Serviços

PLANO DE SAÚDE   PRESTADOR DE SERVIÇOS   PROPRIETÁRIO PACIENTE   FECHAR

TIPO DE SERVIÇO   TIPO DE PLANO   FECHAR SUB-MENU

Plano Saúde   Tipos de serviços do plano

Nome Plano Saúde	Data de Criação	Status
Plano Bebe	22/05/2008	Ativo

Descrição do Plano de Saúde

plano minimo

Tipos de serviços cadastrados nesse plano de saúde

Id	Nome	Descrição	Carência(dias)	Status
9	Vacina	2ml	15	Ativo
14	Rashvelan	Tira a dor na parte lombar do animal. 2ml 2x ao dia.	10	Ativo

Excluir   Incluir

Figura 57 – Tela Gerenciamento de Planos com seus respectivos serviços.  
 Fonte: Elaborado pelos autores (2008).

Ao pressionar o botão incluir (figura 57), é possível fazer a busca (código ou nome do serviço) por serviços (figura 58). Após fazer a busca, é selecionado um item para pertencer a esse tipo de plano. Quando selecionado, é atualizado na grade do plano de saúde.

#### 4.2.5 Tela para Buscar os Serviços e Cadastrar no Plano de Saúde



Figura 58 – Tela para buscar os serviços e os cadastrar no plano de saúde.  
Fonte: Elaborado pelos autores (2008).

Na tela (figura 59), ao pressionar o modulo Proprietário, um sub-menu será apresentado, nesta segunda aba é apresentado Proprietário/Paciente, quando clicado no botão Proprietário/Paciente aparecerá a tela de gerenciamento do proprietário que é administrado pelo administrador.



#### 4.2.6 Tela de Gerenciamento de Proprietário

**PLANO DE SAÚDE** **PRESTADOR DE SERVIÇOS** **PROPRIETÁRIO PACIENTE** **FECHAR**

**PROPRIETÁRIO/PACIENTE** **FECHAR SUB-MENU**

**Cadastro de Proprietário** **Cadastro de Paciente**

**.:FILTRO DE TELA DE PROPRIETÁRIO / PACIENTE:.**

CPF:  Nome Proprietário:

**Limpar** **Filtrar**

**.:DADOS PACIENTE / PROPRIETÁRIO:.**

Id	Nome	CPF	RG	Est. Civil	Data Nasc.	Contato	Email
1	mario	111111111111	1111	Solteiro	20/02/1984	(48) - 2222 - 2222	mscinho@gmail.com
2	cinho	999999999999	99999	Solteiro	22/02/1984	(48) - 2843 - 9483	mscinho@gmail.com
3	mario	222222222222	78932	Casado(a)	22/02/1983	(45) - 3458 - 3049	mscinho@hotmail.cc
5	Marina Marcon	777777777777	143	Solteiro(a)	05/04/1987	(48) - 3659 - 0599	aaaa@hotmail.com

**Alterar Proprietário** **Cadastrar Proprietário**

Figura 59 – Tela de Gerenciamento de Proprietário.

Fonte: Elaborado pelos autores (2008).

#### 4.2.7 Tela de Cadastro de Proprietário

O cadastro do usuário (figura 60) possibilita o cadastro de dados pessoais dos usuários.

**PLANO DE SAÚDE** | **PROPRIETÁRIO/PACIENTE**

**Cadastro de Proprietário**

Nome: \*

Data Nascimento: \*

Estado Civil: \* **Casado(a)** ▼

CPF: \*

RG: \*

Sexo: \* ☒ Masculino ☐ Feminino

Senha: \*

Confirmar Senha: \*

**Endereço** | **Contato**

Cidade: \*

Bairro: \*

Endereço: \*

CEP: \*

UF: \* **SC** ▼

Limpar | Filtrar

Cancelar | Limpar | OK | Cadastrar Proprietário

Id	Nome
1	marcio
2	cinho
3	marcio
5	Marina Marcon

Contato	Email
2222 - 2222	mscinho@gmail.com
2843 - 9483	mscinho@gmail.com
3458 - 3049	mscinho@hotmail.cc
3659 - 0599	aaaa@hotmail.com

Figura 60 – Tela de Cadastro do Proprietário.

Fonte: Elaborado pelos autores (2008).

#### 4.2.8 Tela de Gerenciamento de Paciente

Para realizar a busca de espécie(s), basta clicar duas vezes sobre o proprietário (figura 58) desejado, que apresentará a grade do paciente (figura 61).

The screenshot displays a web application interface for patient management. At the top, there are four buttons: 'PLANO DE SAÚDE' (Health Plan), 'PRESTADOR DE SERVIÇOS' (Service Provider), 'PROPRIETÁRIO PACIENTE' (Patient Owner), and 'FECHAR' (Close). Below these, there are two more buttons: 'PROPRIETÁRIO/PACIENTE' and 'FECHAR SUB-MENU'. The main content area is divided into two tabs: 'Cadastro de Proprietário' (Owner Registration) and 'Cadastro de Paciente' (Patient Registration). The 'Cadastro de Paciente' tab is active, showing a form with the following fields:

Id	Nome	CPF
5	Marina Marcon	05/04/1987

Below the patient table, there are three more fields: 'Telefone de Contato' (Contact Phone), 'E-mail de contato' (Contact Email), and 'Data Nascimento' (Date of Birth). The values are: (48) - 3659 - 0599, aaaa@hotmail.com, and 05/04/1987. A note states: 'O proprietario é responsavel pelos seguintes animais.' (The owner is responsible for the following animals.). Below this note is a table for animal records:

Id	Nome	Espécie	Raça	Data Nasc.	Forma Pagamento	Dia Vencimento	Status

At the bottom right of the form, there are two buttons: 'Alterar Paciente' (Change Patient) and 'Cadastrar Paciente' (Register Patient).

Figura 61 – Tela de Gerenciamento Paciente.

Fonte: Elaborado pelos autores (2008).

#### 4.2.9 Tela do Cadastro de Paciente

Na tela (figura 61), são cadastrados pacientes ao clicar em cadastrar Paciente (figura 62), permitindo, também, alteração de um paciente.

**PLANO DE SAÚDE - CADASTRO PACIENTE:**

Nome: \*

Data Nascimento: \*/\*/

Número do Registro: \*

Número Microchip: \*

Sexo: ☒ Macho ☐ Fêmea

Espécie: **Canino** ▼

Raça: \*

Peso: \* Altura: **0-10cm** ▼

Porte: **Pequeno** ▼

Pelagem: ☒ Curta ☐ Longa

Características Físicas: \*

Status: ☒ Ativo ☐ Desativado ☐ A autorizar

Plano de Saúde: **Plano Master** ▼

Forma de Pagamento: **Boleto Bancário** ▼

Dia de Vencimento: **05** ▼

Cancelar Limpar OK

**PLANO DE SAÚDE**

**PROPRIETÁRIO/PACIENTE**

Id	Nome
5	Marina Marcon

Telefone de Contato

(48) - 3659 - 0599

O proprietário é responsável por

Id	Nome

Vencimento Status

Paciente Cadastrar Paciente

Figura 62 – Tela do Cadastro de Paciente.

Fonte: Elaborado pelos autores (2008)

#### 4.2.10 Tela de Gerenciamento de Prestador de Serviço

No módulo Prestador de Serviços (figura 63), veja que meio de filtro para realizar as consultas. O resultado das consultas pode ser feito da seguinte maneira: Id do Prestador, nome do prestador ou pelo nome do veterinário responsável, também pode ser identificado pelo tipo de pessoa (física ou jurídica) e o status, mostrando a situação da busca, caso não opte por nem um desse recurso, ao apertar o botão filtrar todas, as informações são apresentadas na tela.

PLANO DE SAÚDE PRESTADOR DE SERVIÇOS PROPRIETÁRIO PACIENTE FECHAR

PRESTADOR DE SERVIÇOS FECHAR SUB-MENU

Prestador de Serviço Cadastro de Tipo de Serviço

.:FILTRO DE TELA DE PRESTADOR DE SERVIÇOS.:  
 Id Prestador: Nome Prestador: Veterinário Responsável:  
 .:Tipo de Pessoa.: .:Status.:  
☒ Todos ☐ Pessoa Física ☐ Pessoa Jurídica ☒ Todos ☐ Ativos ☐ Inativos ☐ A Autorizar Limpar Filtrar

.:DADOS DO PRESTADOR DE SERVIÇOS.:  

Id	Nome	Tipo de Pessoa	Contato	Email	Vet. Responsável	Situação
1	Sandro Hostfield	Pessoa Física	(47) - 3687 - 0	m@m.com	Dr. Rangerdelpt Corais	Ativo

Alterar Cadastrar

Figura 63 – Tela de Gerenciamento do Prestador de Serviços.

Fonte: Elaborado pelos autores (2008).

#### 4.2.11 Tela de Cadastro do Prestador de Serviço

Nessa tela (figura 64), após clicado no botão cadastrar, surge uma nova tela ao centro da página. Nela você cadastra as informações que são necessitadas, qualquer campo obrigatório que não for preenchido o processo de inclusão será travado a partir de uma mensagem de erro e identifica os campos em vermelho para facilitar onde foi afetado o erro. Para alterar algum dado, basta selecionar a informação na tela e apertar o botão alterar para modificar os dados, quando for realizada a alteração, a informação atualiza imediatamente na *grid*.

Figura 64 – Tela de Cadastro do Prestador de Serviço.

Fonte: Elaborado pelos autores (2008).

#### 4.2.12 Tela de Controle do Tipo de Serviço

Quando efetuado um duplo clique no prestador de serviço ou o prestador selecionado e clicado na segunda aba carrega os detalhes do prestador de serviço e os tipos de serviços que o mesmo pode executar.

Nessa tela podem ser executadas as seguintes tarefas: A inclusão do tipo de serviço que o prestador pode executar em sua clínica e a exclusão do tipo de serviço que está cadastrado no prestador de serviço.

PLANO DE SAÚDE PRESTADOR DE SERVIÇOS PROPRIETÁRIO PACIENTE FECHAR

PRESTADOR DE SERVIÇOS FECHAR SUB-MENU

Prestador de Serviço Cadastro de Tipo de Serviço

Id	Nome/Razão Social	Data de Cadastro	Tipo Pessoa	Status
1	Sandro Hostfield	31/05/2008	Pessoa Física	Ativo

Telefone de Contato E-mail de contato Veterinário Responsável

(47) - 3687 - 0000 m@m.com Dr. Rangerdelpt Corais

Serviços que o prestador pode realizar pode atender.

Id	Nome	Descrição	Carência(dias)	Status
1	Sutura Superficial	Sutura Superficial	15	Ativo

Excluir Incluir

Figura 65 – Tela de Controle de Tipo de Serviço.

Fonte: Elaborado pelos autores (2008).

#### 4.2.13 Tela de Inclusão de Serviço ao Prestador de Serviço

Após clicar no botão incluir, o usuário pode executar um filtro de serviço, quando ele clica em buscar, o filtro é carregado com os parâmetros selecionados pelo usuário, efetuando a busca e carregado na grid de tipo de serviços. Com isso, o usuário seleciona um tipo de serviço nessa grid e, após clicado no botão selecionar, o tipo de serviço é cadastrado junto ao prestador de serviço.



Figura 66 – Tela de Inclusão de Serviço ao Prestador de Serviço.

Fonte: Elaborado pelos autores (2008).

### 4.3 PROBLEMAS ENCONTRADOS NO DESENVOLVIMENTO

O primeiro problema encontrado foi a falta de conhecimento sobre o processo de gerência de um plano de saúde. Isto fez com que a equipe tivesse muito re-trabalho, o uso do jogo de cartas, por exemplo, foi executado duas vezes.

Um segundo problema surgiu relacionado à tecnologia. O uso de uma ferramenta de certa forma inovadora, lançada há pouco tempo no mercado, transformou-se em um problema. Pois os desenvolvedores não a conheciam, tornando uma dificuldade o processo de aprendizagem da tecnologia por falta de material para estudo. O uso da internet foi fundamental na busca de soluções para os problemas. Muitos testes foram realizados, e muito código foi desprezado.

### 4.4 VALIDAÇÃO DO PROTÓTIPO

A entrevista foi feita a partir da apresentação do sistema e solicitação de execução das seguintes tarefas:



- Cadastrar tipo de serviço.
- Cadastro de plano de saúde.
- Incluir tipo de serviço ao plano.
- Cadastro de prestador de serviço.
- Incluir tipo de serviço ao prestador de serviço.
- Cadastro proprietário.
- Incluir paciente ao proprietário.

A validação foi realizada com o usuário final. O objetivo foi à consideração do usuário sobre sugestões e críticas. Houve sugestões na hora de filtrar ou buscar alguma informação. Como, por exemplo, ao efetuar uma consulta, digitando algo nos campos para filtro, deve-se clicando o botão “Enter” do teclado para realizar a busca de dados. Outro tópico foi a solicitação de uma consulta no campo nome por qualquer intervalo de dados cadastrados e não somente pelo início. Na tela (figura 56), ao realizar a busca de serviços para ser adicionado ao plano de saúde, solicitou-se apenas clicar em serviço, adicionando, automaticamente, o plano.

O usuário final sugeriu as seguintes alterações:

- Na tela de cadastro do prestador de serviço, o usuário sugeriu que, ao selecionar pessoa física, ao terminar de preencher o campo nome e passar para a próxima etapa, o campo veterinário responsável seja preenchido, respectivamente.
- Observou-se que algumas telas de cadastro ainda não estavam abrindo de forma centralizada, desconfigurando-as.

De forma geral, o usuário final considerou o sistema fácil e amigável.

## 4.5 CONCLUSÕES RELACIONADAS AO USO DA MÉTODOLOGIA

O uso integrado das metodologias permitiu uma visão diferenciada do processo, mais voltada ao usuário final. Observou-se no uso da metodologia, a necessidade do refinamento de algumas técnicas:

- O jogo de cartas deve ser usado de forma diferenciada, em um primeiro momento utilizando-se para a definição dos atributos e conseqüentemente a atividade estará integrada ao Modelo de domínio.
- Finalizada a solicitação de classificação das cartas, serão definidas novas cartas com o nome das classes de domínio que o usuário deve novamente organizar.
- Na segunda etapa do jogo as cartas serão usadas para a classificação e estruturação dos casos de uso sendo parte do modelo de casos de uso.
- O uso do cenário deve ser específico para casos complexos ou casos que devem ser apresentados para o usuário final.
- O jogo de cartas deve ser retomado na fase dos diagramas de classe, validando o modelo final junto ao usuário alvo.
- Aplicação do questionário de satisfação na ultima fase, classe de domínio junto ao usuário alvo.
- Observou-se de forma geral que a integração dos modelos é possível, mas deve sofrer uma validação mais apurada a fim de eliminar redundâncias e aumentar o tempo de execução de cada fase.

## 4.6 CONSIDERAÇÕES FINAIS

Observou-se, nessa etapa, a aplicação ode conceitos introduzidos no uso relacionado no desenvolvimento de software. O sistema apresentou-se como uma boa solução, sendo confirmado por seu usuário final.

## 5. CONCLUSÕES E TRABALHOS FUTUROS

Com este trabalho, foi possível alcançar os objetivos, de conhecer e aplicar as metodologias e ferramentas, até então, desconhecidas.

A integração entre o método de design centrado no usuário e a metodologia ICONIX tiveram suas expectativas alcançadas, pois o usuário final teve uma integração durante o desenvolvimento. As alterações que seriam realizadas somente no final do projeto, alterando de forma negativa o cronograma e o custo do projeto, não aconteceu, o usuário percebeu várias falhas durante o desenvolvimento que foram corrigidas durante o processo.

Tivemos grandes dificuldades no decorrer do projeto. A falta de conhecimento da tecnologia e o método design centrado no usuário geraram grandes dificuldades em sua aplicação. A primeira dificuldade foi a descrição dos requisitos para o desenvolvimento. Com a aplicação do jogo de cartas, foram definidas várias tarefas. Após um mês de modelagem, o usuário final percebeu que não estava de acordo com a visão das tarefas aplicadas na interface. Percebendo isso, os desenvolvedores realizaram uma reunião para refinar as tarefas apresentadas. Com essa reunião, pôde-se esclarecer as tarefas que deveriam de ser aplicadas no sistema, gerando uma alteração na modelagem do projeto. O segundo problema foi a tecnologia escolhida. Com a falta do conhecimento, em tecnologia FLEX, perdeu-se muito tempo no desenvolvimento. Várias tentativas foram desanimadoras para o grupo de desenvolvedores, mas, com o apoio de terceiros, o projeto foi concluído.

A tecnologia aplicada respondeu suas expectativas satisfatoriamente, pois a integração entre Java e Flex. teve um dinamismo compensador.

O desenvolvimento do projeto apresenta-se na monografia em três módulos: Módulo plano saúde, módulo prestador de serviço e módulo proprietário. A modelagem por completa foi realizada somente no módulo plano de saúde. Os demais módulos foram modelados apenas até os casos de uso.

Observou-se que a hipótese de integração do método Centrado no Usuário e o Iconix é possível, viável e compensador. Facilita o reconhecimento de requisitos e a adaptação das necessidades do usuário à interface. Observou-se que a metodologia usada, híbrida, pode ser

ainda mais trabalhada, intensificando a integração, o que deve ser estudado em trabalhos futuros.

Algumas perspectivas futuras se delineiam para este projeto relacionadas ao desenvolvimento do sistema, são elas:

- Concluir a modelagem restante dos módulos prestador e proprietário.
- Desenvolver o módulo prestador de serviço, em que o mesmo gerenciará a parte de consultas e serviços a serem prestados ao paciente.
- Desenvolver o módulo proprietário em que o proprietário poderá ter um conhecimento sobre as consultas que foram realizadas em animal de estimação (paciente).
- Desenvolver um site com o intuito de propagar o plano de saúde.

## REFERÊNCIAS

ADOBE, Flex Builder. Disponível em: <<http://www.adobe.com/products/flex>>. Acesso em 16 de mai. 2008.

AGNER, Luiz Carlos e MORAES, Ana Maria de. **Design Centrado no Usuário e Diálogo Clientes-Organizações através de Interfaces na Web**. Disponível em: <<http://www.senac.br/informativo/BTS/281/boltec281c.htm>>. Acesso em 08 de set. 2007.

ANTONIO FERREIRA, José Sérgio. **Revista pet shop Brasil Business**. Disponível em:<<http://www.arteditora.com.br/petshopbusiness/banco1.asp?Codigo=4>>. Acesso em 11 de ago. 2007.

BLAHA, Michael e RUMBAUGH,James. **Modelagem e Projetos Baseados em Objetos com UML 2.0**. 2 ed. Rio de janeiro:Editora Campus. 2006.

BONA, Cristina. **Avaliação de processos de software: um estudo de caso em XP e Iconix**. 2002.122 f. Dissertação (Pós-Graduação em Engenharia de Produção) - Universidade Federal de Santa Catarina, Florianópolis, 2002.

BONATTO, Klayson Sesana. DBDesigner - Software livre para modelagem de dados. Disponível em: <<http://www.dicas-l.com.br/dicas-l/20030922.php>>. Acesso em 16 mai. 2008.

CYBIS, Walter de Abreu, **Engenharia de usabilidade: uma abordagem ergonômica**, 1 ed. Florianópolis, 2003..

CYSNEIROS, Luiz Marcio e LEITE, Júlio César Sampaio do Prado. **Utilizando Requisitos Não Funcionais para Análise de Modelos Orientados a Dados**. Disponível em: <[http://wer.inf.puc-rio.br/WERpapers/artigos/artigos\\_WER98/cysneiros.pdf](http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER98/cysneiros.pdf)>. Acesso em 13 de out. 2007.

D'ÀVILA, Márcio. **Tutorial Tomcat - Instalação e Configuração Básica**. Disponível em: <<http://www.mhavila.com.br/topicos/java/tomcat.html>>. Acesso em 16 de mai. 2008.

DEITEL, H. M. e DEITEL, P. J. **Java Como Programar**. Trad. Edson Furnankiewicz, 3 ed. Porto Alegre:Bookman,2001.

DEHAAN, Jen. **FLASH MX 2004 Guia Autorizado Macromedia**. Trad. Marcos Vieira, Rio de Janeiro: Elsevier, 2004.

IMASTERS. **O que é o DBDesigner?**. Disponível em: <<http://imasters.uol.com.br/faq/897>>. Acesso em 16 de mai. 2008.

INFOWESTER. **Banco de dados MySQL e PostgreSQL**. Disponível em: <<http://www.infowester.com/postgresql.php>>. Acesso em 17 mai. 2008.

JAVAFREE. **Tutorial Java: O que é Java?**. Disponível em: <<http://www.javafree.org/content/view.jf?idContent=84>>. Acesso em 16 de mai. 2008.

LOCASITES, Internet Services. **Corel Draw**. Disponível em: <[http://www.locasite.com.br/design/tec\\_corel.shtml](http://www.locasite.com.br/design/tec_corel.shtml)>. Acesso em 17 mai. 2008.

MENDES, Daniel Nóbrega, **Design Centrado no Usuário: filosofia e necessidade**. Disponível em: <<http://www.intranetportal.com.br/colab1/col5>>. Acesso em 08 de set. 2007.

MICROSOFT. **Sistema Operacional Windows**. Disponível em: <<http://www.microsoft.com>>. Acesso em 17 mai. 2008.

MUNDO JAVA: **Eclipse 3 dicas e truques**. Paraná: Ed.2005, número 9, ano II.

ORLANDINI, Leandro. **A importância dos Sistemas de Informação**. Administração e Tecnologia. Disponível em: <[http://www.bonde.com.br/colunistas/colunistasd.php?id\\_artigo=1646](http://www.bonde.com.br/colunistas/colunistasd.php?id_artigo=1646)>. Acesso em 06 de ago. 2007.

PINHO, Carlos e CASTELHANO, Ricardo, **Criar aplicações ricas para internet interativas e compatíveis com várias plataformas**. Disponível em: <<http://www.garotosdeweb.com/index.php/tutoriais/tutoriais-flex/tutorial-porque-o-adobe-flex/>>. Acesso em 16 de mai. 2008.

PRESMAN Roger S. **Engenharia de Software**. 6 ed. São Paulo: McGraw-Hill, 2006.

RATIONAL, Corporation, **Rational Unified Process**. ed.2001, Disponível em: <<http://www.wthreex.com/rup/>>. Acesso em 08 de set. 2007.

RIBEIRO, Cassandra de Oliveira e Silva. **Bases pedagógicas e ergonômicas para a concepção e avaliação de produtos educacionais informatizados**. Dissertação (Pós-Graduação em Engenharia de Produção) - Universidade Federal de Santa Catarina, Florianópolis, 1998.

SILVA, Alberto. **Análise e Concepção de Sistemas de Informação**. Metodologia Iconix. Disponível em: <<https://dspace.ist.utl.pt/bitstream/2295/74372/1/11-acsi-metodologia-iconix-jlb.ppt>> Acesso em 16 de mar. 2007.

SOARES, Michel dos Santos, **Metodologias Ágeis Extreme Programming e Scrum para o Desenvolvimento de Software**. Universidade Presidente Antônio Carlos, BR 482 Km 3, Gigante, Conselheiro Lafaiete, Disponível em: <<http://www.inf.ufsc.br/resi/edicao04/artigo06.pdf>>. Acesso em 06 de nov. 2007.

SOARES, Wallace, **MySQL Conceitos e Aplicações**. 1 ed. São Paulo:Eda. Érica Ltda, 2001.

SOMMERVILLE Ian. **Engenharia de Software**. 6 ed. São Paulo: Addison Wesley, 2003.

SOUZA, Clarisse Sieckenius, et al. **Projeto de Interfaces de Usuário: Perspectivas Cognitivas e Semióticas**. Disponível em: <[http://www.dimap.ufrn.br/~jair/piu/JAI\\_Apostila.pdf](http://www.dimap.ufrn.br/~jair/piu/JAI_Apostila.pdf)>. Acesso em 08 de nov. 2007.

SPARX SYSTEMS, **Enterprise Architect**. Disponível em: <<http://www.sparxsystems.com.au/products/ea/index.html>> Acesso em 04 jun. de 2008.

ZANLORENCI, Edna Pacheco e BURNETT, Robert Carlisle, **Requisitos funcionais e não-funcionais, as duas faces da moeda aplicáveis à engenharia de software**. Disponível em: <<http://www.pr.gov.br/batebyte/edicoes/2001/bb115/requisitos.htm>>. Acesso em 13 de set. 2007.

ZINDEL, José Eduardo, **Breve introdução aos Diagramas UML**. Disponível em: <<http://www.voxxel.com.br/pages/introdiauml.html#classes>>. Acesso em 10 de jun. 2008.