

UNIVERSIDADE ESTADUAL DE MONTES CLAROS - UNIMONTES
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS - CCET
DEPARTAMENTO DE CIÊNCIAS DA COMPUTAÇÃO - DCC
CURSO DE SISTEMAS DE INFORMAÇÃO

**SISTEMA DE INFORMAÇÃO PARA CONTROLE DE
PROCESSOS DE UMA COOPERATIVA DE CRÉDITO**

ADRIANO ANTUNES PRATES

MONTES CLAROS
2008

ADRIANO ANTUNES PRATES

**SISTEMA DE INFORMAÇÃO PARA CONTROLE DE
PROCESSOS DE UMA COOPERATIVA DE CRÉDITO**

Projeto Orientado de Conclusão de Curso apresentado ao Departamento de Ciências da Computação da Universidade Estadual de Montes Claros, como requisito parcial para a conclusão do curso de Sistemas de Informação, orientado pelo professor Dr. Nilton Alves Maia.

**MONTES CLAROS
2008**

ADRIANO ANTUNES PRATES

**SISTEMA DE INFORMAÇÃO PARA CONTROLE DE PROCESSOS DE
UMA COOPERATIVA DE CRÉDITO.**

Aprovada em de Junho de 2008.

Prof. Dr. Nilton Alves Maia – Orientador.
DCC - Departamento de Ciências da
Computação.
Universidade Estadual de Montes Claros.

Prof.(a) Sônia Beatriz de Oliveira e Silva
Maia.
DCC - Departamento de Ciências da
Computação.
Universidade Estadual de Montes Claros.

Prof. Frederico Bida de Oliveira
DCC – Departamento de Ciências da
Computação.
Universidade Estadual de Montes Claros.

**MONTES CLAROS
2008**

Dedico este trabalho a minha querida
amiga, companheira e namorada
Elisângela Fernandes Carneiro.

AGRADECIMENTOS

Agradeço, em primeiro lugar, ao nosso Senhor, por todas as oportunidades oferecidas na minha vida. A minha família e amigos pelo apoio incondicional. Ao meu orientador, professor Dr. Nilton Alves Maia pela orientação sempre oportuna, esclarecedora e inteligente, fazendo com que a realização deste trabalho fosse possível. Aos meus queridos colegas por toda a ajuda.

RESUMO

O trabalho teve como objetivos a análise e o desenvolvimento de um sistema de informação para o controle dos processos internos de uma empresa do segmento cooperativista de crédito. Para tanto, foram realizados estudos das principais técnicas de modelagem e desenvolvimento. O processo foi realizado em três fases: especificação de requisitos, análise de sistema e implementação do software, através das técnicas de programação orientada a objetos e linguagem de modelagem unificada – UML. O resultado obtido através deste trabalho foi o desenvolvimento de um sistema de informação para o controle dos processos de venda de produtos e serviços da empresa cooperativa de crédito cliente.

Palavras Chaves: Sistema, informação, controle de processos, software, cooperativa, crédito, informatização, UML, objetos, SASC.

LISTA DE QUADROS

Quadro 1 – Multiplicidades.....	19
Quadro 2 – Módulos e Funções do SASC.....	24
Quadro 3 – Benefícios do SASC.....	26
Quadro 4 – Descrição dos Atores.....	27
Quadro 5 – Características dos Atores.....	28
Quadro 6 – Descrição do Caso de uso “Efetuar Login”	31
Quadro 7 – Descrição do Caso de uso “Cadastrar Agencia”	31
Quadro 8 – Descrição do Caso de uso “Cadastrar Angariador”	31
Quadro 9 – Descrição do Caso de uso “Cadastrar Usuário”	31
Quadro 10 – Descrição do Caso de uso “Excluir Usuário”	32
Quadro 11 – Descrição do Caso de uso “Cadastrar Plano Saúde”	33
Quadro 12 – Descrição do Caso de uso “Alterar Plano Saúde”	34
Quadro 13 – Descrição do Caso de uso “Cancelar Plano Saúde”	35
Quadro 14 – Descrição do Caso de uso “Processar Fechamento Fatura”	35
Quadro 15 – Descrição do Caso de uso “Atualizar Saldo das C/Cs”	35
Quadro 16 – Descrição do Caso de uso “Cadastrar Seguro Vida”	37
Quadro 17 – Descrição do Caso de uso “Renovar Seguro”	37
Quadro 18 – Descrição do Caso de uso “Cadastrar Associado”	38
Quadro 19 – Descrição do Caso de uso “Importar Tarifas do Dia”	39
Quadro 20 – Descrição do Caso de uso “Atualizar Saldo das C/Cs”	40
Quadro 21 – Descrição do Caso de uso “Pagar Comissões”	41
Quadro 22 – Descrição do Caso de uso “Cadastrar Comissão”	41
Quadro 23 – Definição da Classe “Associado”	43
Quadro 24 – Definição da Classe “Agência”	43
Quadro 25 – Definição da Classe “Angariador”	44
Quadro 26 – Definição da Classe “Seguros”	44
Quadro 27 – Definição da Classe “Seguro_Não_Nominado”	45
Quadro 28 – Definição da Classe “Seguro_Veículos”	45
Quadro 29 – Definição da Classe “Seguro_Equipamento”	46
Quadro 30 – Definição da Classe “Seguro_Vida”	46
Quadro 31 – Definição da Classe “Seguro_Residência”	47

Quadro 32 – Definição da Classe “Plano_Saúde”	47
Quadro 33 – Definição da Classe “Dependente”	48
Quadro 34 – Definição da Classe “Comissão”	49
Quadro 35 – Definição da Classe “Tarifas”	49
Quadro 36 – Definição da Classe “Cheque_Devolvido”	50
Quadro 37 – Definição da Classe “ContaCorrente”	50

LISTA DE FIGURAS

Figura 1 – Exemplo de caso de uso.....	16
Figura 2 – Notação UML para objetos.....	17
Figura 3 – Representações da UML para classes.....	17
Figura 4 – Associações da UML.....	19
Figura 5 – Passagem unidirecional por uma associação.....	19
Figura 6 – Classe de associação.....	20
Figura 7 – Agregação e composição.....	20
Figura 8 – Generalização e especialização.....	20
Figura 9 – Dependência entre classes.....	21
Figura 10 – Exemplo de diagrama de seqüência.....	22
Figura 11 – Diagrama de contexto.....	29
Figura 12 – Caso de uso do módulo “Administração do Sistema”.....	30
Figura 13 – Caso de uso do módulo “Plano de Saúde”.....	33
Figura 14 – Caso de uso do módulo “Seguros”.....	36
Figura 15 – Caso de uso do módulo “Associados”.....	38
Figura 16 – Caso de uso do módulo “Tarifas”.....	39
Figura 17 – Caso de uso do módulo “Comissões”.....	41
Figura 18 – Diagrama de Classes.....	42
Figura 19 – Diagrama de Seqüência do módulo “Plano de Saúde”.....	51
Figura 20 – Diagrama de Seqüência do módulo “Plano de Saúde”.....	52
Figura 21 – Diagrama de Seqüência do módulo “Seguros”.....	53
Figura 22 – Diagrama de Seqüência do módulo “Tarifas”.....	54
Figura 23 – Diagrama de Estado do módulo “Plano de Saúde”.....	55
Figura 24 – Diagrama de Estado do módulo “Seguros”.....	56
Figura 25 – Diagrama de Estado do módulo “Comissões”.....	57
Figura 26 – Diagrama de Estado do módulo “Administração do Sistema”.....	57
Figura 27 – Diagrama de Estado do módulo “Associados”.....	58
Figura 28 – Tela de Login do SASC.....	60
Figura 29 – Janela principal do SASC.....	60
Figura 30 – Janela para impressão do “Perfil do Associado”.....	61
Figura 31 – Janela para seleção de Associados.....	62

Figura 32 – Janela para consulta de Seguros.....	62
Figura 33 – Consulta seguro “não-nominado”	63
Figura 34 – Formulário para consulta de Plano de Saúde.....	64
Figura 35 – Consulta de Plano de Saúde cadastrado.....	65
Figura 36 – Formulário para consulta de Tarifas.....	66
Figura 37 – Lançamento manual de crédito.....	66
Figura 38 – Formulário para baixa/inclusão de tarifas em atraso.....	67
Figura 39 – Formulário para gerenciamento de comissões.....	67
Figura 40 – Formulário para consulta de comissão.....	68
Figura 41 – Formulário para cadastramento de associado.....	68
Figura 42 – Formulário para cadastramento de usuários.....	69
Figura 43 – Modelo padrão de relatórios do SASC.....	69

SUMÁRIO

1	INTRODUÇÃO.....	13
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	FUNDAMENTOS DA PROGRAMAÇÃO ORIENTADA A OBJETOS.....	16
2.1.1	Classes.....	17
2.1.2	Objetos	17
2.1.3	Encapsulamento.....	18
2.1.4	Mensagens	18
2.1.5	Herança / Generalização	18
2.1.6	Polimorfismo.....	19
2.2	BENEFÍCIOS DA TECNOLOGIA ORIENTADA A OBJETOS	19
2.3	LINGUAGEM DE MODELAGEM UNIFICADA (UML).....	22
2.3.1	Diagramas de Caso de Uso	23
2.3.2	Diagrama de Classe	24
2.3.3	Diagramas de Seqüência	29
2.3.4	Diagrama de Estado	30
3	ESPECIFICAÇÃO DE REQUISITOS DE SOFTWARE	32
3.1	MISSÃO DO PRODUTO	32
3.2	TABELA DE FUNÇÕES / MÓDULOS DO SISTEMA	33
3.3	REQUISITOS DE QUALIDADE.....	34
3.4	LIMITES	35
3.5	BENEFÍCIOS.....	35
3.6	ATOES.....	36
3.7	DIAGRAMA DE CONTEXTO.....	38
4	ANÁLISE DO SISTEMA	39
4.1	DIAGRAMAS DE CASO DE USO	39
4.1.1	Módulo Administração do Sistema.....	39
4.1.2	Módulo Planos de Saúde.....	41
4.1.3	Módulo Seguros.....	45
4.1.4	Módulo Associados.....	46
4.1.5	Módulo Tarifas	48
4.1.6	Módulo Comissões	49
4.2	DIAGRAMA DE CLASSES.....	51

4.3	DIAGRAMAS DE SEQUÊNCIA.....	60
4.4	DIAGRAMAS DE ESTADO	63
4.4.1	Módulo Plano de Saúde.....	63
4.4.2	Módulo Seguros.....	64
4.4.3	Módulo Comissões.....	65
4.4.4	Módulo Administração do Sistema.....	66
4.4.5	Módulo Associados.....	67
5	IMPLEMENTAÇÃO DO SISTEMA.....	68
5.1	TECNOLOGIAS ADOTADAS	68
5.2	DESCRIÇÃO DO SISTEMA	68
5.3	PADRONIZAÇÃO DE RELATÓRIOS.....	78
6	CONCLUSÕES.....	79
6.1	RESULTADOS OBTIDOS	79
6.2	DIFICULDADES ENCONTRADAS.....	80
6.3	TRABALHOS FUTUROS	80
ANEXO A	74

1 INTRODUÇÃO

A Empresa X é uma cooperativa de crédito que proporciona aos seus 12 mil associados uma gama de variedades de produtos e serviços bancários, e por estar presente em mais de quinze municípios norte-mineiros, é um importante agente de desenvolvimento regional.

A utilização eficiente da informação é parte indispensável da doutrina e prática cotidiana da Cooperativa, sendo componente vital ao êxito de suas atividades.

O desenvolvimento de um sistema de informação para administrar os produtos e serviços oferecidos por esta instituição financeira é algo imprescindível para expandir sua atuação de maneira satisfatória. Somente com o auxílio da modelagem pode-se visualizar e controlar o desenvolvimento de sistemas de maneira eficaz, identificando e gerenciando riscos, estipulando e cumprindo prazos, dentro das estimativas de custo (SEABRA JR, 2001).

A principal vantagem proporcionada pela tecnologia aos Sistemas de Informação é a capacidade de processar um gigantesco número de dados simultaneamente, tornando a disponibilização das informações demandadas, praticamente on-line. Mas de pouco adianta esse potencial se os sistemas (rotinas, processos, métodos) não estiverem muito bem coordenados e analisados. Informatizar sistemas ruins traz novos problemas e nenhuma solução, além de nublar as possíveis causas dessas falhas (ORLANDINI, 2007).

Os Sistemas de Informação são peças fundamentais para as empresas, não apenas na elaboração de relatórios, mas fazem parte de todos os departamentos e atividades da companhia, desde o simples controle até a confecção de planos estratégicos complexos (ORLANDINI, 2007).

Uma implementação bem feita faz com que um sistema de informação seja espelho de todas as atividades realizadas dentro de uma empresa, mas é importante ressaltar que esse sucesso depende de um levantamento característico, eficiente, e principalmente, sucinto dos requisitos.

O tema deste trabalho é a análise e desenvolvimento de um sistema de informação para controle e administração dos processos da cooperativa de crédito

aqui denominada Empresa X, com utilização das técnicas de POO (Programação Orientada a Objetos) e UML (Linguagem de Modelagem Unificada).

O problema a ser abordado pelo projeto é a inexistência de um sistema de informação adequado para solucionar a deficiência no controle de vários processos da cooperativa, como exemplo: a venda dos produtos e serviços oferecidos pela Empresa X. Dessa forma, pode-se propor a seguinte questão: “De que forma, um sistema de informação poderá contribuir para a melhor gestão e controle dos processos da Empresa X?”.

As Cooperativas de Crédito atuam no sistema financeiro nacional com um papel bem definido com base em normas ditadas pelo Banco Central do Brasil, oferecendo aos seus associados vários produtos e serviços bancários e não bancários. “São instituições financeiras privadas, com personalidade jurídica própria, especializadas em propiciar crédito e prestar serviço a seus associados, constituídas sob a forma de sociedade de pessoas de natureza civil” (NIYAMA, 2002).

A Empresa X é uma cooperativa de crédito que há 21 anos atua como intermediária financeira entre o associado e diversas instituições com vistas ao desenvolvimento sustentável regional.

A cooperativa enfrenta grandes desafios a serem superados, como melhorar a produtividade, mediante adequação dos processos, agilizar os atendimentos e serviços prestados, disponibilizar informações rápidas e confiáveis para as tomadas de decisões, melhorar o nível de atendimento dos associados, melhorar seu desempenho e conseqüentemente, seus resultados.

Nesse contexto, a implantação de um sistema informatizado poderia resultar em uma administração mais eficiente visando qualidade e produtividade, onde permitirá que todos os funcionários, desde a diretoria até a linha de produção possam fazer consultas relacionadas à sua alçada.

Diante disso, o objetivo deste trabalho é o desenvolvimento de um sistema de informação para a empresa X, responsável pelo controle de processos internos além da administração e controle dos produtos e serviços oferecidos por esta, bem como servir de ferramenta de auxílio à tomada de decisão gerencial.

Os objetivos específicos são:

- Estudo das técnicas de programação orientada a objetos, com utilização da linguagem de modelagem unificada - UML;

- Realização de entrevistas e questionários com objetivo de conhecer os requisitos necessários ao desenvolvimento do sistema;
- Realização da análise e modelagem de um sistema de informação com base nas técnicas da UML.
- Desenvolvimento e implantação do sistema de informação.

O trabalho foi desenvolvido em três fases. A primeira fase consistiu na realização de entrevistas, através da aplicação de questionários na empresa em estudo, com objetivo de se conhecer as reais necessidades desta. Após a aplicação dos formulários foi feita uma análise dos dados obtidos e então foram gerados relatórios, que serviram de base para a modelagem e desenvolvimento do software.

A segunda fase deste projeto consistiu na modelagem do sistema, utilizando a linguagem de modelagem unificada - UML. A modelagem é a parte central de todas as atividades que levam à implantação de um bom sistema (JACOBSON, 2000).

A fase final do projeto consistiu no desenvolvimento do **SASC - Sistema de Apoio Cooperativa X**, através das técnicas de programação orientada a objetos, seguindo a modelagem realizada na fase anterior.

Este documento está organizado da seguinte forma: o capítulo 2 trata da fundamentação teórica utilizada para desenvolvimento deste trabalho, o capítulo 3 aborda a especificação de requisitos de software, detalhando um conjunto de requisitos estabelecidos em entrevistas e questionários e que devem ser satisfeitos pela solução implementável do problema abordado. O capítulo 4 mostra a análise de sistema de software baseada na tecnologia Orientada a Objetos e expressa na notação UML. O capítulo 5 descreve o software SASC, mostrando suas telas e as tecnologias adotadas para o desenvolvimento deste e, finalmente, o capítulo 6 conclui o trabalho, apresentando os resultados obtidos, dificuldades encontradas durante sua realização e os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 FUNDAMENTOS DA PROGRAMAÇÃO ORIENTADA A OBJETOS

Segundo Martin & Odell (1995, p.195):

Um analista orientado a objeto vê o mundo como objetos com estruturas de dados e métodos, e eventos que disparam as operações que mudam o estado dos objetos. As operações ocorrem quando os objetos fazem solicitações a outros objetos. O analista cria diagramas das estruturas do objeto e dos eventos que mudam os objetos.

Segundo Lee & Tapfenhart (2001), a programação orientada a objetos é um paradigma de programação avançado. Paradigma de programação pode ser definido como um “conjunto de teorias, padrões e métodos que, juntos, representam um modo de organizar conhecimento” (LEE & TAPFENHART, 2001, p. 24).

Até então, o paradigma de programação utilizado pela maioria dos programadores era a programação imperativa, ou método estruturado. Nos dias atuais, em que as linguagens mais conceituadas como Java e C++ são mais utilizadas, o paradigma orientado a objetos é utilizado para modelar a realidade.

Segundo Lee & Tapfenhart (2001), para um compilador, a diferença entre uma linguagem imperativa como C e uma linguagem orientada a objeto como C++ é apenas a adição de algumas palavras-chave e tipos de dados. Entretanto, efetuar uma utilização eficiente dessas facilidades requer que os desenvolvedores mudem suas percepções para um enfoque totalmente diferente para fins de modelagem e resolução de problemas. A POO é uma nova maneira de pensar sobre o que significa fazer cálculos, como organizamos nossas informações dentro de um sistema de computador e de que forma descrevemos nossa visão (modelo) da realidade.

2.1.1 Classes

“Uma classe é um gabarito para a definição de objetos. Através da definição de uma classe, descreve-se que propriedades – ou atributos – o objeto terá” (RICARTE, 2001).

O termo *classe* refere-se à implementação de software de um tipo de objeto. Tipo de objeto é uma noção conceitual. Ela especifica uma família de objetos sem estipular como o tipo de o objeto são implementados. Os tipos de objetos são especificados durante a análise OO.

“Uma classe é uma implementação de um tipo de objeto. Ela especifica uma estrutura de dados e os métodos permissíveis que se aplicam a cada um de seus objetos” (MARTIN & ODELL, 1995).

Uma classe é composta pelo **nome da classe**, que é seu identificador, e necessária, por exemplo, no momento da criação de um objeto. O conjunto de **atributos** descreve as propriedades da classe. Cada atributo é identificado por um nome e tem um tipo associado. Em uma linguagem de programação orientada a objetos pura, o tipo é o nome de uma classe. Na prática, a maior parte das linguagens de programação orientada a objetos oferecem um grupo de tipos primitivos, como inteiro, real e caracter. Os **métodos** definem as funcionalidades e operações da classe, ou seja, o que será possível fazer com objetos dessa classe.

2.1.2 Objetos

“Um objeto é um elemento que representa, no domínio da solução, alguma entidade (abstrata ou concreta) do domínio de interesse do problema sob análise. Objetos similares são agrupados em classes” (RICARTE, 2001).

Um objeto é uma instância de uma classe, que consiste de atributos (dados) e métodos desta classe. O acesso aos dados do objeto é conferido somente por meio dos métodos providos pelo objeto.

2.1.3 Encapsulamento

Segundo Lee & Tapfenhart (2001), o ato de empacotar ao mesmo tempo dados e métodos é denominado *encapsulamento*. O objeto esconde seus dados de outros objetos e permite que os dados sejam acessados por intermédio de seus próprios métodos. O encapsulamento protege os dados de um objeto contra adulteração.

O encapsulamento é importante porque separa a maneira como um objeto se comporta da maneira como ele é implementado. Isso permite que as implementações do objeto sejam modificadas sem exigir que os aplicativos que as usam sejam também modificadas.

2.1.4 Mensagens

“Um objeto pode comunicar-se com outro objeto unicamente via mecanismo de passagem de mensagens” (LEE & TEPFENHART, 2001).

Para fazermos um objeto realizar alguma coisa, enviamos uma *solicitação*. Esta faz com que uma operação seja invocada pela classe deste objeto. A operação executa o método apropriado e, opcionalmente, retorna uma resposta. A mensagem que constitui a solicitação contém o nome do objeto, o nome da operação, e às vezes, um grupo de parâmetros.

2.1.5 Herança / Generalização

Segundo Lee & Tapfenhart (2001), as classes podem ser organizadas utilizando-se uma estrutura de herança hierárquica. Nessa estrutura, a subclasse herdar os atributos, os relacionamentos e os métodos da superclasse, que estão em um ponto mais alto da árvore. Uma superclasse abstrata é uma classe utilizada para criar somente subclasses. Portanto não há instâncias diretas dessa classe.

Entretanto, para uma regra há sempre exceções. Com o intuito de tratar das exceções à regra dentro de nossa estrutura hierárquica, definimos o princípio de

polimorfismo. Existe também o conceito de herança múltipla, na qual uma subclasse pode herdar as estruturas de mais de uma classe.

2.1.6 Polimorfismo

Segundo Lee & Tapfenhart (2001), uma subclasse criada a partir da técnica de herança, poderá criar seu próprio método para substituir um método de qualquer uma de suas superclasses ao prover um serviço que seja disponível no nível da superclasse quando uma instância daquela subclasse é o agente.

Para a subclasse, o método dela irá *ignorar* o método da superclasse para prestar o mesmo serviço.

Muito embora a generalização seja um conceito poderoso, há relacionamentos entre objetos que não podem ser capturados utilizando-se esse conceito.

2.2 BENEFÍCIOS DA TECNOLOGIA ORIENTADA A OBJETOS

Segundo Martin & Odell (1995, p.114), os principais benefícios da tecnologia orientada a objetos são:

Reusabilidade: Classes são projetadas para que possam ser reusadas em muitos sistemas. Para maximizar o reuso, classes são construídas de forma que possam ser customizadas. Um repositório deve ser preenchido por uma coleção sempre crescente de classes reutilizáveis. As bibliotecas de classes têm a probabilidade de crescer rapidamente.

Estabilidade: As classes projetadas para repetido reuso tornam-se estáveis da mesma maneira que os microprocessadores e outros chips se tornam-se estáveis. Aplicativos são construídos a partir de chips de software, quando possível.

Encapsulamento: o projetista pensa em termos de comportamento dos objetos, não em detalhes de baixo nível. As classes, graças ao encapsulamento, são caixas pretas; o desenvolvedor usa a caixa-preta e não olha o que há dentro dela. Ele tem de entender o seu comportamento e como comunicar-se com ela.

Classes de complexidade sempre crescente são construídas: Classes são sempre construídas de classes, as quais são por sua vez, são constituídas de outras classes. Isso possibilita que complexos componentes de software sejam construídos, os quais se tornam, eles próprios, blocos para um software mais complexo.

Confiabilidade: um software construído a partir de classes estáveis bem comprovadas provavelmente terá um número menor de falhas do que um software idealizado desde os primeiros rascunhos.

Projeto mais rápido: aplicativos são criados a partir de componentes preexistentes. Muitos componentes são construídos de forma que possam ser customizados para um projeto particular. Os componentes podem ser vistos e interligados nas telas de uma ferramenta CASE.

Projeto de qualidade mais elevada: os projetos freqüentemente têm uma qualidade mais elevada porque são construídos a partir de componentes bem comprovados que foram testados e aprimorados repetidamente.

Integridade: as estruturas de dados podem ser usadas apenas com métodos específicos. Isso é particularmente importante com sistemas cliente-servidor e distribuídos em que usuários desconhecidos poderiam tentar acessar o sistema.

Programação mais fácil: os programas são construídos em pequenas peças, cada uma das quais geralmente é mais fácil de ser criada. O programador cria um método para uma classe de cada vez. O método muda o estado dos objetos de uma maneira que usualmente é simples quando considerada em si mesma.

Manutenção mais fácil: o programador de manutenção comumente muda um método de uma classe de cada só vez. Cada classe executa sua operação independentemente de outras classes.

Inventividade: implementadores proficientes com as mais poderosas ferramentas CASE-OO, rodando numa estação de trabalho, descobrem que podem gerar idéias rapidamente. As ferramentas encorajam-nos a criar e implementar rapidamente suas invenções.

Ciclo dinâmico de vida: o alvo do desenvolvimento do sistema muitas vezes muda durante a implementação. Ferramentas CASE-OO tornam as mudanças de ciclo de vida mais fáceis. Isso possibilita aos implementadores atenderem melhor aos usuários finais, adaptarem-se às mudanças no negócios, aprimorarem metas à medida que o sistema chega a uma focalização mais apurada e melhorarem constantemente o projeto durante a implementação.

Refinamento durante a construção: pessoas criativas, tais como escritores e teatrólogos, mudam constantemente o roteiro de sua obra enquanto a estão implementando. Isto leva a resultados finais muito melhores. As ferramentas CASE-OO dão aos construtores de software a capacidade de aprimorarem o projeto à medida que o implementam.

Modelagem mais realística: a análise OO modela a empresa ou a área de aplicação de uma maneira mais próxima da realidade do que a análise convencional. A análise traduz-se diretamente em projeto e implementação.

Melhor comunicação entre profissionais de informática e pessoas de negócios: as pessoas de negocio entendem mais facilmente o paradigma OO. Elas pensam em termos de eventos, objetos e políticas de negócios que descrevam o comportamento dos objetos.

Especificação e projeto declarativos: as especificações e o projeto, construídos com a formalidade das ferramentas CASE, devem ser declarativos sempre que possível – declarando explicitamente o que é necessário. Isso possibilita ao projetista pensar como usuário final em vez de pensar como um computador.

Interoperabilidade: softwares de muitos fornecedores diferentes podem funcionar juntos. A interoperabilidade de software de diversos fornecedores é uma das mais importantes metas dos padrões OO. Softwares desenvolvidos independentemente em lugares distintos devem ser capazes de funcionarem juntos e se apresentarem como uma única unidade ao usuário.

Independência de projeto: as classes são projetadas para ser independentes de plataformas, hardware e ambientes de software. Elas empregam solicitações e respostas de formatos padrões. Isso lhes possibilita serem usadas com múltiplos sistemas operacionais, gerenciadores de banco de dados, gerenciadores de redes, interfaces gráficas, e assim por diante.

Computação cliente-servidor: em sistemas cliente-servidor, as classes de software cliente devem enviar solicitações às classes do software servidor e receber respostas. Uma classe servidora pode ser usada por muitos clientes diferentes. Esses clientes somente podem acessar os dados servidores com os métodos da classe. Portanto, os dados ficam protegidos da adulteração.

Melhores ferramentas CASE: as ferramentas CASE usarão técnicas gráficas para projetar classes e suas interações e para usar objetos existentes adaptados a novas aplicações. As ferramentas devem facilitar a modelagem em termos de eventos,

acionadores, estados do objeto e assim por diante. As ferramentas CASE-OO devem gerar código tão logo as classes sejam definidas e permitir que o projetista use e teste os métodos criados.

2.3 LINGUAGEM DE MODELAGEM UNIFICADA (UML)

A UML (Linguagem de Modelagem Unificada) é uma linguagem de modelagem utilizada para visualizar e documentar todos os recursos que especificamos na análise e desenho de um sistema, bem como, unifica as melhores práticas de engenharia para fins de modelagem de sistemas. “Ela normalmente é utilizada para descrever um sistema de computador da forma como este é percebido em vários pontos durante a análise e desenho” (LEE & TAPFENHART, 2001). A sintaxe foi definida, originalmente, por Jim Rumbaugh e Grady Booch para conter todas as construções disponíveis em cada um de seus próprios métodos, mas com uma representação gráfica comum. Posteriormente, receberam contribuição de Ivar Jacobson, que acrescentou a sintaxe para a definição de requisitos com casos de uso e, depois a linguagem foi adotada por um comitê de especialistas em orientação a objetos.

A sintaxe da UML foi projetada para ser independente de quaisquer linguagens de programação, processo de software ou ferramenta, porém, “é suficientemente genérica e flexível a ponto de poder ser utilizada sob uma forma personalizada usando extensões definidas pelo próprio usuário para acomodar praticamente qualquer linguagem, ferramenta ou requisito de processo”. (LEE & TAPFENHART, 2001).

Embora a sintaxe em si seja bem definida e de razoável compreensão, a sua aplicação a um determinado projeto mostra-se muito menos fácil de definir. Para isso é preciso a definição de um conjunto de regras semânticas apropriadas para uma arquitetura e processos de software específicos.

Os objetivos estabelecidos para a UML são:

- Ser uma linguagem de modelagem visual, expressiva, que se revele relativamente simples e extensível;

- Contar com mecanismos de extensibilidade e especialização para estender, preferencialmente a modificar, os conceitos centrais;
- Ser independente de qualquer linguagem de programação;
- Ser independente de processo;
- Suportar conceitos de alto nível (estrutura, padrões e componentes);
- Tratar de temas complexos arquiteturais recorrentes utilizando os conceitos de alto nível;
- Ser flexível e amplamente aplicável;

A UML define nove tipos de diagrama: de classe, objeto, caso de uso, seqüência, colaboração, estado, atividade, componente e implantação. Em todos os diagramas, os conceitos são representados como símbolos, e os relacionamentos entre conceitos são representados como trajetórias (linhas) conectando os símbolos. Cada um desses elementos terá um nome específico.

2.3.1 Diagramas de Caso de Uso

Segundo Pádua Filho (2000, p.99), os casos de uso descrevem o comportamento esperado do produto como um todo. Os diagramas de caso de uso descrevem os relacionamentos dos casos de uso entre si e com os atores do sistema.

Os diagramas de caso de uso são utilizados para capturar o modo com o qual os usuários querem utilizar o sistema. O diagrama de caso de uso descreve a funcionalidade e os usuários (atores) do sistema. Eles formam uma definição de fora para dentro dos requisitos para o sistema de computador, para que possam ser entendidos por usuários e desenvolvedores.

Um caso de uso individual pode também ser visto como um procedimento por meio do qual um ator externo pode utilizar o sistema. Tomados em conjunto, os casos de uso definem a funcionalidade plena do sistema partindo-se de uma perspectiva de fora para dentro, e podem ser utilizados como base para o desenvolvimento de testes do sistema. Os dois conceitos principais em um diagrama de caso de uso são:

Ator: Representa os usuários do sistema, podendo ser um ser humano ou não, como outro sistema, por exemplo.

Caso de Uso: Representa serviços ou a funcionalidade provida pelo sistema aos usuários.

Existem ainda os fluxos de caso de uso, que descrevem os detalhes de cada caso de uso.

Um caso de uso é representado por uma figura oval. O nome do caso de uso pode aparecer dentro ou abaixo da figura oval. O caso de uso é anexado a, no mínimo, um ator, representado por um *stick man*, por intermédio de uma linha denominada relacionamento. Ela mostra como um ator ou vários atores externos a um sistema estão implicados no caso de uso. Um exemplo é mostrado na Figura 1:

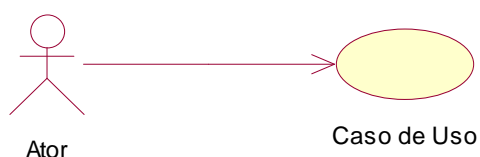


Figura1 - Exemplo de Caso de Uso.
Fonte: UML Guia do Usuário. (JACOBSON et al, 2000).

Um caso de uso pode estar relacionado com outro caso de uso, quer por um relacionamento de inclusão, ou por um de exclusão. Um relacionamento de inclusão (incluir) indica que o procedimento do caso de uso utilizado é parte do procedimento do caso de uso que está sendo empregado. Se o procedimento do caso de uso for um curso alternativo ou parcialmente alternativo para um curso definido em um outro caso de uso, então um relacionamento 'estender' será utilizado desde o caso de uso que estende, então, substitui uma parte ou todo o procedimento no caso de uso 'estendido' sob as condições especificadas no caso de uso 'que estende'.

2.3.2 Diagrama de Classe

Segundo Pádua Filho (2000, p.99), as classes representam conceitos do mundo da aplicação que sejam relevantes para a descrição mais precisa dos

requisitos. Os diagramas de classe mostram os relacionamentos entre estas, e as especificações das classes descrevem os respectivos detalhes.

Os diagramas de classe são utilizados para definir o modelo de estrutura estática do sistema. O modelo de estrutura estática identifica os objetos, classes e relacionamentos entre eles, em outras palavras, ele descreve como o sistema é estruturado e não como ele se comporta. Descreve os que são os elementos e seus relacionamentos estáticos com outros elementos. Diagramas de classe são, portanto, utilizados com mais frequência do que diagramas de objeto para mostrar essa perspectiva.

Classes definem os tipos de objetos que existem dentro do sistema. Classes podem ter atributos que são geralmente membros de dados primitivos de objetos e operações definidoras de métodos que podem ser aplicados sobre os objetos. A visibilidade de atributos e operações de outros objetos pode ser definida, igualmente a suas assinaturas, incluindo tipos, valores padrão, parâmetros, tipos de parâmetros e retorno.

Uma classe representa entidades com características e comportamentos comuns (ou seja, atributos, serviços, regras e relacionamentos). Um objeto é ilustrado como um retângulo tendo o nome sublinhado. O nome integral de um objeto é o nome do objeto e o nome da classe separados por dois pontos. Em alguns casos, o nome do objeto ou o nome da classe não estão presentes. Em todos os casos, o nome do objeto está sublinhado para distingui-lo de uma classe. A Figura 2 mostra a notação de objetos na UML.

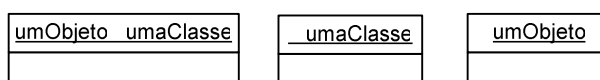


Figura 2 - Notação UML para Objetos.

Fonte: UML Guia do Usuário. (JACOBSON et al, 2000).

Reconhece-se na UML que uma classe pode aparecer em múltiplos diagramas. A Figura 3 apresenta dois modos pelos quais as classes são representadas nos diagramas UML. O primeiro exemplo revela como uma classe é representada quando os atributos e as operações a ela associados são suprimidos. No segundo exemplo, o nome da classe aparece na divisão superior, os atributos na divisão intermediária, e as operações na divisão inferior.

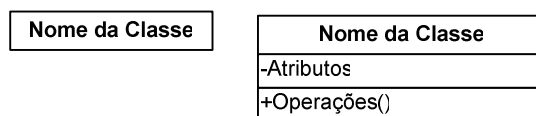


Figura 3 - Representações da UML para classes.

Fonte: UML Guia do Usuário. (JACOBSON et al, 2000).

Os atributos são especificados de acordo com a seguinte notação:

Visibilidade nome: tipo = valor padrão, onde:

Visibilidade: (+) é pública; (#) é protegida; (-) é privada;

Nome: é uma cadeia de caracteres pela qual o atributo é identificado;

Tipo: é o tipo de atributo;

Valor padrão: é um valor designado ao atributo a menos que de outra forma seja especificado.

As operações são especificadas de acordo com a seguinte notação:

Visibilidade nome (lista de parâmetros): tipo de retorno [propriedade-*string*], onde:

Visibilidade: (+) é pública; (#) é protegida; (-) é privada;

Nome: é uma cadeia de caracteres pela qual o atributo é identificado;

Lista de parâmetros: contém parâmetros separados por vírgulas, os quais são dados por:

Direção nome: tipo = valor padrão, onde:

Direção: indica se o parâmetro é para *input* (in), *output* (out) ou para ambos (inout);

Nome: indica o nome do parâmetro;

Tipo: é o tipo do parâmetro;

Valor padrão: identifica o valor padrão do parâmetro.

Tipo de retorno: é a lista de tipos de retorno separados por ponto e vírgula;

Propriedade-string: indica valores da propriedade aplicáveis à operação.

Segundo Lee & Tapfenhart (2001), os relacionamentos de objetos entre classes evidenciam que tipos de vínculos existem entre objetos, e definem restrições sobre esses vínculos incluindo a quantidade relativa de instâncias vinculadas por uma associação. Diagramas de classe também mostram pacotes que agrupam

classes, dependências entre classes e dependências entre os pacotes que as contêm. Relacionamentos de generalização/especialização que correlacionam classes em relacionamento do tipo supertipo/subtipo também podem ser incluídos.

Relacionamentos podem existir entre instâncias de classes ou entre classes. Associações e agregações são relacionamentos que conectam duas ou mais outras instâncias de classes (associação e agregação). Generalização e especialização são relacionamentos entre duas classes.

A Figura 4 mostra a representação básica de uma associação UML. A linha que interliga as classes é rotulada com o nome da associação, as multiplicidades das duas classes (número de objetos da determinada classe) participando da associação e os papéis que as instâncias de cada classe assumem dentro da associação. Todos estes rótulos são opcionais.

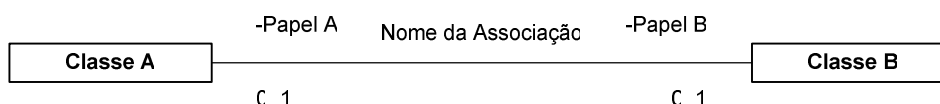


Figura 4 - Associações na UML.

Fonte: UML Guia do Usuário. (JACOBSON et al, 2000).

A multiplicidade associada a uma extremidade de uma associação pode ser dada em qualquer uma das formas mostradas no Quadro 1.

Quadro 1 - Multiplicidades

Chave	Interpretação
*	Qualquer número de objetos (inclusive nenhum)
1	Exatamente um objeto.
N	Exatamente n objetos (n é um número inteiro).
0..1	Zero ou um (indica que a associação é opcional).
n..m	Varia de n como mínimo até m como máximo.
2,4	Combinações distintas (como em duas ou quatro portas).

Fonte: UML Guia do Usuário. (JACOBSON et al, 2000).

A hipótese padrão da associação é que ela pode ser navegada por um objeto a partir de qualquer uma das extremidades dela. Para indicar a passagem

unidirecional (Figura 5), a linha que conecta as duas classes é substituída por uma seta apontando para a direção da navegação (passagem)

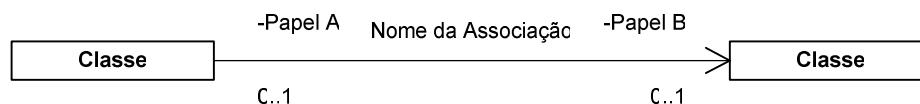


Figura 5 - Passagem unidirecional por uma associação é indicada por uma seta.
Fonte: UML Guia do Usuário. (JACOBSON et al, 2000).

A classe de associação (Figura 6) é conectada ao vínculo entre as duas classes participantes da associação por uma linha tracejada. A classe de associação pode ter atributos e operações associadas por ela.

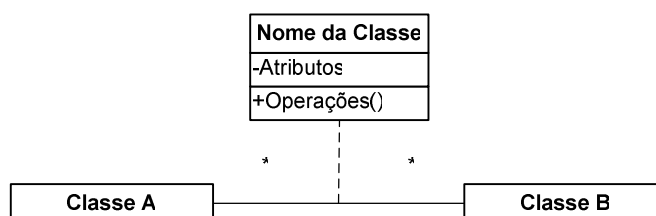


Figura 6 - Classe de Associação.
Fonte: UML Guia do Usuário. (JACOBSON et al, 2000).

A Figura 7 mostra como agregação e composição são documentadas. A navegação unidirecional pela agregação ou composição é ilustrada utilizando-se uma seta da mesma maneira que em associações, assim como a multiplicidade.

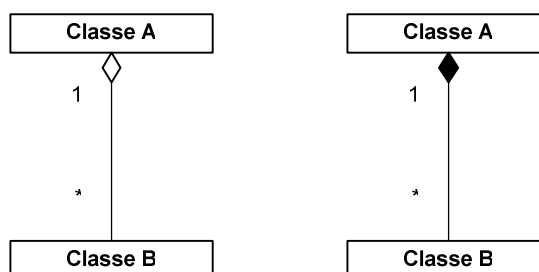


Figura 7 - Agregação e Composição.
Fonte: UML Guia do Usuário. (JACOBSON et al, 2000).

A Figura 8 mostra a generalização e a especialização. A generalização é indicada como a classe com a ponta do triângulo apontado para ela. A subclasse 1

ilustra o caso de generalização simples enquanto a subclasse 2 ilustra o caso de generalização múltipla. Generalização pode ter um discriminador que distingue entre diferentes tipos de generalizações.

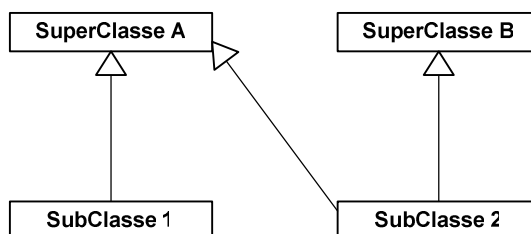


Figura 8 - Generalização e especialização.

Fonte: UML Guia do Usuário. (JACOBSON et al, 2000).

A Figura 9 mostra como dependências entre classes são documentadas. Dependências são tipicamente criadas dinamicamente. Exemplo de dependências são: *friend*, *derived*, *import*, etc.

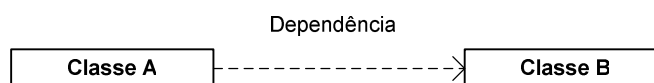


Figura 9 - Dependência entre classes.

Fonte: UML Guia do Usuário. (JACOBSON et al, 2000).

2.3.3 Diagramas de Seqüência

Segundo Pádua Filho (2000, p.397), os diagramas de seqüência enfatizam o ordenamento temporal das ações, ou seja, devem exprimir o desenrolar temporal de seqüência de ações.

O diagrama de seqüência captura a interação entre objetos. Essas interações são modeladas como intercâmbios de mensagens. Esses intercâmbios resultarão em algum comportamento desejado. Ele pode mostrar as interações que ocorrem: todas através da fronteira do sistema; e entre objetos dentro dele para satisfazer os requisitos definidos para um ou mais caminhos por meio de um caso de uso.

Um diagrama de seqüência é um diagrama que mostra objetos reais e interações entre objetos no sentido horizontal, e seqüência no sentido vertical. As linhas pontilhadas verticais representam a linha de vida do objeto e as setas

horizontais representam as interações ou mensagens entre objetos. Caixas alongadas estreitas nas linhas de vida de um objeto representam a sua “ativação” quando interações são seqüenciais e representam chamadas de operações. A operação permanece ativa até que todas as operações seqüenciais que ela invoca sejam finalizadas e devolvidas; permitindo dessa maneira retornar controle ao objeto solicitador.

A Figura 10 mostra um exemplo de diagrama de seqüência simples.

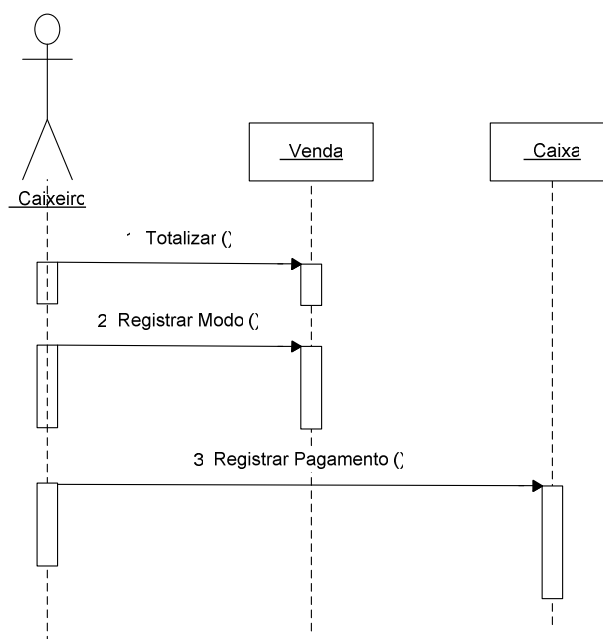


Figura 10 - Exemplo de Diagrama de Seqüência.
Fonte: UML Guia do Usuário. (JACOBSON et al, 2000).

2.3.4 Diagrama de Estado

Segundo Martin & Odell (1995, p.345), um diagrama de estado descreve como a funcionalidade de um objeto depende de seu estado e como este estado é alterado em face do resultado dos eventos que ele recebe. Ele é utilizado para mostrar a dependência de operações sob a ordem de suas invocações (ou seja, mediante o estado do objeto quando uma solicitação de uma operação chegar na fronteira de um objeto).

Os conceitos e trajetórias deste diagrama são:

Estado: representa uma abstração dos valores dos atributos de um objeto;

Evento: representa uma condição que pode ser detectada pelo objeto. Eventos podem causar uma transição para outro estado e/ou provocar o disparo de uma ou mais ações;

Transição: representa uma resposta de um objeto a um evento por ele recebido. A resposta produz uma mudança no objeto, que pode constituir uma mudança no estado.

Ação: representa um conjunto de operações que é feito dentro de um estado ou em uma transição.

O próximo capítulo aborda a especificação de requisitos de software, detalhando um conjunto de requisitos estabelecidos em entrevistas e questionários e que devem ser satisfeitos pela solução implementável do problema abordado.

3 ESPECIFICAÇÃO DE REQUISITOS DE SOFTWARE

A fase de especificação de requisitos foi realizada a partir da coleta de informações através de entrevistas e questionários com gerentes e funcionários de 1diversos setores da Empresa X (vide anexo A).

Foi constatado nestas entrevistas que grande parte dos setores da empresa necessitam de um sistema informatizado que dê apoio aos seus processos internos. Entretanto o sistema utilizado na empresa atualmente não oferece suporte a essas atividades e, conseqüentemente, são realizadas manualmente ou com utilização de ferramentas que tendem a falhas e erros, gerando prejuízos.

A proposta de desenvolvimento de um novo sistema para atender às necessidades de controle de processos dos diversos setores da cooperativa foi bem recebida pelo corpo gerencial da empresa, contudo, foi destacado pelo gerente geral desta, a prioridade no desenvolvimento de alguns módulos. Durante a realização das entrevistas, o gerente geral ressaltou os módulos prioritários: *“O controle da venda, pagamentos e inadimplência dos convênios de planos de saúde, além dos seguros de várias modalidades oferecidas como vida, não-nominado, veículos, equipamento e residencial, além do controle do comissionamento gerado pela venda de cada plano. Operações de crédito complexas, que atualmente são controlados por meio de planilhas eletrônicas”*.

Diante do exposto, este trabalho abordará os módulos considerados prioritários no desenvolvimento do sistema como um todo, módulos estes que serão descritos a seguir.

O sistema de informação será denominado **SASC**: Sistema de Apoio Empresa X.

3.1 MISSÃO DO PRODUTO

O SASC visa oferecer apoio informatizado aos processos de controle realizados na Empresa X.

3.2 QUADRO DE FUNÇÕES / MÓDULOS DO SISTEMA

Para facilitar a interação entre usuário e sistema, O SASC foi dividido em diversos módulos. Cada módulo representa determinado setor ou área da empresa e possui as atividades relacionadas ao mesmo. O Quadro 2, a seguir, apresenta os módulos do SASC, suas funções, e os benefícios que a implantação do novo sistema buscará atingir.

Quadro 2 – Módulos e Funções do SASC

Número de Ordem	Nome da Função	Módulo	Necessidades	Benefícios
1	Controle dos seguros.	Seguros	Cadastramento de seguros, renovações, geração de comissões, emissão de relatórios.	Informações centralizadas e confiáveis, diminuição de erros, auxílio à tomada de decisão.
2	Controle de planos de saúde.	Planos de Saúde	Cadastramento de planos de saúde, dependentes, geração de comissões, automatização de pagamentos e recebimento, emissão de relatórios.	Informações centralizadas e confiáveis, diminuição de erros, auxílio à tomada de decisão. Automatização do processo de recebimento e pagamento, diminuição de prejuízos. Maior agilidade no processo de fechamento de faturas.
3	Controle de Comissões.	Comissões	Controle de comissões geradas pela venda de produtos, baixa de comissões pagas, cadastramento de angariadores, emissão de relatórios.	Informações centralizadas e confiáveis, diminuição de erros e prejuízo, auxílio à tomada de decisão. Maior rapidez para processo de pagamento de comissões.

4	Controle de Tarifas.	Tarifas	Controle de tarifas bancárias em atraso. Automatização do processo de armazenamento e baixa (recebimento) de tarifas, emissão de relatórios.	Informações geradas automaticamente, centralizadas e confiáveis, diminuição de erros e prejuízo, auxílio à tomada de decisão.
5	Controle de Associados	Associados	Cadastramento de associados, C/Cs, atualização de saldo de C/Cs, emissão de relatórios.	Informações centralizadas e confiáveis, visualização de produtos adquiridos, ferramenta de apoio à venda de novos produtos.
6	Controle de Usuários e Informações do Sistema	Administração do Sistema	Cadastramento de usuários e permissões de acesso para cada módulo do sistema. Cadastramento de novas agências e controle das informações do sistema.	Controle de acesso ao sistema, prevenção de informações errôneas.

3.3 REQUISITOS DE QUALIDADE

O SASC deverá atender aos seguintes requisitos de qualidade:

- A utilização será feita através de interface gráfica amigável;
- Possibilidade de interação e utilização de dados gerados por sistema já em uso pela empresa;
- Possibilidade de acréscimo de novos módulos e novas funcionalidades de acordo com a demanda da empresa;
- Deverá ser possível a expansão do sistema para todas as agências da Empresa X;

- Controle de acesso ao sistema e identificação dos autores de operações realizadas.

3.4 LIMITES

O SASC não substituirá os sistemas já existentes e em uso pela cooperativa, porém, poderá oferecer funcionalidades já existentes em outro sistema, ou utilizar dados gerados por este de modo a tornar mais eficientes os processos realizados pela empresa.

Embora o SASC possua mecanismos de importação de dados provenientes de outros sistemas, é necessária a constante atualização de sua base de dados.

3.5 BENEFÍCIOS

O Quadro 3 mostra o levantamento dos benefícios que se espera obter com o SASC.

Quadro 3 – Benefícios do SASC

Nº. de Ordem	Benefício	Valor para o Cliente
1	Automatizar processos da empresa que são realizados manualmente ou com ferramentas limitadas.	Essencial
2	Eliminar falhas, agilizar os processos, evitar re-trabalho e consequentemente redução de custos e prejuízos.	Essencial
3	Melhor utilização dos recursos tecnológicos da empresa, utilizando uma ferramenta específica para solução de cada problema.	Essencial
4	Garantir consistência de dados gerados.	Essencial
5	Promover controle de acesso às informações de cada setor da empresa.	Essencial
6	Servir como ferramenta de suporte à tomada de decisões.	Desejável

7	Integrar e disponibilizar as informações geradas para todas as agências da empresa economizando recursos.	Desejável
---	---	-----------

3.6 ATORES

O Quadro 4 mostra o diagrama de descrição dos atores do SASC:

Quadro 4 – Descrição dos Atores

Nº. de Ordem	Ator(es)	Definição
1	Administrador Sistema	Funcionário responsável pelo cadastramento de novos usuários e definição das suas permissões, realização de backup e suporte aos demais usuários do sistema. Cadastramento de novas agências e angariadores.
2	Funcionário Seguros	Funcionário do setor de seguros responsável pela consistência do cadastro de seguros.
3	Supervisor Seguros	Supervisor da unidade de seguros responsável pelo gerenciamento dos seguros com vencimento próximo. Realiza a renovação e/ou exclusão de seguros vencidos.
4	Funcionário Plano de Saúde	Funcionário do setor de planos de saúde responsável pela consistência do cadastro de planos de saúde (cadastramentos de beneficiários, dependentes e geração de comissões).
5	Supervisor Plano de Saúde	Supervisor da unidade de planos de saúde responsável pela conciliação e fechamento da fatura mensal dos planos de saúde, atualização da base de dados do SASC (saldos de C/Cs) e processamento de pagamentos.
6	Supervisor de produtos	Supervisor da unidade de produtos responsável pelo gerenciamento das comissões geradas na venda dos planos de saúde e seguros. Realiza o pagamento das comissões aos seus devidos angariadores.
7	Funcionário Tarifas	Funcionário da unidade de crédito responsável pela atualização da base de dados do SASC (importação de tarifas do dia e saldo de C/Cs). Gerenciamento das tarifas vencidas e processamento de pagamentos.
8	Funcionário Cadastro	Funcionário da unidade de cadastro responsável pelo cadastramento de novos associados, cadastramento de C/Cs, atualização de cadastros e contas-correntes.

O Quadro 5 mostra o perfil dos atores do SASC:

Quadro 5 – Características dos Atores

Nº. de Ordem	Ator(es)	Permissão de Acesso	Frequência de Uso	Nível Educacional	Proficiência Aplicação	Proficiência em Informática
1	Administrador Sistema	Gestão de Usuários, Acesso a todos os módulos.	Eventual	2º Grau	Completa	Avançado
2	Funcionário Seguros	Consulta todos os módulos. Controle do cadastro de Seguros.	Diário	2º Grau	Operacional	Aplicação
3	Supervisor Seguros	Consulta todos os módulos. Gestão dos Seguros.	Diário	2º Grau	Completa	Aplicação
4	Funcionário Plano de Saúde	Consulta todos os módulos. Controle do cadastro de Planos de Saúde.	Diário	2º Grau	Operacional	Aplicação
5	Supervisor Plano de Saúde	Consulta todos os módulos. Processar Pagamento de Planos.	Diário	2º Grau	Completa	Aplicação
6	Supervisor de produtos	Consulta todos os módulos. Processar Pagamento de Comissões.	Diário	2º Grau	Completa	Aplicação
7	Funcionário Tarifas	Consulta todos os módulos. Processar Pagamento de Tarifas.	Diário	2º Grau	Operacional	Aplicação
8	Funcionário Cadastro	Consulta todos os módulos. Controle do cadastro de associados.	Diário	2º Grau	Operacional	Aplicação

3.7 DIAGRAMA DE CONTEXTO

A Figura 11 mostra o diagrama de contexto da aplicação, segundo Pádua Filho (2000), o diagrama de contexto deve mostrar as interfaces do produto com seu ambiente de aplicação, inclusive os diversos tipos de usuários e outros sistemas do cliente os quais o produto deve interagir.

O diagrama de contexto do SASC exibe as interações entre os módulos existentes no sistema (conforme mostrado no Quadro 2) com os seus atores principais (conforme mostrado no Quadro 4).

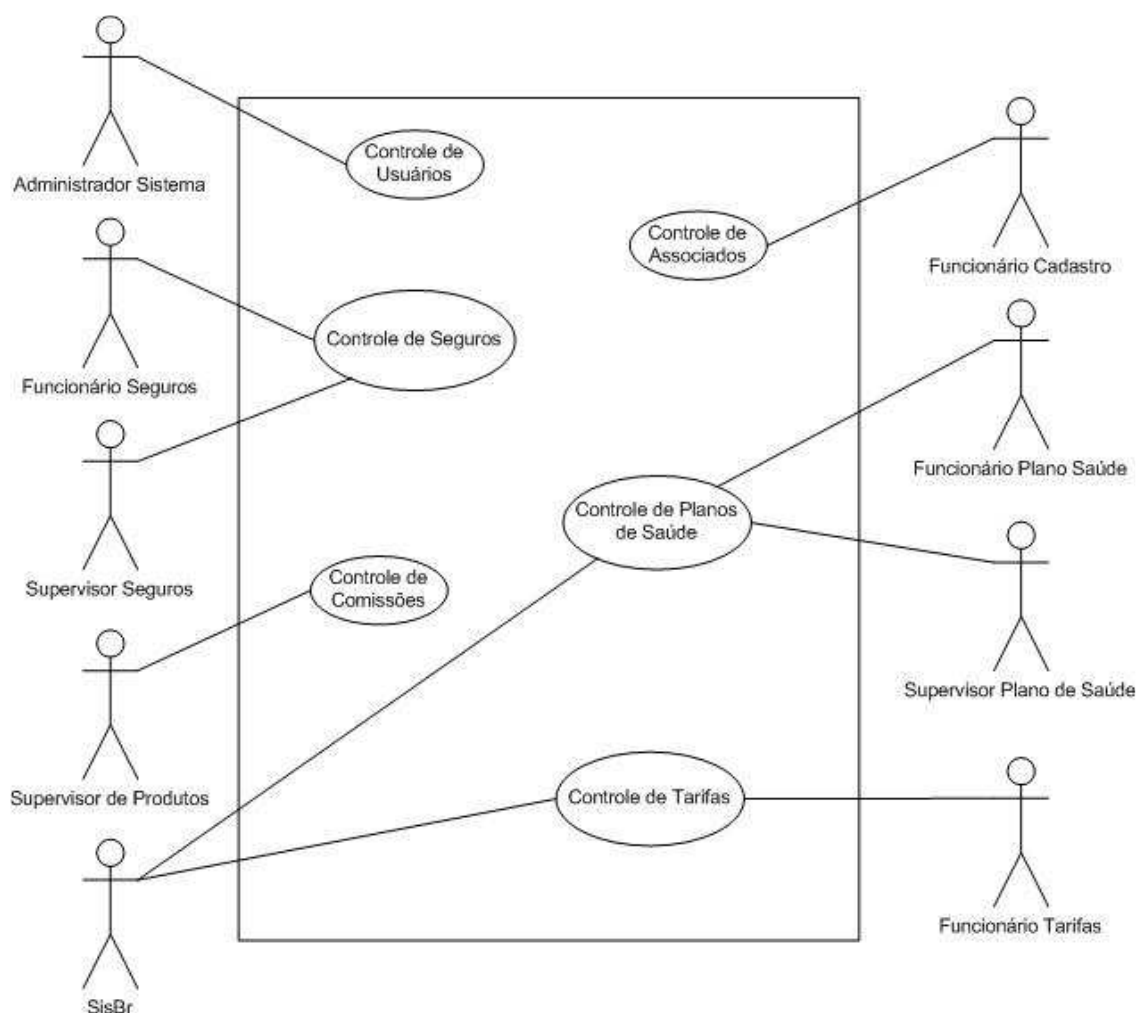


Figura 11 – Diagrama de Contexto

O próximo capítulo mostra a análise de sistema de software baseada na tecnologia Orientada a Objetos e expressa na notação UML.

4 ANÁLISE DO SISTEMA

4.1 DIAGRAMAS DE CASO DE USO

São apresentados a seguir, os diagramas de casos de uso e os principais fluxos de caso de uso de cada módulo do SASC.

4.1.1 Módulo Administração do Sistema

A Figura 12 mostra o diagrama de caso de uso do módulo “Administração do Sistema”.

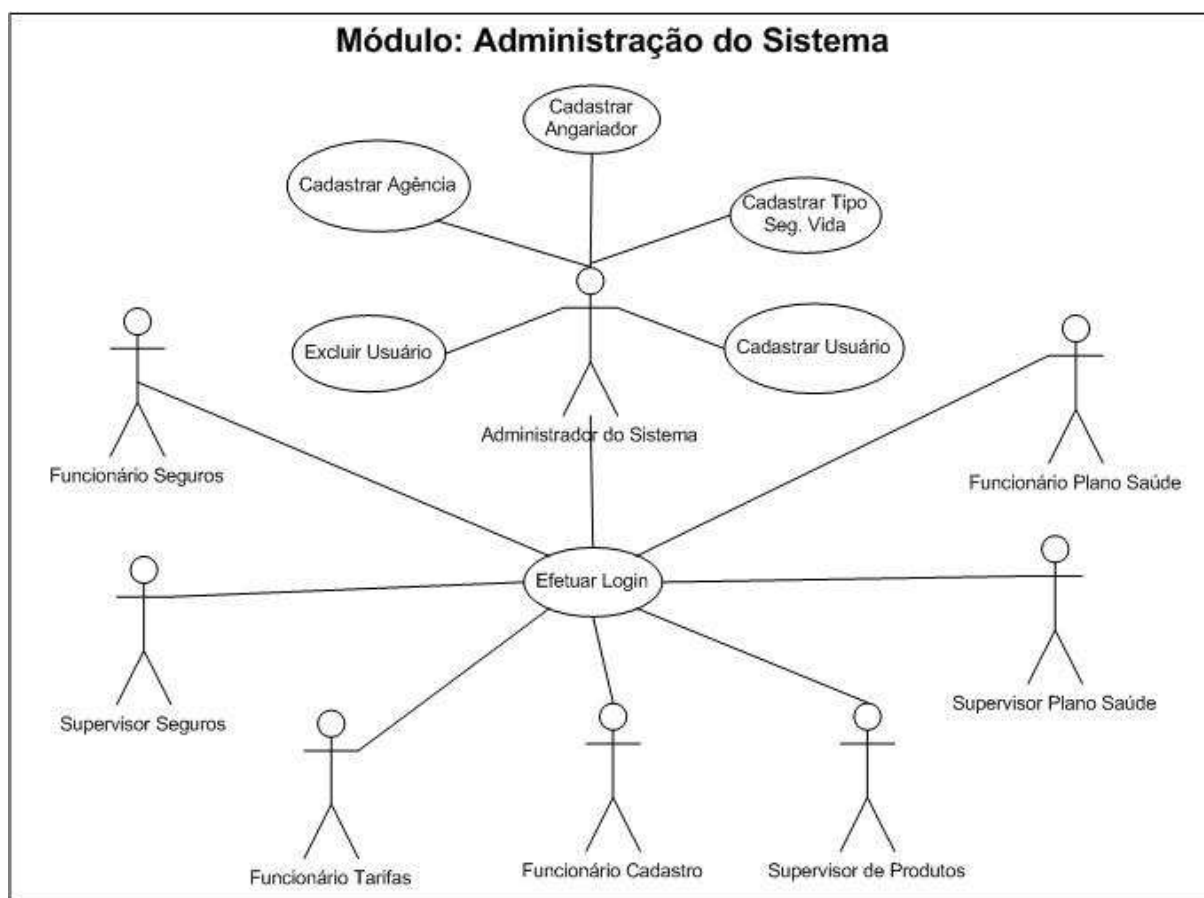


Figura 12 - Caso de Uso do Módulo Administração do Sistema

Quadro 6 – Descrição do Caso de Uso “Efetuar Login”

Caso de Uso: Efetuar Login
Pré-requisitos <ol style="list-style-type: none"> 1. Funcionário ter sido cadastrado no sistema, possuir <i>login</i> e senha.
Fluxo Principal <ol style="list-style-type: none"> 1. Funcionário informa login e senha para autenticação. 2. Sistema é aberto. 3. Restrições de acesso são aplicadas.
Tratamento de exceções: <ol style="list-style-type: none"> 1a. Login e senha não são autenticados: <ol style="list-style-type: none"> 1a.1. Sistema é fechado.

Quadro 7 – Descrição do Caso de Uso “Cadastrar Agência”.

Caso de Uso: Cadastrar Agência
Pré-requisitos <ol style="list-style-type: none"> 1. Sistema ter autenticado <i>login</i> e senha do Administrador do Sistema.
Fluxo Principal <ol style="list-style-type: none"> 1. Administrador informa nome da nova agência. 2. Sistema informa operação concluída com sucesso.

Quadro 8 – Descrição do Caso de Uso “Cadastrar Angariador”.

Caso de Uso: Cadastrar Angariador
Pré-requisitos <ol style="list-style-type: none"> 1. Sistema ter autenticado <i>login</i> e senha do Administrador do Sistema.
Fluxo Principal <ol style="list-style-type: none"> 1. Administrador informa nome do novo angariador. 2. Sistema informa operação concluída com sucesso.

Quadro 9 – Descrição do Caso de Uso “Cadastrar Usuário”.

Caso de Uso: Cadastrar Usuário
Pré-requisitos <ol style="list-style-type: none"> 1. Sistema ter autenticado <i>login</i> e senha do Administrador do Sistema.

Fluxo Principal

1. Administrador informa login do novo usuário.
2. Administrador solicita ao usuário o cadastramento de senha.
3. Usuário informa uma senha de seu conhecimento.
4. Administrador registra restrições de acesso do usuário.
5. Sistema informa operação concluída com sucesso.

Quadro 10 – Descrição do Caso de Uso “Excluir Usuário”.

Caso de Uso: Excluir Usuário**Pré-requisitos**

1. Sistema ter autenticado *login* e senha do Administrador do Sistema.

Fluxo Principal

1. Administrador pesquisa login do usuário a ser excluído.
2. Sistema pede confirmação de exclusão.
3. Administrador confirma exclusão.
4. Sistema informa operação concluída com sucesso.

4.1.2 Módulo Planos de Saúde

A Figura 13 mostra o diagrama de caso de uso do módulo “Plano de Saúde”.

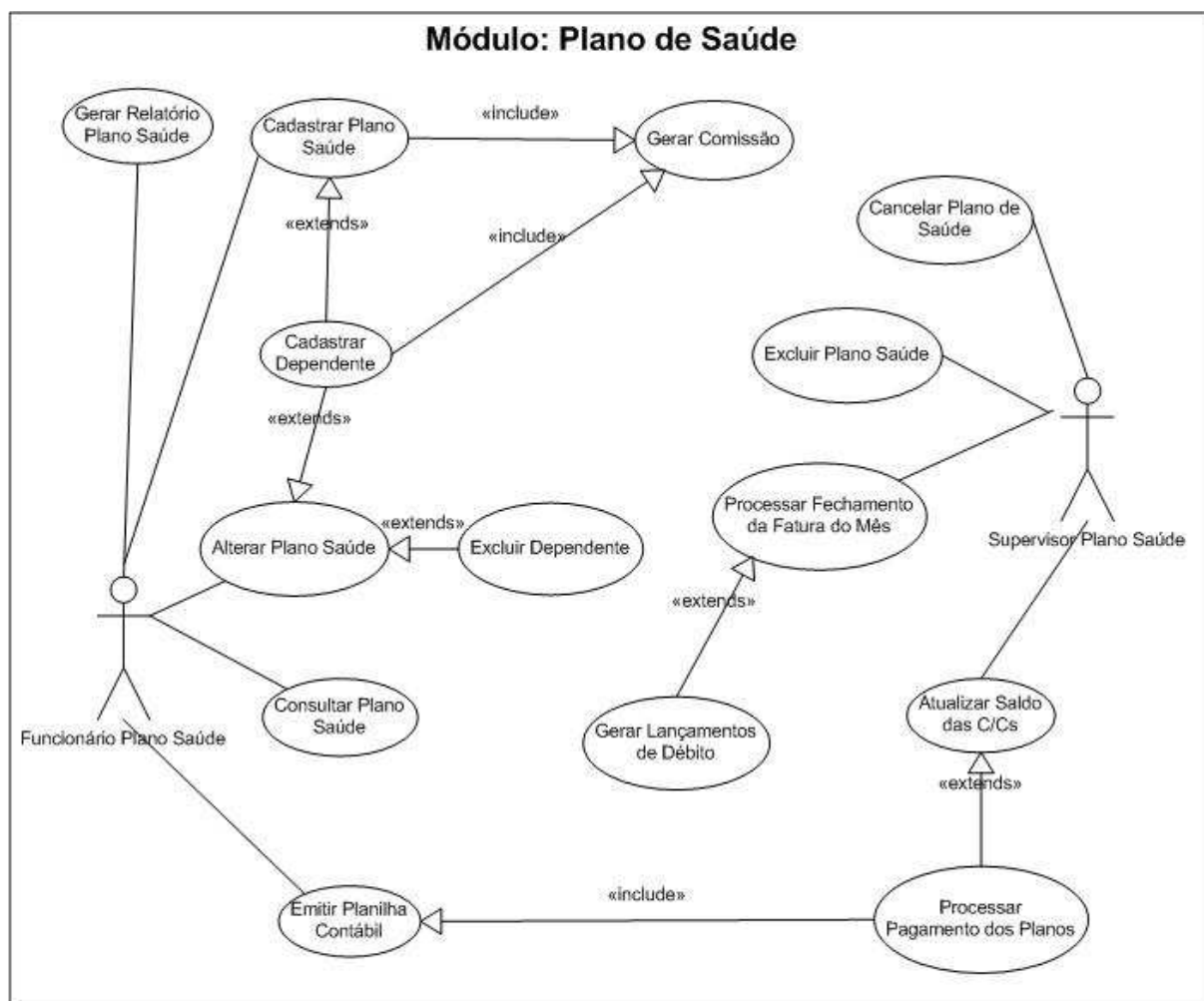


Figura 13 - Caso de uso Plano de Saúde

Quadro 11 – Descrição do Caso de Uso “Cadastrar Plano Saúde”.

Caso de Uso: Cadastrar Plano Saúde
<p>Pré-requisitos</p> <ol style="list-style-type: none"> 1. Sistema ter autenticado <i>login</i> e senha do funcionário. 2. Funcionário ter acesso ao módulo Plano de Saúde. 3. Associado ter sido cadastrado no sistema.

<p>Fluxo Principal</p> <ol style="list-style-type: none"> 1. Associado procura funcionário para aquisição de plano de saúde. 2. Funcionário pesquisa o nome do associado beneficiário do plano de saúde. 3. Funcionário informa dados do plano de saúde. 4. Funcionário efetua cadastramento do novo plano de saúde. 5. Sistema informa operação concluída com sucesso. 6. Sistema solicita ao funcionário confirmação de geração da comissão: <ul style="list-style-type: none"> - Caso geração da comissão confirmada: Ver variante 7.1. 7. Funcionário pergunta ao associado se existem dependentes: <ul style="list-style-type: none"> - Caso existam: ver variante 8.1.
<p>Variantes</p> <ol style="list-style-type: none"> 7.1.1. Sistema gera comissão para funcionário/angariador do plano. 7.1.2. Sistema informa operação concluída com sucesso. 8.1.1. Funcionário informa dados do dependente. 8.1.2. Funcionário efetua cadastramento do dependente. 8.1.3. Sistema informa operação concluída com sucesso.

Quadro 12 – Descrição do Caso de Uso “Alterar Plano Saúde”.

Caso de Uso: Alterar Plano Saúde
<p>Pré-requisitos</p> <ol style="list-style-type: none"> 1. Sistema ter autenticado <i>login</i> e senha do funcionário. 2. Funcionário ter acesso ao módulo Plano de Saúde.
<p>Fluxo Principal</p> <ol style="list-style-type: none"> 1. Funcionário pesquisa o nome do associado beneficiário do plano de saúde. 2. Funcionário consulta dados do plano de saúde. 3. Funcionário Altera dados do plano de saúde. 4. Funcionário salva alterações do plano de saúde. 5. Sistema informa operação concluída com sucesso.

Quadro 13 – Descrição do Caso de Uso “Cancelar Plano Saúde”.

Caso de Uso: Cancelar Plano Saúde
<p>Pré-requisitos</p> <ol style="list-style-type: none"> 1. Sistema ter autenticado <i>login</i> e senha do supervisor. 2. Associado ter plano de saúde cadastrado no sistema.
<p>Fluxo Principal</p> <ol style="list-style-type: none"> 1. Associado solicita ao supervisor cancelamento de plano de saúde. 2. Supervisor pesquisa o nome do associado beneficiário do plano de saúde. 3. Supervisor cancela plano de saúde. 4. Sistema informa operação concluída com sucesso.

Quadro 14 – Descrição do Caso de Uso “Processar Fechamento Fatura do Mês”.

Caso de Uso: Processar Fechamento da Fatura do Mês
<p>Pré-requisitos</p> <ol style="list-style-type: none"> 1. Sistema ter autenticado <i>login</i> e senha do supervisor.
<p>Fluxo Principal</p> <ol style="list-style-type: none"> 1. Convênio envia para supervisor fatura mensal dos planos de saúde. 2. Supervisor processa fechamento da fatura mensal no sistema. 3. Supervisor concilia fatura gerada pelo sistema com fatura emitida pelo convênio: <ul style="list-style-type: none"> - Caso concilie: Supervisor gera lançamentos de débitos. - Caso não concilie: Supervisor encontra diferenças entre faturas. 4. Sistema informa operação concluída com sucesso.

Quadro 15 – Descrição do Caso de Uso “Atualizar Saldo das C/Cs”.

Caso de Uso: Atualizar Saldo das C/Cs
<p>Pré-requisitos</p> <ol style="list-style-type: none"> 1. Sistema ter autenticado <i>login</i> e senha do supervisor. 2. Supervisor ter exportado banco de dados de C/Cs do SiS-Br.

Fluxo Principal

1. Supervisor importa saldo das C/Cs para o sistema.
2. Supervisor processa pagamento dos planos cujo saldo em C/C for suficiente.
3. Sistema emite planilha contábil para realização de débito das C/Cs.
4. Supervisor assina planilha contábil
5. Supervisor envia planilha contábil para setor de contabilidade.

4.1.3 Módulo Seguros

A Figura 14 mostra o diagrama de caso de uso do módulo “Seguros”.

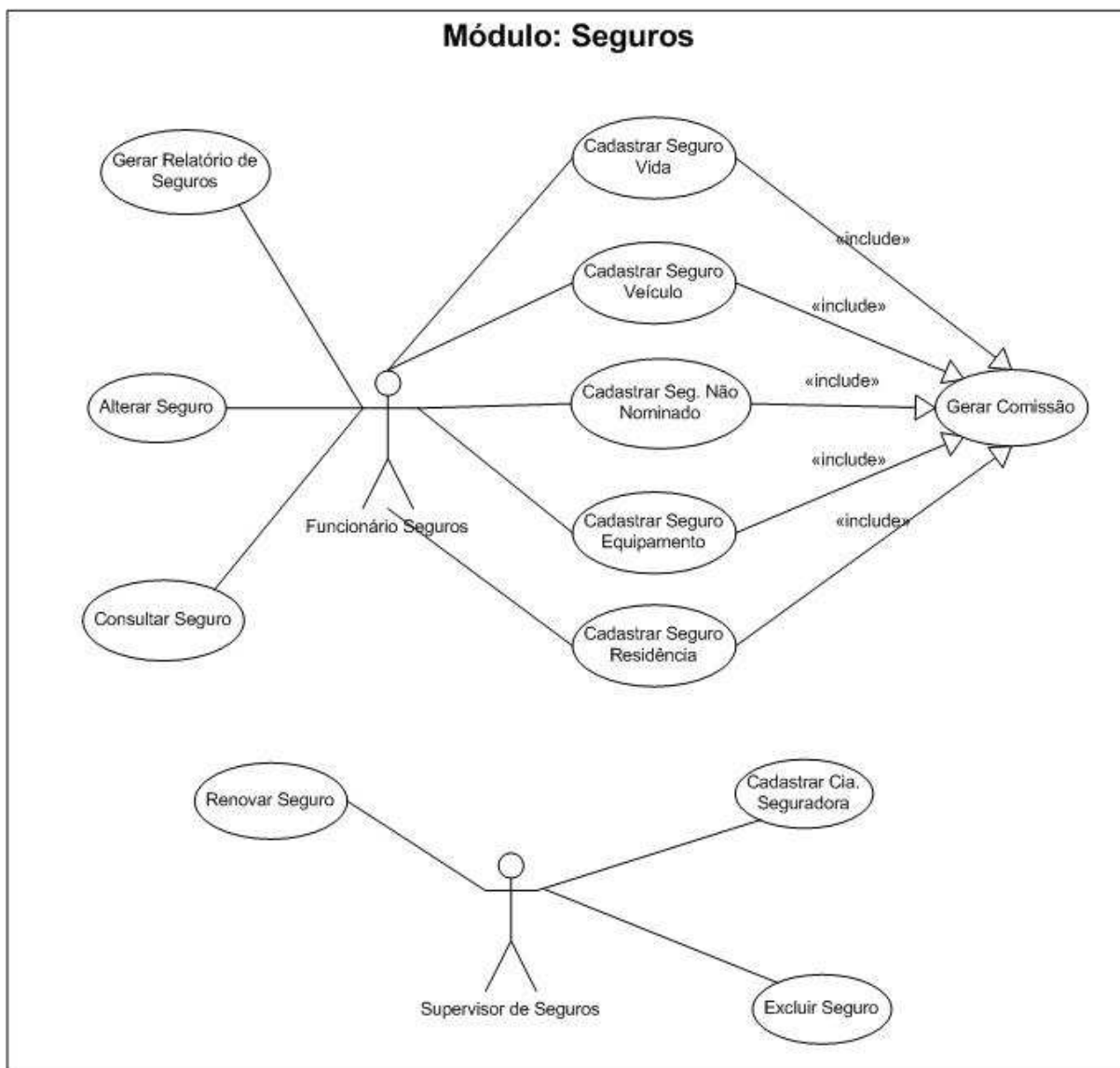


Fig. 14 – Caso de Uso do Módulo Seguros

Quadro 16 – Descrição do Caso de Uso “Cadastrar Seguro Vida”.

Caso de Uso: Cadastrar Seguro Vida
<p>Pré-requisitos</p> <ol style="list-style-type: none"> 1. Sistema ter autenticado <i>login</i> e senha do funcionário. 2. Funcionário ter acesso ao módulo de Seguros. 3. Associado ter sido cadastrado no sistema.
<p>Fluxo Principal</p> <ol style="list-style-type: none"> 1. Associado procura funcionário para aquisição de seguro. 2. Funcionário pesquisa o nome do associado beneficiário do seguro. 3. Funcionário informa dados do seguro. 4. Funcionário efetua cadastramento do novo seguro. 5. Sistema informa operação concluída com sucesso. 6. Sistema gera comissão para funcionário/angariador do plano.

Quadro 17 – Descrição do Caso de Uso Renovar Seguro.

Caso de Uso: Renovar Seguro
<p>Pré-requisitos</p> <ol style="list-style-type: none"> 1. Sistema ter autenticado <i>login</i> e senha do Supervisor.
<p>Fluxo Principal</p> <ol style="list-style-type: none"> 1. Supervisor contata associado para autorização de renovação de seguro. 2. Associado autoriza renovação do seguro. 3. Supervisor localiza seguro no sistema. 4. Supervisor efetua renovação do seguro. 5. Sistema informa operação concluída com sucesso. 6. Sistema gera comissão para funcionário/angariador do plano.

4.1.4 Módulo Associados

A Figura 15 mostra o diagrama de caso de uso do módulo “Associados”.

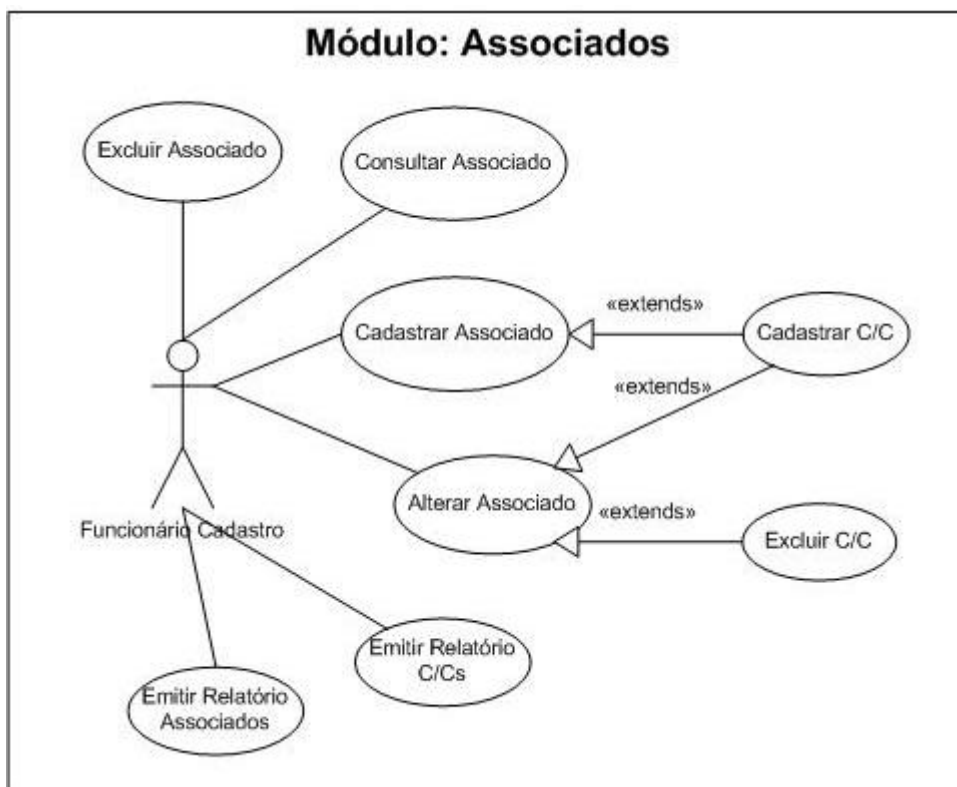


Fig. 15 – Casos de uso do Módulo Associados

Quadro 18 – Descrição do Caso de Uso Cadastrar Associado.

Caso de Uso: Cadastrar Associado
<p>Pré-requisitos</p> <ol style="list-style-type: none"> 1. Sistema ter autenticado <i>login</i> e senha do funcionário. 2. Funcionário ter acesso ao módulo de Associados.
<p>Fluxo Principal</p> <ol style="list-style-type: none"> 1. Cliente procura funcionário solicitando processo de associação na cooperativa de crédito. 2. Funcionário solicita documentação necessária à associação. 3. Cliente entrega documentação necessária. 4. Funcionário cadastra os dados do novo associado com base na documentação entregue. 5. Sistema informa operação concluída com sucesso. 6. Sistema solicita confirmação para cadastramento de C/C para o associado: <ul style="list-style-type: none"> - Caso funcionário confirme: Ver variante 6.1.

Variantes

Funcionário informa número da conta-corrente do associado.

Sistema informa operação concluída com sucesso.

4.1.5 Módulo Tarifas

A Figura 16 mostra o diagrama de caso de uso do módulo “Tarifas”.

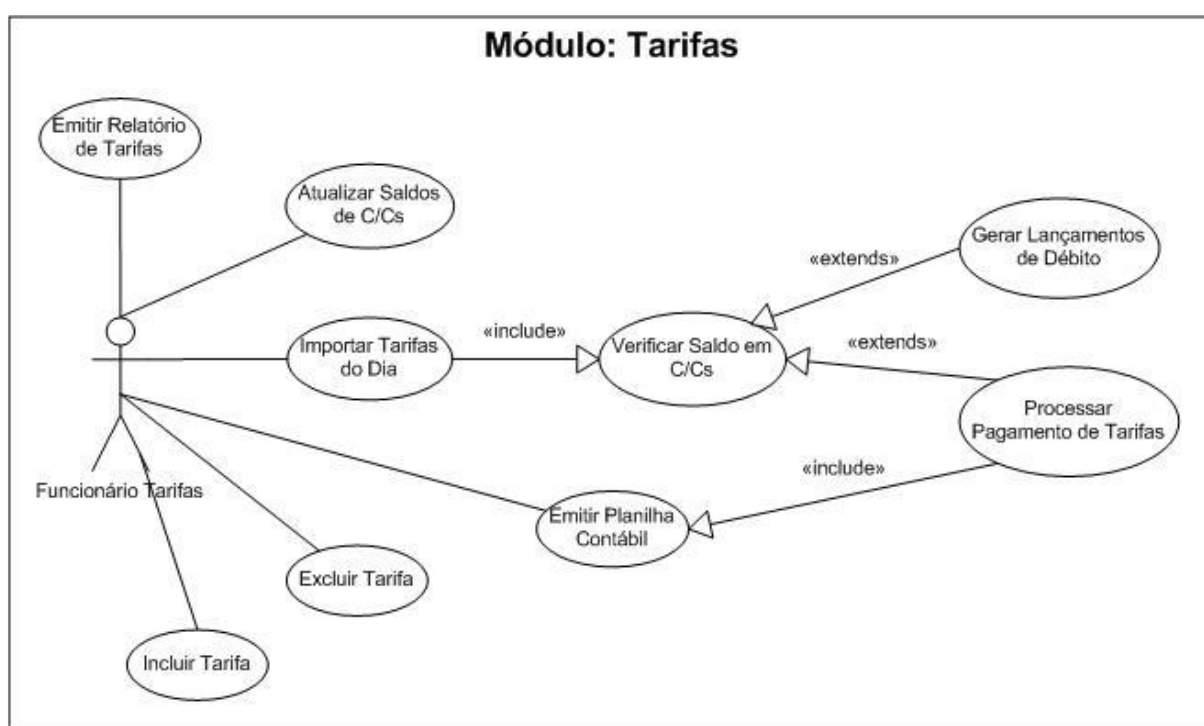


Fig. 16 - Casos de uso do Módulo Tarifas.

Quadro 19 – Descrição do Caso de Uso Importar Tarifas do Dia.

Caso de Uso: Importar Tarifas do Dia

Pré-requisitos

1. Sistema ter autenticado *login* e senha do funcionário.
2. Funcionário ter acesso ao módulo de Tarifas.
3. Funcionário ter exportado banco de dados de tarifas do SiS-Br.
4. Funcionário ter atualizado o saldo das C/Cs.

<p>Fluxo Principal</p> <ol style="list-style-type: none"> 1. Funcionário importa as tarifas do dia para o sistema. 2. Sistema verifica qual(is) C/Cs possuem saldo para pagamento de tarifas: <ul style="list-style-type: none"> - Contas que possuem saldo: Ver variante 2.1. - Contas que não possuem saldo: Ver variante 2.2.
<p>Variantes:</p> <p>Sistema processa liquidação/amortização das tarifas.</p> <p>Sistema gera planilha contábil.</p> <p>Funcionário assina planilha contábil.</p> <p>Funcionário envia planilha contábil ao setor de contabilidade.</p> <p>Sistema grava lançamento de débito para esta C/C.</p>

Quadro 20 – Descrição do Caso de Uso Atualizar Saldo das C/Cs.

Caso de Uso: Atualizar Saldo das C/Cs
<p>Pré-requisitos</p> <ol style="list-style-type: none"> 1. Sistema ter autenticado <i>login</i> e senha do funcionário. 2. Funcionário ter acesso ao módulo de Tarifas. 3. Funcionário ter exportado banco de dados de C/Cs do SiS-Br.
<p>Fluxo Principal</p> <ol style="list-style-type: none"> 1. Supervisor importa saldo das C/Cs para o sistema.

4.1.6 Módulo Comissões

A Figura 17 mostra o diagrama de caso de uso do módulo “Comissões”.

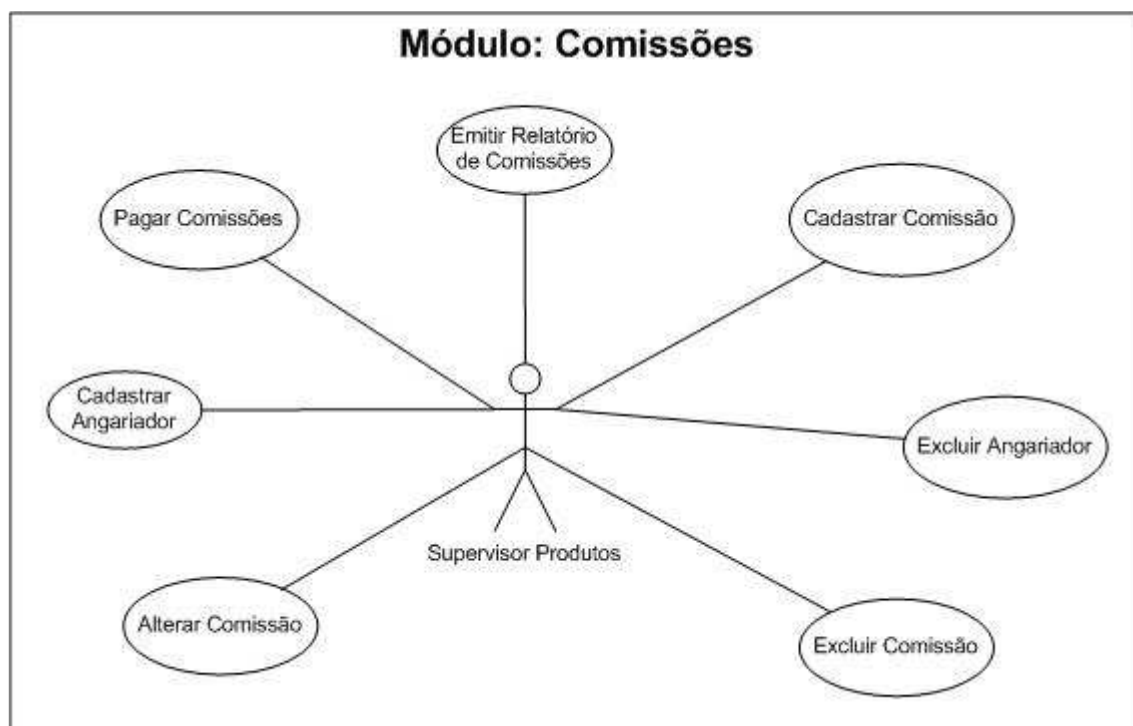


Fig. 17 - Casos de uso do Módulo Comissões.

Quadro 21 – Descrição do Caso de Uso Pagar Comissões.

Caso de Uso: Pagar Comissões
Pré-requisitos <ol style="list-style-type: none"> 1. Sistema ter autenticado <i>login</i> e senha do Supervisor de Produtos.
Fluxo Principal <ol style="list-style-type: none"> 1. Supervisor informa período data-base de referência para consulta/pagamento de comissões. 2. Supervisor processa pagamento de comissões. 3. Sistema informa operação realizada com sucesso.

Quadro 22 – Descrição do Caso de Uso Cadastrar Comissão.

Caso de Uso: Cadastrar Comissão
Pré-requisitos <ol style="list-style-type: none"> 1. Sistema ter autenticado <i>login</i> e senha do Supervisor de Produtos.

1. Supervisor seleciona angariador beneficiário da comissão.
2. Supervisor seleciona produto gerador da comissão.
3. Supervisor informa periodicidade e motivo da comissão.
4. Supervisor confirma cadastramento da comissão.
5. Sistema informa operação realizada com sucesso.

4.2 DIAGRAMA DE CLASSES

A Figura 18 mostra o diagrama de classes do SASC.

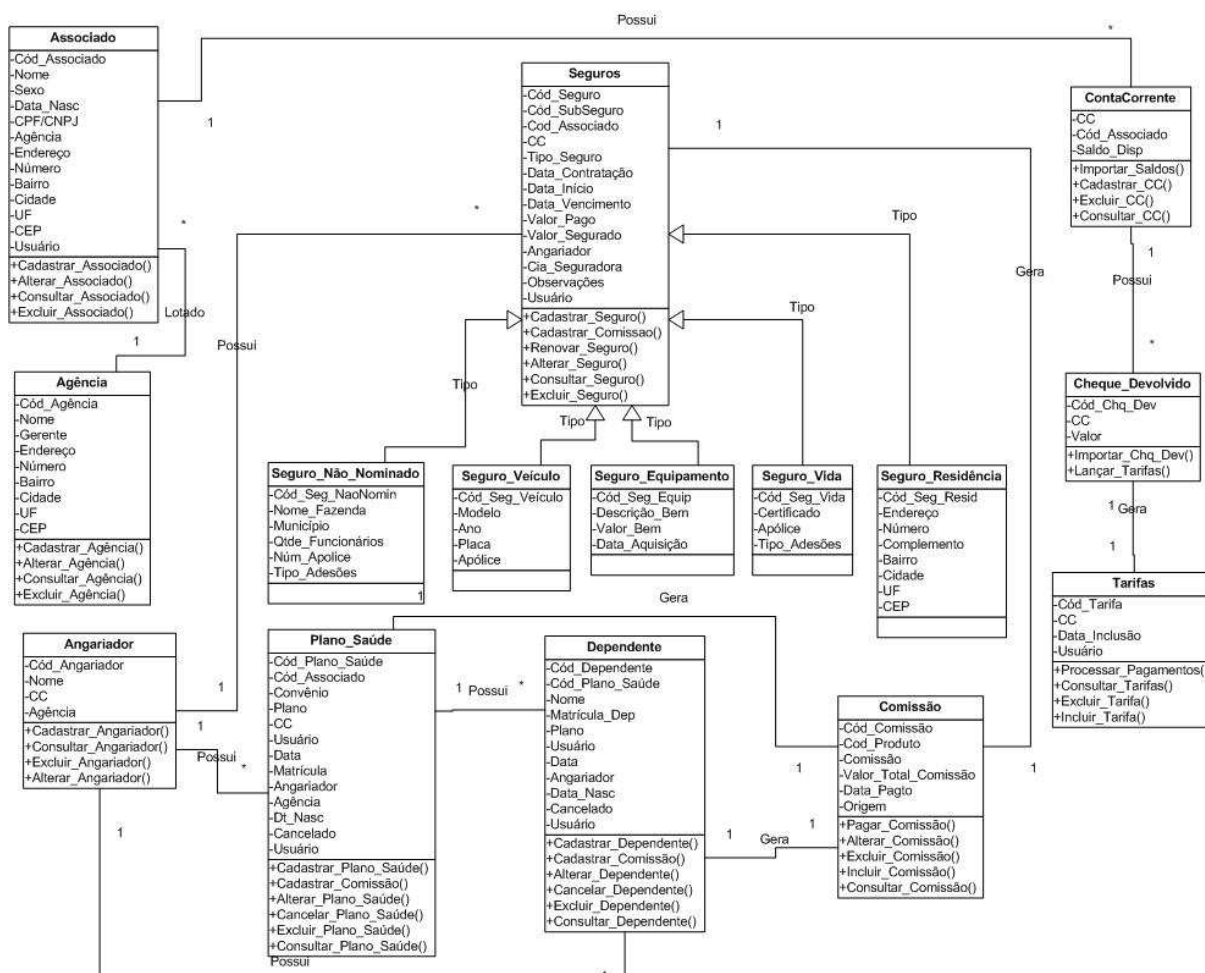


Figura 18 – Diagrama de Classes

A seguir, é apresentada a descrição sucinta de cada classe mostrada no Diagrama de Classes do SASC. Segundo Pádua Filho (2000, pg.105), a

documentação de cada classe deve conter sua definição clara e concisa, uma lista de responsabilidades e colaborações e uma lista de regras e restrições aplicáveis.

Ainda segundo Pádua Filho (2000, p.103), as **responsabilidades** representam o conhecimento e ações que possibilitam às classes cumprirem seu papel nos casos de uso. As **colaborações** representam outras classes que colaboram para o cumprimento das responsabilidades das classes já descobertas.

O Quadro 23 mostra a definição da classe “Associado”.

Quadro 23 – Definição da Classe “Associado”.

Classe: Associado
Descrição: Armazena as informações pessoais de um associado.
Responsabilidades: Realizar o cadastramento de novos associados; Visualizar e alterar dados do cadastro de um associado; Excluir cadastro de um associado.
Colaborações: Agência, ContaCorrente.
Regras e Restrições: Cada registro de associado deve estar lotado a um registro de agência.

O Quadro 24 define a classe “Agência”.

Quadro 24 – Definição da Classe “Agência”.

Classe: Agência
Descrição: Armazena as informações referentes a uma agência da Empresa X.
Responsabilidades: Realizar o cadastramento de novas agências; Visualizar e alterar dados do cadastro de uma agência; Excluir cadastro de uma agência.

O Quadro 25 define a classe “Angariador”.

Quadro 25 – Definição da Classe “Angariador”.

Classe: Angariador
<p>Descrição:</p> <p>Armazena as informações referentes a um angariador de produtos da Empresa X.</p>
<p>Responsabilidades:</p> <p>Realizar o cadastramento de novos angariadores; Visualizar e alterar dados do cadastro de um angariador; Excluir cadastro de um angariador.</p>

O Quadro 26 define a classe “Seguros”.

Quadro 26 – Definição da Classe “Seguros”.

Classe: Seguros
<p>Descrição:</p> <p>Armazena as informações referentes a um seguro vendido.</p>
<p>Responsabilidades:</p> <p>Realizar o cadastramento de novos seguros; Gerar comissão de seguros para seu angariador; Visualizar e alterar dados do cadastro de um seguro; Realizar a renovação de um seguro. Excluir cadastro de um seguro.</p>
<p>Colaborações:</p> <p>Associado, Angariador, Comissão.</p>
<p>Regras e Restrições:</p> <p>Cada registro de seguro deve estar adquirido por um registro de associado. Cada registro de seguro deve estar associado a um registro de angariador. Cada registro de seguro deve estar associado a um registro de comissão.</p>

O Quadro 27 define a classe de seguros não nominados.

Quadro 27 – Definição da Classe “Seguro_Não_Nominado”.

Classe: Seguro_Não_Nominado
<p>Descrição:</p> <p>Armazena as informações referentes a um seguro não-nominado vendido.</p>
<p>Responsabilidades:</p> <p>Realizar o cadastramento de novos seguros;</p> <p>Gerar comissão de seguros para seu angariador;</p> <p>Visualizar e alterar dados do cadastro de um seguro;</p> <p>Realizar a renovação de um seguro.</p> <p>Excluir cadastro de um seguro.</p>
<p>Colaborações:</p> <p>Seguros.</p>
<p>Regras e Restrições:</p> <p>Cada registro de Seguro_não_Nominado deve estar associado a um registro de Seguros.</p>

O Quadro 28 define a classe de seguros de veículos.

Quadro 28 – Definição da Classe “Seguro_Veículos”.

Classe: Seguro_Veículos
<p>Descrição:</p> <p>Armazena as informações referentes a um seguro de veículo vendido.</p>
<p>Responsabilidades:</p> <p>Realizar o cadastramento de novos seguros;</p> <p>Gerar comissão de seguros para seu angariador;</p> <p>Visualizar e alterar dados do cadastro de um seguro;</p> <p>Realizar a renovação de um seguro.</p> <p>Excluir cadastro de um seguro.</p>
<p>Colaborações:</p> <p>Seguros.</p>
<p>Regras e Restrições:</p> <p>Cada registro de Seguro_Veículos deve estar associado a um registro de Seguros.</p>

O Quadro 29 define a classe de seguros de equipamentos.

Quadro 29 – Definição da Classe “Seguro_Equipamento”.

Classe: Seguro_Equipamento
<p>Descrição:</p> <p>Armazena as informações referentes a um seguro de equipamento vendido.</p>
<p>Responsabilidades:</p> <p>Realizar o cadastramento de novos seguros;</p> <p>Gerar comissão de seguros para seu angariador;</p> <p>Visualizar e alterar dados do cadastro de um seguro;</p> <p>Realizar a renovação de um seguro.</p> <p>Excluir cadastro de um seguro.</p>
<p>Colaborações:</p> <p>Seguros.</p>
<p>Regras e Restrições:</p> <p>Cada registro de Seguro_Equipamento deve estar associado a um registro de Seguros.</p>

O Quadro 30 define a classe de seguros de vida.

Quadro 30 – Definição da Classe “Seguro_Vida”.

Classe: Seguro_Vida
<p>Descrição:</p> <p>Armazena as informações referentes a um seguro de vida vendido.</p>
<p>Responsabilidades:</p> <p>Realizar o cadastramento de novos seguros;</p> <p>Gerar comissão de seguros para seu angariador;</p> <p>Visualizar e alterar dados do cadastro de um seguro;</p> <p>Realizar a renovação de um seguro.</p> <p>Excluir cadastro de um seguro.</p>
<p>Colaborações:</p> <p>Seguros.</p>
<p>Regras e Restrições:</p> <p>Cada registro de Seguro_Vida deve estar associado a um registro de Seguros.</p>

O Quadro 31 define a classe de seguros residenciais.

Quadro 31 – Definição da Classe “Seguro_Residência”.

Classe: Seguro_Residência
<p>Descrição:</p> <p>Armazena as informações referentes a um seguro residencial vendido.</p>
<p>Responsabilidades:</p> <p>Realizar o cadastramento de novos seguros;</p> <p>Gerar comissão de seguros para seu angariador;</p> <p>Visualizar e alterar dados do cadastro de um seguro;</p> <p>Realizar a renovação de um seguro.</p> <p>Excluir cadastro de um seguro.</p>
<p>Colaborações:</p> <p>Seguros.</p>
<p>Regras e Restrições:</p> <p>Cada registro de Seguro_Residência deve estar associado a um registro de Seguros.</p>

O Quadro 32 define a classe dos planos de saúde.

Quadro 32 – Definição da Classe “Plano_Saúde”.

Classe: Plano_Saúde
<p>Descrição:</p> <p>Armazena as informações referentes a um plano de saúde vendido.</p>
<p>Responsabilidades:</p> <p>Realizar o cadastramento de novos planos de saúde;</p> <p>Gerar comissão do plano de saúde para seu angariador;</p> <p>Visualizar e alterar dados do cadastro de um plano de saúde;</p> <p>Realizar o cancelamento de um plano de saúde;</p>
<p>Colaborações:</p> <p>Associado, Angariador, Dependentes, Comissão.</p>

Regras e Restrições:

Cada registro de plano de saúde deve estar adquirido por um registro de associado.

Cada registro de plano de saúde deve estar associado a um registro de angariador.

Cada registro de plano de saúde deve estar associado a um registro de comissão.

O Quadro 33 define a classe de dependentes de planos de saúde.

Quadro 33 – Definição da Classe “Dependente”.

Classe: Dependente
Descrição: Armazena as informações referentes a um dependente de plano de saúde.
Responsabilidades: Realizar o cadastramento de novos dependentes; Gerar comissão do plano para seu angariador; Visualizar e alterar dados do cadastro de um dependente; Realizar o cancelamento do plano de um dependente.
Colaborações: Plano_Saúde, Angariador, Comissão.
Regras e Restrições: Cada registro de dependentes deve estar associado a um registro de Plano_Saúde. Cada registro de dependentes deve estar associado a um registro de angariador. Cada registro de plano de saúde deve estar associado a um registro de comissão.

O Quadro 34 define a classe das comissões geradas na venda de algum produto.

Quadro 34 – Definição da Classe “Comissão”.

Classe: Comissão
<p>Descrição:</p> <p>Armazena as informações referentes a uma comissão gerada pela venda de um produto.</p>
<p>Responsabilidades:</p> <p>Processar o pagamento da comissão; Consultar e alterar dados da comissão; Cadastrar novas comissões; Excluir comissões cadastradas.</p>
<p>Colaborações:</p> <p>Angariador, Seguros, Plano_Saúde, Dependente.</p>
<p>Regras e Restrições:</p> <p>Cada registro de comissão deve estar associado a um registro de angariador. Cada registro de comissão deve identificar o produto que a gerou.</p>

O Quadro 35 define a classe de tarifas em atraso.

Quadro 35 – Definição da Classe “Tarifas”.

Classe: Tarifas
<p>Descrição:</p> <p>Armazena as informações referentes a uma tarifa em atraso.</p>
<p>Responsabilidades:</p> <p>Processar pagamento de tarifas. Consultar, excluir ou incluir tarifas.</p>
<p>Colaborações:</p> <p>Cheque_Devolvido.</p>

O Quadro 36 define a classe “Cheque_Devolvido”.

Quadro 36 – Definição da Classe “Cheque_Devolvido”.

Classe: Cheque_Devolvido
Descrição: Armazena as informações referentes a um cheque devolvido.
Responsabilidades: Importar cheques devolvidos no dia; Gerar tarifa em atraso.
Colaborações: ContaCorrente, Tarifas.
Regras e Restrições: Cada registro de Cheque_Devolvido deve estar associado a um registro em ContaCorrente.

O Quadro 37 define a classe de contas correntes.

Quadro 37 – Definição da Classe “ContaCorrente”.

Classe: ContaCorrente
Descrição: Armazena as informações referentes a uma conta-corrente.
Responsabilidades: Importar saldo disponível da conta-corrente; Visualizar e excluir conta-corrente; Cadastrar nova conta-corrente;
Colaborações: Associado
Regras e Restrições: Cada registro de ContaCorrente deve estar associado a um registro em Associado.

4.3 DIAGRAMAS DE SEQÜÊNCIA

Segundo Pádua Filho (2000, p.398), os diagramas de seqüência devem enfatizar o ordenamento temporal das ações. Para isto, é preferível usar diagramas separados para representar roteiros resultantes de diferentes caminhos lógicos. As Figuras 19, 20, 21 e 22 mostram os diagramas de seqüência das principais ações dos módulos do SASC.

A Figura 19 mostra o diagrama de seqüência do módulo “*Plano de Saúde*” para cadastramento de novos planos.

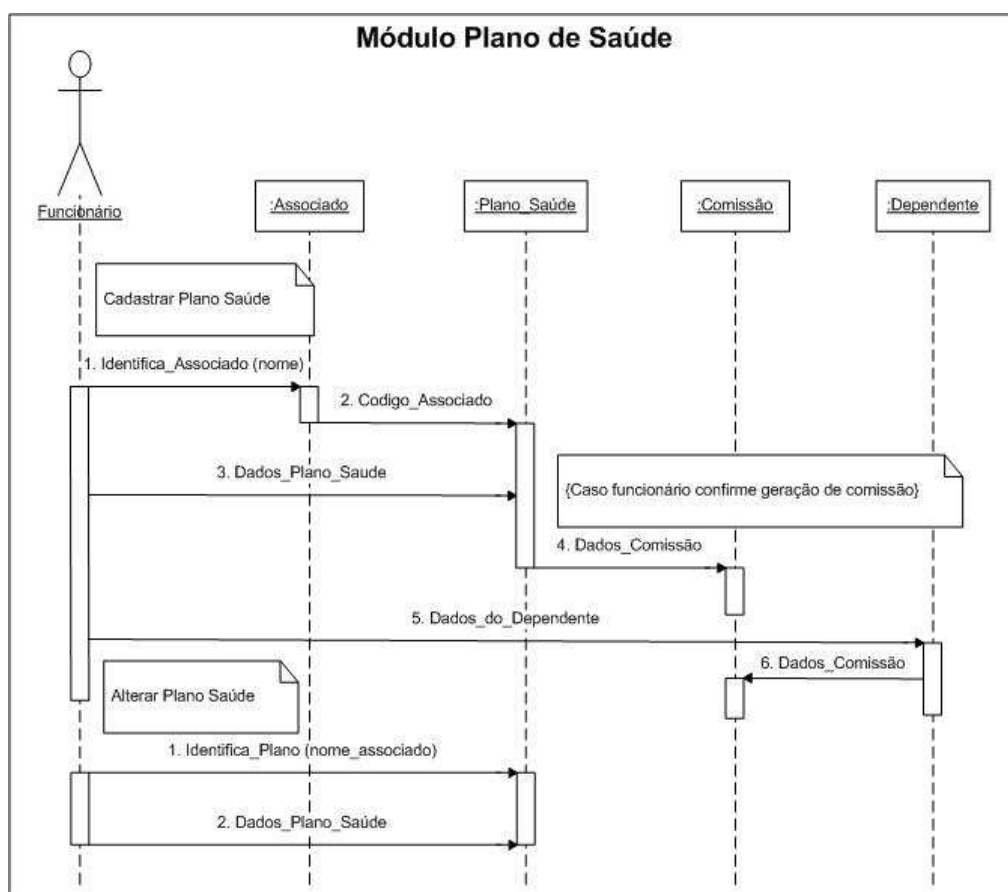


Figura 19 – Diagrama de Seqüência do Módulo Plano de Saúde

A Figura 20 mostra o diagrama de seqüência do módulo “*Plano de Saúde*” para fechamento de fatura e geração de lançamentos de débito.

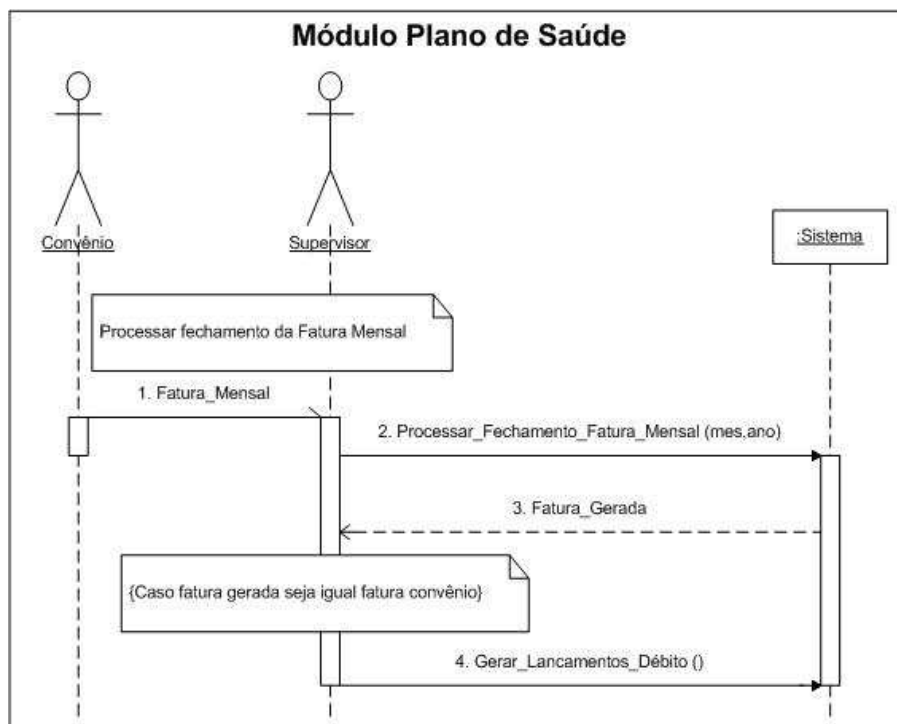


Figura 20 – Diagrama de Seqüência do Módulo Plano de Saúde

A Figura 21 mostra o diagrama de seqüência do módulo “Seguros”.

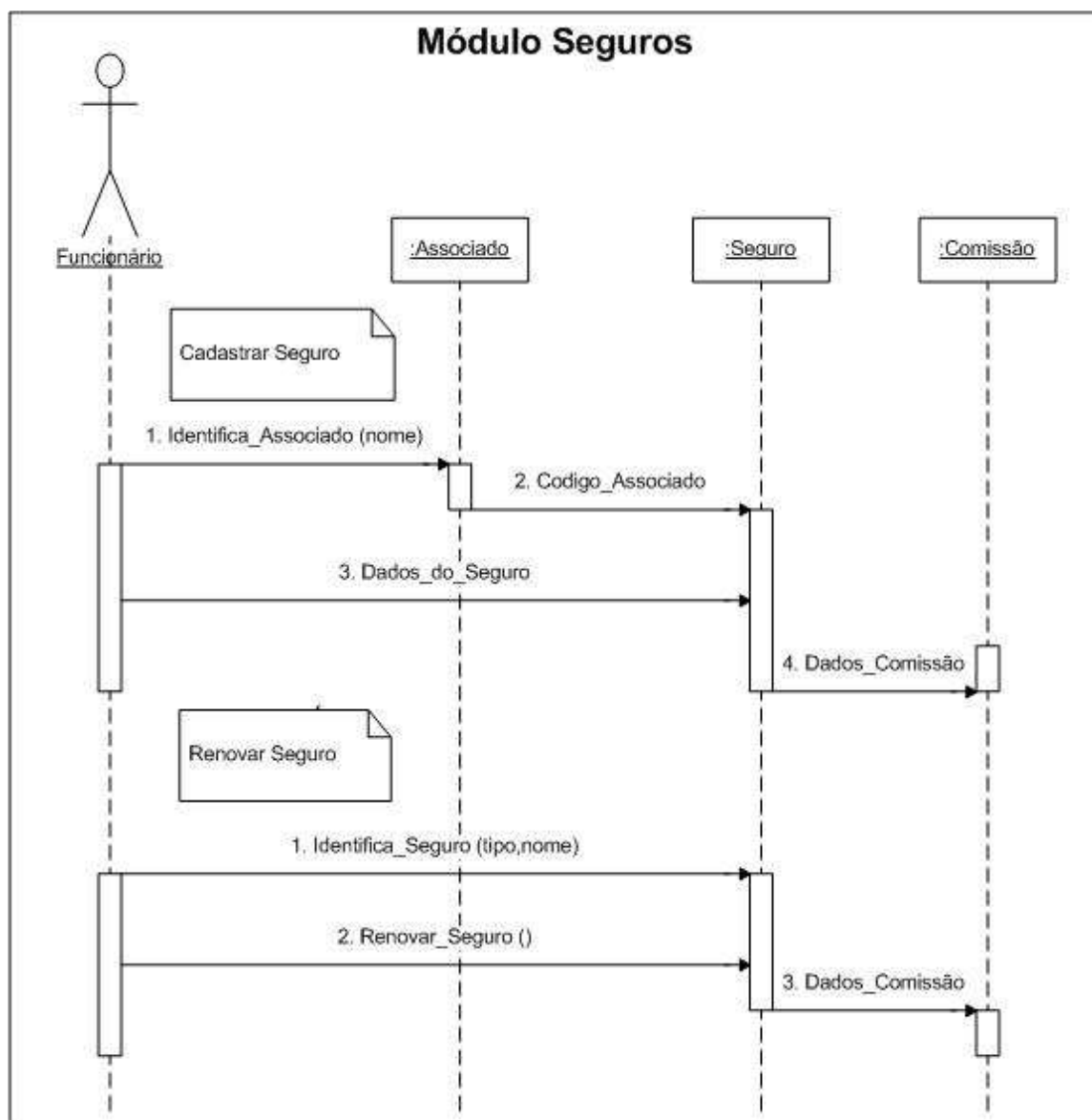


Figura 21 – Diagrama de Seqüência do Módulo Seguros

A Figura 22 mostra o diagrama de seqüência do módulo “*Tarifas*”.

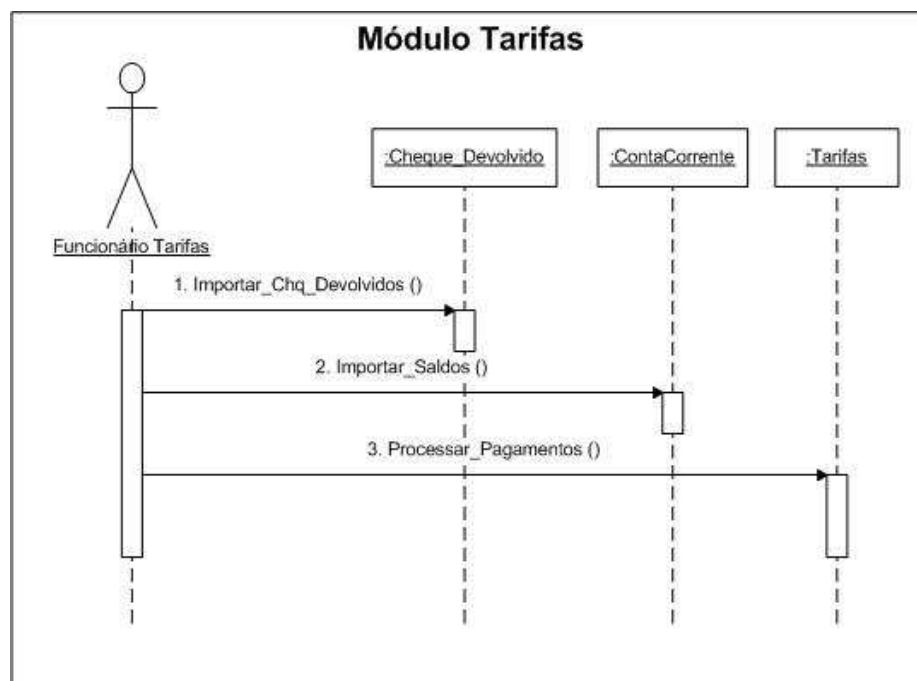


Figura 22 – Diagrama de Seqüência do Módulo Tarifas.

4.4 DIAGRAMAS DE ESTADO

Segundo Pádua Filho (2000, p.381), os diagramas de estado são utilizados para descrever fluxos de lógica complexa ou com muitos detalhes. O Diagrama de estado permite visualizar a execução do caso de uso de forma mais abrangente que os roteiros textuais, embora menos detalhada. Apresentamos a seguir, os diagramas de estados dos módulos do SASC.

4.4.1 Módulo Plano de Saúde

A Figura 23 mostra o diagrama de estado de navegação do módulo “Plano de Saúde”.

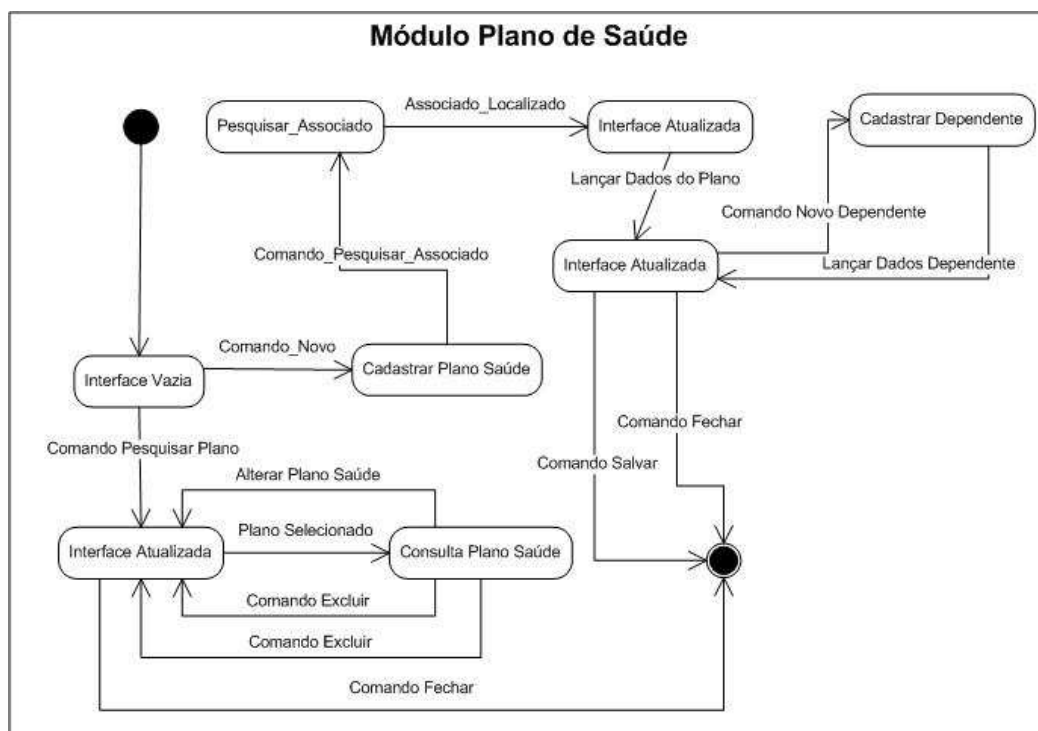


Figura 23 – Diagrama de Estado Módulo Plano de Saúde

4.4.2 Módulo Seguros

A Figura 24 mostra o diagrama de estado de navegação do módulo “Seguros”.

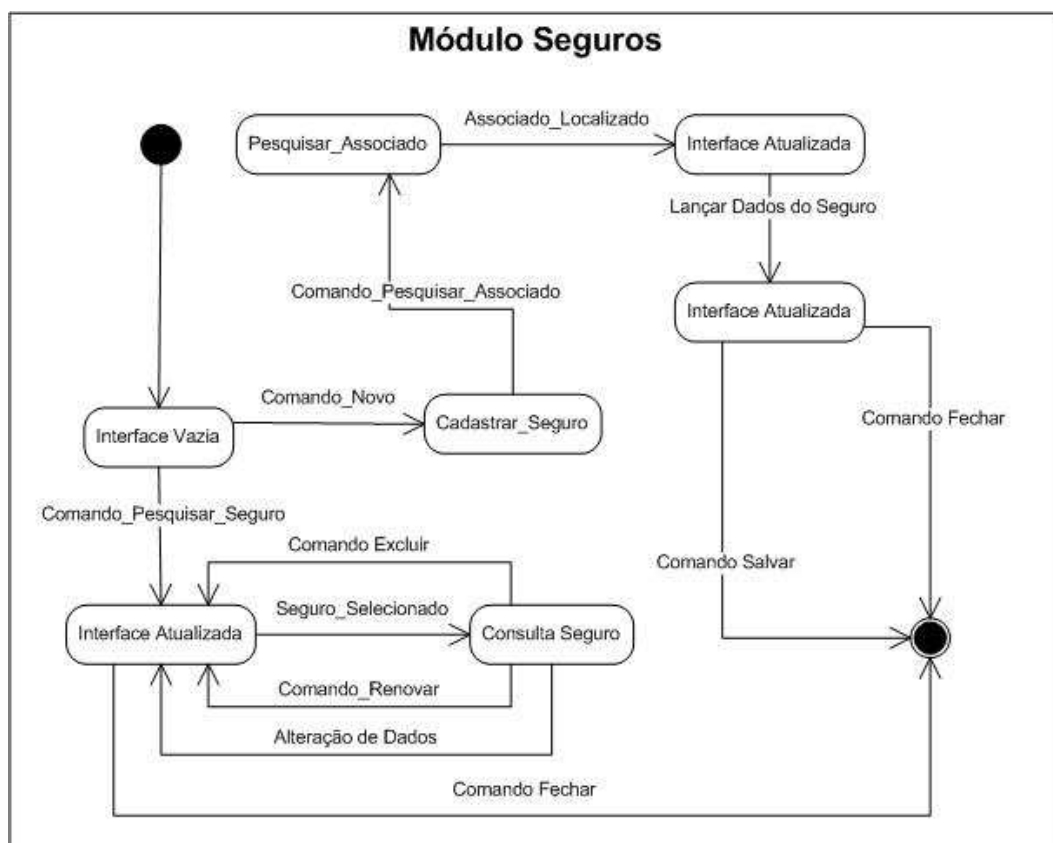


Figura 24 – Diagrama de Estado Módulo Seguros

4.4.3 Módulo Comissões

A Figura 25 mostra o diagrama de estado de navegação do módulo “Comissões”.

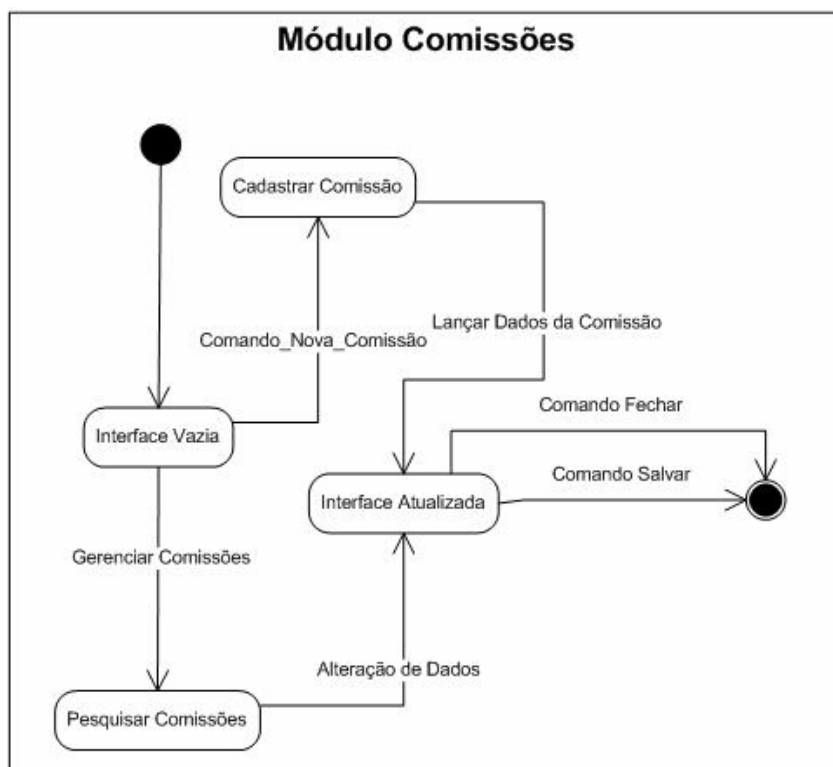


Figura 25 – Diagrama de Estado Módulo Comissões.

4.4.4 Módulo Administração do Sistema

A Figura 26 mostra o diagrama de estado de navegação do módulo “Administração do Sistema”.

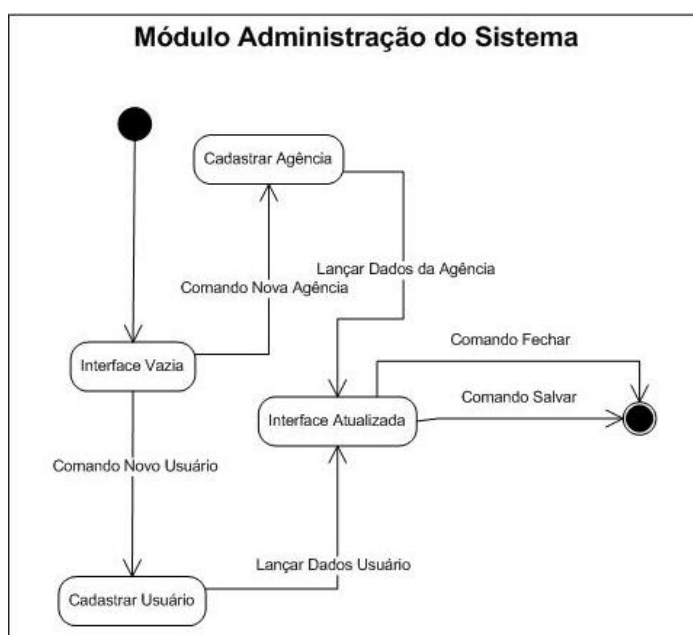


Figura 26 – Diagrama de Estado Módulo Administração do Sistema

4.4.5 Módulo Associados

A Figura 27 mostra o diagrama de estado de navegação do módulo “Associados”.

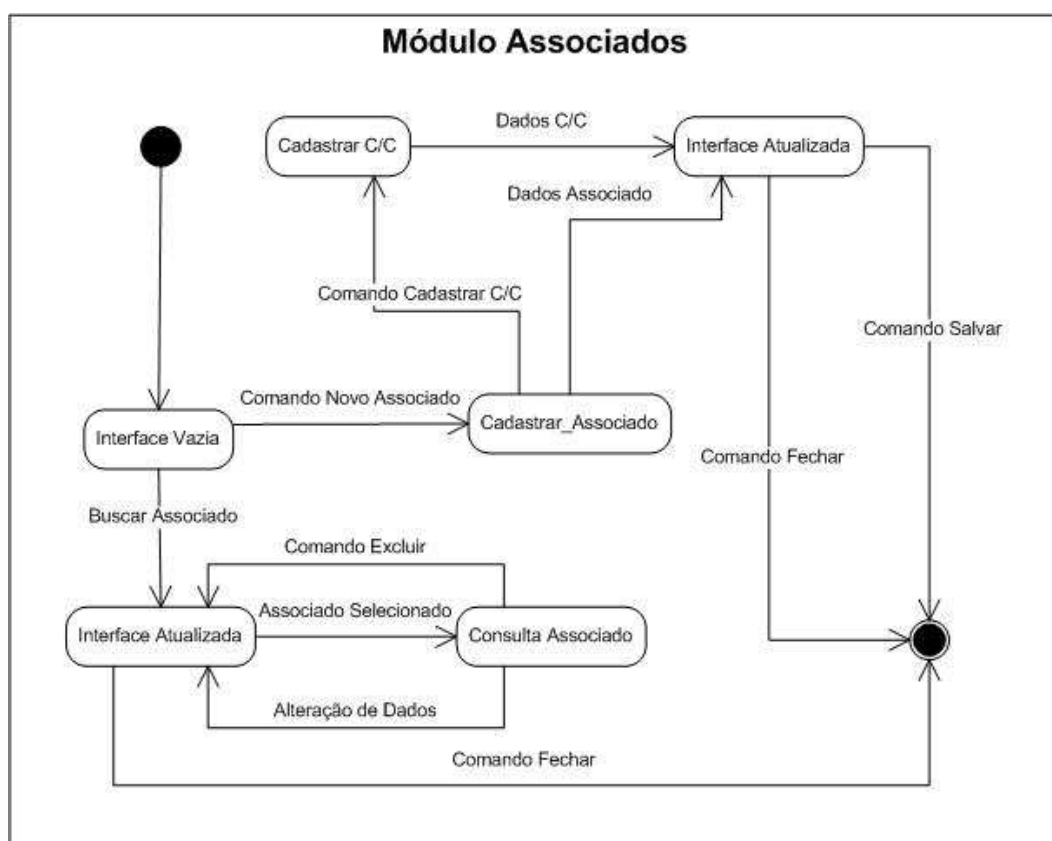


Figura 27 – Diagrama de Estado Módulo Associados

O próximo capítulo descreve o software SASC, mostrando suas telas e as tecnologias adotadas para o desenvolvimento do sistema.

5 IMPLEMENTAÇÃO DO SISTEMA

5.1 TECNOLOGIAS ADOTADAS

Para o desenvolvimento do SASC foi adotada a ferramenta de programação Borland Delphi 5. Esta ferramenta além de utilizar uma linguagem baseada na programação orientada a objetos - *Object Pascal*, possui um vasto repositório de componentes reutilizáveis uma vez que é amplamente utilizado por programadores em todo mundo.

O sistema gerenciador de banco de dados utilizado no SASC foi o MySQL. Pode-se citar como algumas de suas vantagens:

- Utilização da linguagem SQL como interface.
- SGBD *Open-Source* mais popular do mundo.
- Existência de componentes Delphi para acesso nativo ao SGBD.
- Excelente desempenho e estabilidade.
- Pouco exigente a recursos de hardware.
- Multi-tarefa e Multi-usuário.
- Controle tranSASCional - Transações ACID.
- Implementa *Stored Procedures*, *Views* e *Triggers*.

5.2 DESCRIÇÃO DO SISTEMA

A Figura 28 mostra a tela de abertura do SASC. O acesso ao sistema somente é permitido após autenticação de senha do usuário.



Fig. 28 - Tela de Login do SASC

Ao clicar em “OK” o sistema tentará autenticar o login e senha do usuário e, ocorrendo validação, será aberta a janela principal do SASC, mostrada na Figura 29 abaixo.

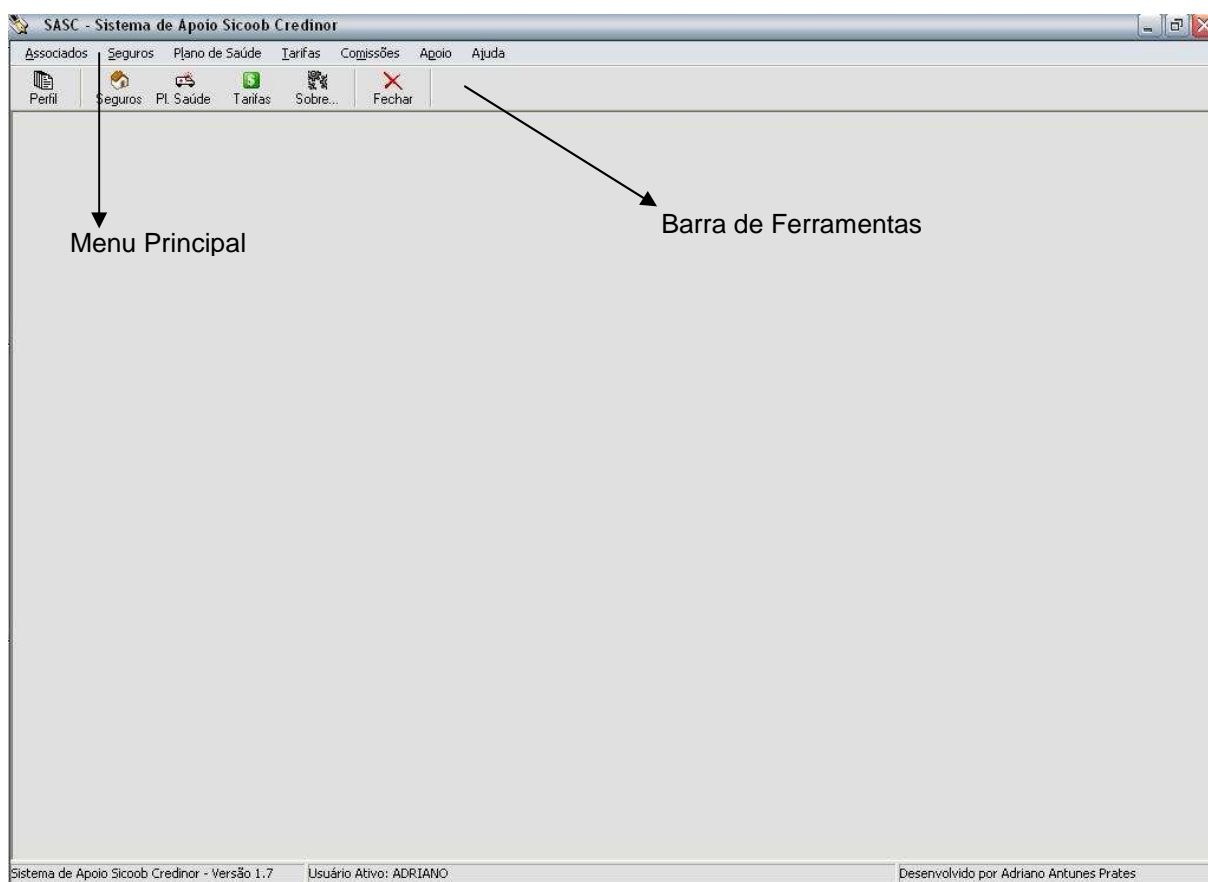



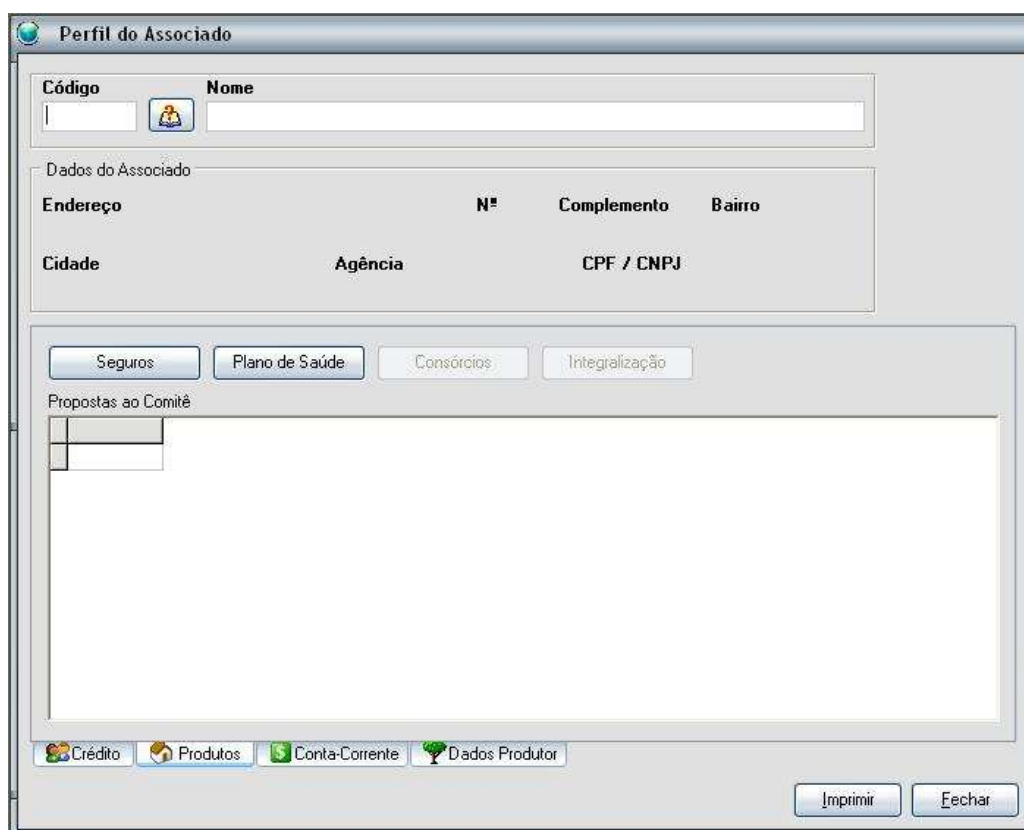
Figura 29 – Janela Principal do SASC

A janela principal do SASC busca dar acesso rápido às principais funcionalidades do sistema através da barra de ferramentas.

O menu principal, para simplificar o acesso dos usuários, foi dividido de acordo com os módulos existentes no sistema.

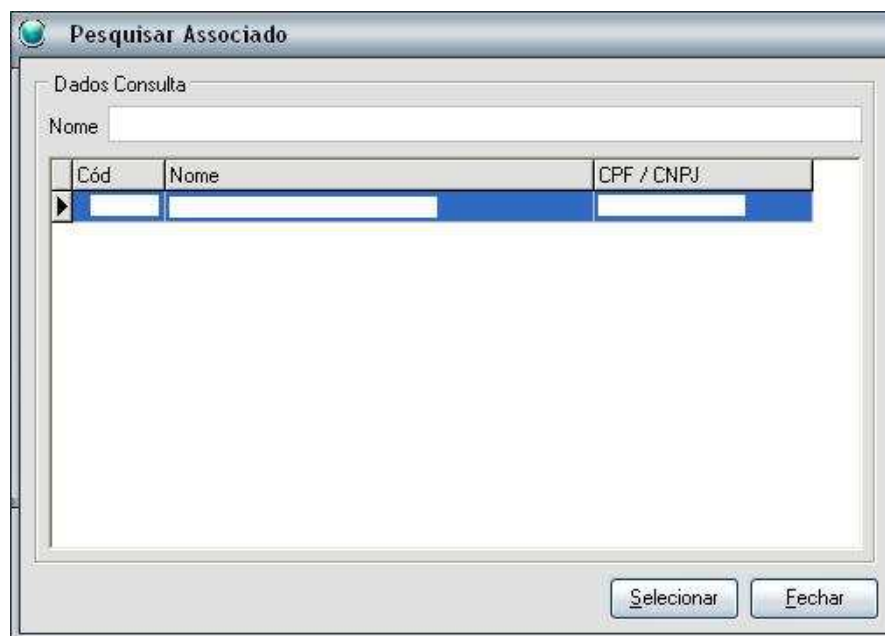
Apresentamos a seguir as janelas filhas acessadas a partir da barra de ferramentas, começando pelo botão “Perfil” (Figura 29).

Na janela “Perfil do Associado”, o usuário poderá exibir e imprimir um relatório-resumo do associado, contendo informações relativas a seu cadastro e produtos e serviços adquiridos pelo mesmo. A seleção do associado é realizada por meio do botão  que, ao ser clicado, abre a janela de seleção do associado, mostrada na Figura 30.



Endereço	Nº	Complemento	Bairro
Cidade	Agência	CPF / CNPJ	

Figura 30 – Janela para impressão do “Perfil do Associado”



Pesquisar Associado

Dados Consulta

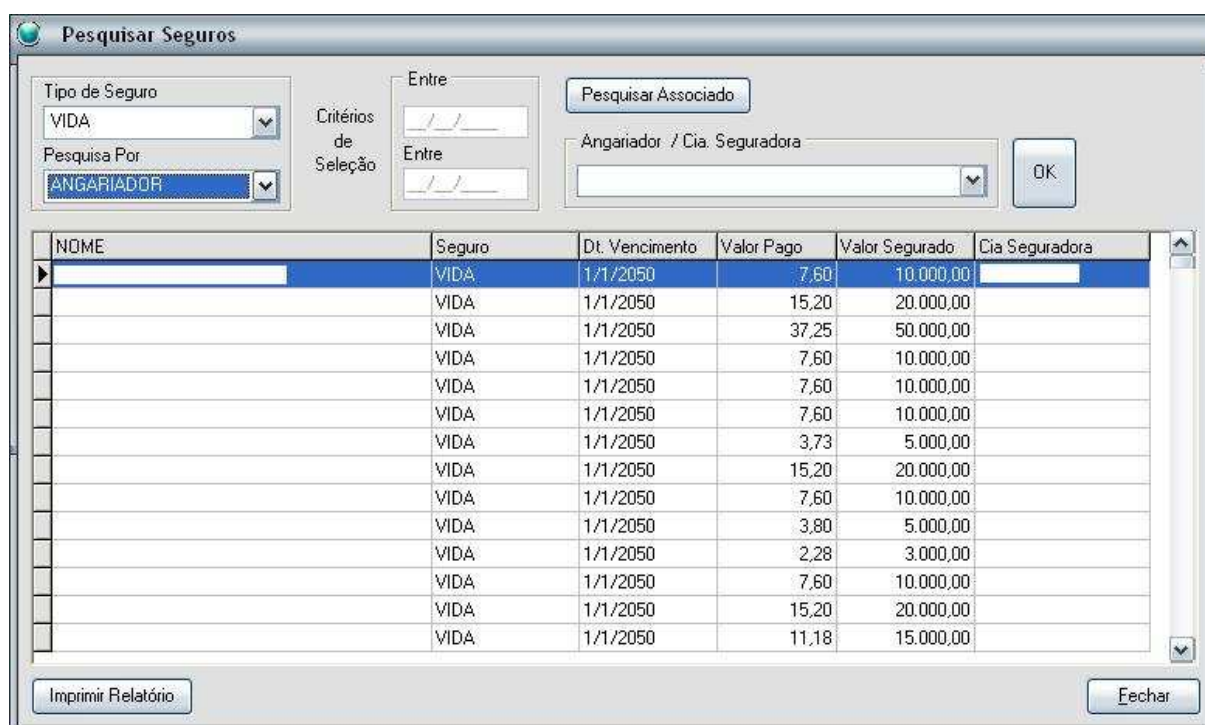
Nome:

Cód	Nome	CPF / CNPJ
▶		

Selecionar Fechar

Figura 31 – Janela para seleção de Associado

Voltando à janela principal do SASC, e selecionando o segundo botão da barra de ferramentas - “Seguros”, abrimos o formulário de pesquisa de seguros, mostrada na Figura 32.



Pesquisar Seguros

Tipo de Seguro: VIDA

Pesquisa Por: ANGARIADOR

Entre:

Entre:

Pesquisar Associado

Angariador / Cia. Seguradora:

OK

NOME	Seguro	Dt. Vencimento	Valor Pago	Valor Segurado	Cia Seguradora
▶	VIDA	1/1/2050	7,60	10.000,00	
	VIDA	1/1/2050	15,20	20.000,00	
	VIDA	1/1/2050	37,25	50.000,00	
	VIDA	1/1/2050	7,60	10.000,00	
	VIDA	1/1/2050	7,60	10.000,00	
	VIDA	1/1/2050	7,60	10.000,00	
	VIDA	1/1/2050	3,73	5.000,00	
	VIDA	1/1/2050	15,20	20.000,00	
	VIDA	1/1/2050	7,60	10.000,00	
	VIDA	1/1/2050	3,80	5.000,00	
	VIDA	1/1/2050	2,28	3.000,00	
	VIDA	1/1/2050	7,60	10.000,00	
	VIDA	1/1/2050	15,20	20.000,00	
	VIDA	1/1/2050	11,18	15.000,00	

Imprimir Relatório

Fechar

Figura 32 – Janela de consulta de seguros

A janela “Pesquisar Seguros” oferece vários critérios de filtros para facilitar a localização de determinado seguro ou grupo de seguros, como por exemplo, filtragem por tipo de seguro, angariador, data de vencimento, data de contratação, cia. seguradora ou nome de associado. Além de permitir a impressão de relatório com o resultado destes filtros.

Pode-se também consultar todos os dados relativos a um seguro, bastando para isso efetuar um duplo clique sobre o registro escolhido, desta forma, será aberta a janela de consulta de seguros referente àquela modalidade de seguro. Na Figura 33 mostramos a janela de consulta seguro da modalidade “não-nominado”.

Consultar Seguro Não Nominado

Código: 28592 Nome: _____

Dados do Seguro Não-Nominado

Nome da Fazenda: _____ Município: _____ Agência: JANUÁRIA

Vigência: Data Início: 24/ 7 /2007 Data Vencimento: 24/ 7 /2008

Valor Anual: 153,35 Valor Cobertura: 0 Qtde. Funcionários: _____ Dt. Contratação: 29/ 8 /2007

Cia. Seguradora: _____ Apólice: A000006203865-8

Abrangência: ☐ Morte ☐ IPA ☐ IEA ☐ IPD ☐ DMHO ☐ IFPD ☐ Auxílio Funeral

Angariador: _____ Comissão: 40

Observações: _____

Renovar Excluir Alterar Salvar Fechar

Figura 33 – Consulta seguro não-nominado

A partir da consulta de um seguro, pode-se realizar diversas ações sobre o mesmo (se o usuário autenticado tiver acesso para tal), como por exemplo, realizar a renovação do seguro, exclusão do seguro e alterações de dados.

Voltando à janela principal do SASC (Figura 29) e selecionando o terceiro botão da barra de ferramentas: “Pl. Saúde”, conseguimos visualizar o formulário de controle de Planos de Saúde, mostrada na Figura 34.

Pesquisar Planos de Saúde

Filtrar Planos de Saúde

Nome: ADR

Filtrar Planos: Filtrar Convênios:

Contratado em

30/12/1899 19/5/2008

Plano de Saúde

Nome	Convênio	Plano	Matrícula	Data
			033144-9	01/12/2004
			028966-3	01/03/2004
			044230-5	10/01/2008

Figura 34 – Formulário para consulta dos planos de saúde.

A partir deste formulário pode-se realizar a consulta dos planos de saúde (titulares e dependentes) através de filtragens por data de contratação, tipo de planos e tipo de convênio e, a partir dos resultados obtidos, gerar relatórios.

Ao executar um duplo clique em um registro de plano de saúde, abre-se um formulário para exibição dos dados analíticos deste. Este formulário é mostrado na Figura 35 e, além da alteração dos dados do plano, é possível a inclusão ou exclusão de dependentes do plano ou mesmo a exclusão do plano como um todo.

Consultar Plano de Saúde

Código: 10286 Nome: _____

Plano Consultas Extras

Dados do Plano de Saúde

Convênio: ☒ _____ ☐ _____ ☐ _____

Plano: _____ Agência: MONTES CLAROS

Data Contratação: 01/03/2004 Dt. Nascimento: 07/06/1965 Matrícula: 028963-9 C/C Débito: 11002

Angariador: _____ Mensalidade: 126,05 ☐ Plano Inativo **Salvar Alterações**

Dependentes

Nome	Matricula
▶ _____	028966-3
_____	028970-1
_____	028971-0
_____	028969-8

Inserir Dependente **Remover Dependente**

Excluir Plano de Saúde **Fechar**

Figura 35 – Consulta de um Plano de Saúde cadastrado.

A Figura 36 mostra a tela para visualização e impressão do extrato de tarifas do SASC. Esta tela pode ser acessada através da barra de ferramentas, na janela principal, botão “Tarifas”. Pode-se também, através deste formulário – botão “Lançar Pagamento”, efetuar lançamentos manuais de crédito para amortização ou liquidação de tarifas.

The screenshot shows a window titled "Gerenciamento de Tarifas". Inside, there is a section labeled "Pesquisar Tarifas" with two input fields: "C/C" and "Nome". Below these fields is a large empty table area. At the bottom left, it says "Total Devedor: 0". At the bottom right, there are three buttons: "Lançar Pagamento", "Imprimir Extrato", and "Fechar".

Figura 36 – Consulta de um Plano de Saúde cadastrado.

Ao clicarmos em “Lançar Pagamento” abre-se a janela mostrada na Figura 37 para lançamento manual de pagamentos de tarifas em atraso.

The screenshot shows a window titled "Lançar Pagamento de Tarifa". Inside, there is a section labeled "Liquidar Tarifa" with two input fields: "Nome" and "C/C". Below these fields, there are two more fields: "Total da Amortização" (with a value of 0,00) and "Data" (with a value of 26/ 5 /2008 and a dropdown arrow). At the bottom right, there are two buttons: "Processar" and "Fechar".

Figura 37 – Lançamento manual de Crédito.

Entretanto, como visto na fase de análise, o processo de importação e pagamento de tarifas é automatizado pelo SASC. Após a importação dos cheques devolvidos no dia, e dos saldos de conta-corrente atualizados, o sistema processa as tarifas que poderão ser pagas e as que serão armazenadas para posterior pagamento. Este processo pode ser acessado através da janela principal em *Tarifas > Processar Tarifas do Dia*. A Figura 38 mostra o formulário deste processo.

Saldos Processados...

Tarifas a D... Tarifas a Estornar

Tarifas pendentes, com saldo disponível em C/C: (DEBITAR)

C/C	Nome	Total
▶		

***Duplo Clique para Retirar Tarifa do Relatório.**

Imprimir Planilha de Digitação Baixar Tarifas do Sistema

Última Atualização de Saldo: 07/04/2008 16:20:10

Fechar

Figura 38 – Formulário para baixa/inclusão automática das tarifas em atraso.

A Figura 39 mostra o formulário para gerenciamento e controle das comissões geradas automaticamente pela venda de produtos da Empresa X.

Gerenciamento de Comissões

Filtrar

Origem: Período de: 26/ 5 /2008 a 26/ 5 /2008

Pagar Comissões

Data de Pagamento: 26/ 5 /2008

Comissões

Angariador	Associado / Titular	Data Base	Valor Produto	Com.	Com. Angariador	Origem
▶		20/04/2007	R\$ 0,00	00%	R\$ 0,00	PLS
		20/04/2007	R\$ 0,00	00%	R\$ 0,00	PLS
		14/09/2007	R\$ 35,63	25%	R\$ 8,91	PLS
		14/09/2007	R\$ 35,63	25%	R\$ 8,91	PLS
		19/12/2007	R\$ 86,58	25%	R\$ 21,64	PLS
		19/12/2007	R\$ 86,58	25%	R\$ 21,64	PLS
		27/07/2007	R\$ 66,73	25%	R\$ 16,68	PLS
		14/01/2008	R\$ 0,00	00%	R\$ 0,00	PLS
		14/01/2008	R\$ 0,00	00%	R\$ 0,00	PLS
		14/01/2008	R\$ 66,73	25%	R\$ 16,68	PLS
		08/01/2008	R\$ 38,73	30%	R\$ 0,00	PLS
		08/01/2008	R\$ 38,73	30%	R\$ 0,00	PLS
		31/07/2007	R\$ 258,83	25%	R\$ 64,71	PLS

Imprimir Extrato de Comissões

Fechar

Figura 39 – Formulário para Gerenciamento e Controle de Comissões Geradas.

Ao efetuarmos um duplo - clique sobre um registro de comissão, abre-se uma janela para consulta dos dados, alteração ou exclusão desta comissão. Esta janela pode ser observada na Figura 40.

Figura 40 – Formulário para consulta de Comissão.

O cadastramento de novos associados é realizado através do formulário mostrado na Figura 41 e que pode ser acessado através da janela principal em *Associados > Novo Associado*.

Figura 41 – Formulário para cadastramento de Associados.

A manutenção do cadastro de associados é realizado através do formulário de manutenção, acessado em *Associados > Manutenção de Cadastros*.

O cadastramento de novos usuários do SASC é realizado através do formulário mostrado na Figura 42, e que pode ser acessado através da janela principal em *Apoio > Usuários > Novo Usuário*.

Figura 42 – Formulário para cadastramento de Usuários.

A manutenção do cadastro de usuários e alterações de permissões podem ser acessadas através de formulário específico, em *Apoio > Usuários > Alterar Usuário*.

5.3 PADRONIZAÇÃO DE RELATÓRIOS

Os relatórios do SASC foram desenvolvidos com auxílio de componentes específicos para geração de relatórios da biblioteca *freeware "Fortes Report"* e foram padronizados conforme mostra a Figura 43.

Figura 43 – Modelo Padrão de Relatórios do SASC.

O próximo capítulo conclui o trabalho, apresentando os resultados obtidos, dificuldades encontradas durante sua realização e trabalhos futuros.

6 CONCLUSÕES

6.1 RESULTADOS OBTIDOS

O resultado obtido através da realização deste trabalho foi o desenvolvimento de um sistema de informação para o controle de diversos processos e rotinas administrativas da empresa contratante denominada Empresa X, que antes não eram informatizadas e que agora podem contar com um sistema desenvolvido para dar suporte e gerenciar as informações, conforme análise de requisitos realizada através de entrevistas com gerentes e funcionários da empresa (vide Anexo A).

Este sistema de informação fora denominado SASC – Sistema de Apoio Empresa X, e em sua primeira versão focou aos setores e processos priorizados pelo gerente geral da empresa, que em sua maioria referem-se à venda de produtos bancários oferecidos pela mesma, como planos de saúde e seguros, e de determinados controles de débitos e pagamentos, como por exemplo, a cobrança de tarifas em atraso.

O sistema possui também um controle de permissões de acesso vinculado à conta de cada usuário do software, beneficiando assim a segurança e confiabilidade da base de dados.

Desta forma, através da realização deste trabalho, concluímos que a utilização eficiente da informação é parte indispensável da prática cotidiana da Empresa X, sendo componente vital ao êxito de suas atividades.

Diante disto, pode-se responder à questão de pesquisa afirmando que o sistema de informação desenvolvido pode contribuir para a melhor gestão e controle dos processos da Empresa X através da automatização de tarefas, prevenção e tratamento de erros e informações inconsistentes, geração de relatórios de acompanhamento e índices de produtividade.

6.2 DIFICULDADES ENCONTRADAS

No decorrer do desenvolvimento do SASC foram encontradas algumas dificuldades, principalmente ocasionadas por limitações tecnológicas.

Pode-se destacar como a principal dificuldade enfrentada, a inviabilidade de comunicação do novo sistema com o sistema já utilizado na empresa, ou com o seu banco de dados uma vez que o sistema antigo bem como sua base de dados é restrita e pertencente à outra organização da qual faz parte a empresa contratante. Como um dos requisitos exigidos pela Empresa X refere-se à utilização de informações armazenadas ou geradas pelo sistema antigo, a solução adotada foi o esquema de exportação e importação de arquivos, desta forma, o SASC foi preparado para realizar tratamento de dados e importação de relatórios gerados pelo sistema antigo.

6.3 TRABALHOS FUTUROS

Por se tratar da primeira versão do SASC, este trabalho abordou o desenvolvimento dos módulos priorizados pelo gerente geral da empresa contratante. Em entrevista, ele nos respondeu quais eram estas prioridades:

O controle da venda, pagamentos e inadimplência dos 02 convênios de planos de saúde, seguros de várias modalidades como vida, não-nominado, veículos, equipamento e residencial, além de gerar o comissionamento de cada plano vendido. Operações de crédito complexas, que atualmente são controlados por meio de planilhas eletrônicas.

Contudo, pode-se observar em entrevistas aos demais funcionários da empresa, que existem muitos processos que podem ser otimizados com a adoção da tecnologia da informação. Pode-se citar como exemplo: o controle dos repasses rurais aos associados, controle das dívidas e despesas da empresa com outras instituições financeiras, controle dos associados inadimplentes, controle do estoque de materiais e controle cadastro de restrições de crédito e operações ajuizadas.

Outro projeto futuro será levar o SASC para as outras agências da Empresa X, utilizando a arquitetura cliente/servidor já existente entre as mesmas, e viabilizando assim o acesso de todos os funcionários ao sistema, o que irá permitir

que o SASC se torne uma via de comunicação entre as agências e funcionários, trazendo benefícios como, por exemplo, redução de custos.

REFERÊNCIAS BIBLIOGRÁFICAS

JACOBSON, Ivar; BOOCH, Grady; RUMBAUGH, James; **UML Guia do Usuário**. Rio de Janeiro: Campus, 2000.

LEE, Richard C; TEPFENHART, William M; **UML e C++ / Guia Prático de Desenvolvimento Orientado a Objeto**, 2001 – Makron Books.

MARTIN, James; ODELL, James J.; **Análise e Projeto Orientado a Objeto**, 1995 – Makron Books, São Paulo.

NIYAMA, NIYAMA. **Instituições Financeiras** - 2ª Edição. Editora Atlas, 2002.

ORLANDINI, Leandro; **A Importância dos Sistemas de Informação**. [On-Line]. Consultado em 20/08/2007. Disponível em http://www.bonde.com.br/colunistas/colunistasd.php?id_artigo=1646.

RICARTE, Ivan. L. M; **Introdução a Orientação a Objetos**. Unicamp Web Site, 2001. Disponível em http://www.dca.fee.unicamp.br/cursos/POO_CPP/node3.html.

PÁDUA FILHO, Wilson; **Manual do Engenheiro de Software / Módulo Técnico**, 2000.

SEABRA JR, Rodolfo Moacir; **Dissertação: Análise e Projeto orientado a objetos usando UML e o Processo Unificado**. Belém: 2001.

ANEXO A – ENTREVISTAS

1. Existe(m) dificuldades operacionais encontradas hoje no seu setor? Quais?

R: As operações de crédito rural, analisadas e contratadas na cooperativa, são organizadas e controladas de forma não muito automatizada, o que dificulta o trabalho e a segurança nas informações. As planilhas de orçamento e declarações emitidos para assinatura dos associados juntamente com o contrato de crédito são realizadas via planilhas com o auxílio de macro e fórmulas deste aplicativo, o que dificulta o processo, pois demandam tempo e são passíveis de erros, visto que não é um processo automatizado.

As cédulas mães, de repasses de recursos aos os associados, e a menção adicional são confeccionadas via Word e Excel. Nestes documentos existem informações cruciais ao bom andamento do processo como: nomes dos associados, a quantidade de hectares da fazenda, quantidade de rebanho, valor financiado, data de emissão e data de vencimento das operações, informações estas que são encaminhadas a Central Crediminas.

O controle das cédulas mães é feito via planilha. Estas cédulas são cadastradas em uma outra planilha onde se procura acompanhar mensalmente os encargos gerados no período e onde se procura gerar informações necessárias para digitação na contabilidade da cooperativa como despesas mensais da mesma.

2. Em sua opinião, um sistema de informação ajudaria a melhorar os processos da sua empresa?

R: Um sistema de informação poderia facilitar este processo, devido a quantidade de informações que estão sendo manipuladas e devido a importância quanto a segurança da informação repassada a Central Crediminas e aos associados.

Essas informações poderiam ser feitas de forma automatizadas e poderia dinamizar melhor o processo, evitando retrabalho e deixando que os funcionários obtivessem mais tempo disponíveis para execução de outras tarefas. Através de um sistema a empresa poderá maximizar a utilidade do trabalho dos seus colaboradores.

3. Quais as principais vantagens que você apontaria para um processo de informatização destes processos?

R: A contratação das operações de crédito poderia ser de forma informatizada, onde se alimentaria as informações básicas de cada associado no sistema e através deste banco de dados o sistema poderia gerar todos os documentos necessários para a efetivação do processo de contratação. Poderia gerar os documentos necessários para o associado e os documentos necessários para envio destas informações a Central Crediminas.

As cédulas mães poderiam ser cadastradas no sistema e o próprio sistema calcularia mensalmente os encargos financeiros destas operações, e geraria automaticamente a contabilização destas despesas.

Além de gerar todos estes documentos necessário e de facilitar o processo contábil da cooperativa, um sistema poderia dinamizar o processo de atendimento junto ao associado. Poderia ter uma forma de gerenciar melhor estas informações.