

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

ALINE HELOIZE COSTA VERGO POLAN

APLICATIVO DESKTOP PARA CORRETORAS DE SEGUROS

TRABALHO DE CONCLUSÃO DE CURSO

**PATO BRANCO
2015**

ALINE HELOIZE COSTA VERGO POLAN

APLICATIVO DESKTOP PARA CORRETORAS DE SEGUROS

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Diplomação, do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, como requisito parcial para obtenção do título de Tecnóloga.

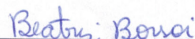
Orientadora: Beatriz Terezinha Borsoi

**PATO BRANCO
2015**

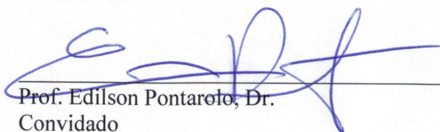
ATA Nº: 266

DEFESA PÚBLICA DO TRABALHO DE DIPLOMAÇÃO DA ALUNA ALINE HELOIZE COSTA VERGO POLAN.

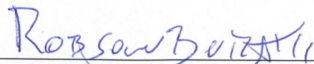
Às 19:00 hrs do dia 24 de junho de 2015, Bloco V da UTFPR, Câmpus Pato Branco, reuniu-se a banca avaliadora composta pelos professores Beatriz Terezinha Borsoi (Orientadora), Edilson Pontarolo (Convidado) e Robson Buratti Rodrigues (Convidado), para avaliar o Trabalho de Diplomação da aluna Aline Heloize Costa Vergo Polan, matrícula 1167960, sob o título **Aplicativo desktop para corretoras de seguros**; como requisito final para a conclusão da disciplina Trabalho de Diplomação do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, COADS. Após a apresentação a candidata foi entrevistada pela banca, examinadora, e a palavra foi aberta ao público. Em seguida, a banca reuniu-se para deliberar considerando o trabalho **APROVADO**. Às 20:10 hrs foi encerrada a sessão.



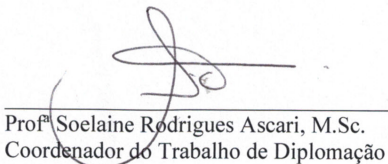
Prof. Beatriz Terezinha Borsoi, Dr.
Orientadora



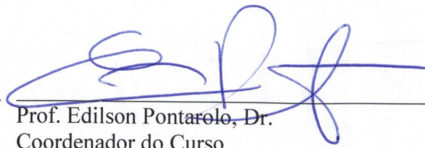
Prof. Edilson Pontarolo, Dr.
Convidado



Prof. Robson Buratti Rodrigues, Esp.
Convidado



Prof. Soelaine Rodrigues Ascari, M.Sc.
Coordenador do Trabalho de Diplomação



Prof. Edilson Pontarolo, Dr.
Coordenador do Curso

RESUMO

POLAN, Aline Heloize Costa Vergo. Aplicativo desktop para corretoras de seguros. 2015. 55f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2015.

O processo realizado por uma corretora para a comercialização de seguros é composto por várias etapas. Algumas dessas etapas são realizadas durante o período de contratação do serviço, como a realização de orçamentos, a elaboração da proposta e a emissão da apólice do seguro pela seguradora. As demais etapas são realizadas durante o período do contrato do seguro, que consiste no atendimento ao segurado e ajuda na utilização do serviço. Durante todas essas etapas são gerados diversos documentos e uma grande quantidade de informações que devem ser armazenadas. Assim, verificou-se uma oportunidade de propor uma maneira de melhorar a forma de armazenamento e a organização dessas informações. Nesse trabalho é apresentado o desenvolvimento de um software para de corretoras de seguros. A modelagem foi elaborada seguindo as orientações do processo unificado, utiliza a Unified Modeling Language e a ferramenta Visual Paradigm for UML para a construção dos modelos e diagramas. A implementação do aplicativo foi realizada utilizando a linguagem Delphi com o banco de dados Firebird.

Palavras-chave: Corretora de seguros. Modelagem UML. Aplicativo para corretoras de seguros.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 – Dados do mercado brasileiro de seguros | 11 |
| Figura 2 – Dados de seguros de pessoas e de danos | 11 |
| Figura 3 - Exemplo de diagrama de caso de uso..... | 18 |
| Figura 4 - Exemplificação de atributos e associações..... | 19 |
| Figura 5 - Tabela de multiplicidade..... | 19 |
| Figura 6 - Diagrama de classes..... | 20 |
| Figura 7 - Diagrama de atividades..... | 21 |
| Figura 8 - Diagrama de sequência | 22 |
| Figura 9 - Diagrama de comunicação | 22 |
| Figura 10 – Etapas do fluxo do Processo Unificado..... | 23 |
| Figura 11 - Esboço do cronograma no Processo Unificado..... | 24 |
| Figura 12 - Diagrama de atividades..... | 33 |
| Figura 13 - Diagrama de caso de uso..... | 34 |
| Figura 14 - Diagrama de classes..... | 37 |
| Figura 15 - Diagrama de classes (registrar nova proposta)..... | 38 |
| Figura 16 - Diagrama de sequência do caso de uso registrar nova proposta | 39 |
| Figura 17 - Diagrama de sequência do caso de uso registrar aviso de sinistro | 39 |
| Figura 18 - Diagrama de sequência do caso de uso acompanhar renovações | 40 |
| Figura 19 - Diagrama de sequência do caso de uso registrar vistoria prévia | 40 |
| Figura 20 - Diagrama de comunicação do caso de uso registrar nova proposta..... | 41 |
| Figura 21 - Diagrama de comunicação do caso de uso registrar aviso de sinistro | 41 |
| Figura 22 - Diagrama de comunicação do caso de uso acompanhar renovações | 41 |
| Figura 23 - Diagrama de comunicação do caso de uso registrar vistoria prévia..... | 42 |
| Figura 24 – Tela de login..... | 42 |
| Figura 25 – Tela inicial do sistema | 43 |
| Figura 26 – Aba pesquisa do formulário de cadastro de clientes..... | 44 |
| Figura 27 – Aba manutenção do formulário de cadastro de clientes | 44 |
| Figura 28 – Exemplo de validação de campos obrigatórios..... | 45 |
| Figura 29 – Exemplo de validação de máscaras e labels | 45 |
| Figura 30 – Formulário para inclusão de clientes..... | 46 |
| Figura 31 – Formulário para exclusão de clientes..... | 46 |
| Figura 32 – Formulário de acompanhamento das propostas..... | 47 |
| Figura 33 – Componentes visuais do módulo de dados DM | 51 |

LISTA DE QUADROS

| | |
|---|----|
| Quadro 1 - Ferramentas e tecnologias de modelagem e implementação | 25 |
| Quadro 2 - Casos de uso | 31 |
| Quadro 3 - Conceitos | 33 |
| Quadro 4 - Consultas e relatórios..... | 33 |
| Quadro 5 - Expansão do caso de uso: registrar nova proposta | 35 |
| Quadro 6 - Expansão do caso de uso: registrar aviso de sinistro..... | 35 |
| Quadro 7 - Expansão do caso de uso: acompanhar renovações | 36 |
| Quadro 8 - Expansão do caso de uso: registrar vistoria prévia..... | 36 |

LISTAGENS DE CÓDIGO

| | |
|--|----|
| Listagem 1 – Unit_Menu | 49 |
| Listagem 2 – Unit_DM | 50 |
| Listagem 3 – Trecho do código da Unit_Produtores | 52 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|-------------|---|
| <i>BPMN</i> | <i>Business Process Model and Notation</i> |
| <i>DFD</i> | <i>Data Flow Diagram</i> |
| DPVAT | Danos Pessoais Causados por Veículos Automotores de Via Terrestre |
| <i>ERD</i> | <i>Entity-Relationship Diagram</i> |
| HTML | HyperText Markup Language |
| IoT | Internet of Things |
| OO | Orientação a Objetos |
| POO | Programação Orientada a Objetos |
| PU | Processo Unificado |
| RTF | Rich Text <i>Format</i> |
| SGBD | Sistema Gerenciador de Banco de Dados |
| SQL | <i>Structured Query Language</i> |
| SUSEP | Superintendência de Seguros Privados |
| UML | <i>Unified Modeling Language</i> |
| XML | <i>Extensible Markup Language</i> |

SUMÁRIO

| | |
|---|-----------|
| 1 INTRODUÇÃO | 9 |
| 1.1 CONSIDERAÇÕES INICIAIS | 9 |
| 1.2 OBJETIVOS | 10 |
| 1.2.1 Objetivo Geral | 10 |
| 1.2.2 Objetivos Específicos | 10 |
| 1.3 JUSTIFICATIVA..... | 10 |
| 1.4 ESTRUTURA DO TRABALHO | 12 |
| 2 FUNDAMENTAÇÃO TEÓRICA | 13 |
| 2.1 DEFINIÇÕES NO MERCADO DE SEGUROS | 13 |
| 2.2 MODELAGEM DE SOFTWARE | 15 |
| 2.2.1 Modelagem Utilizando Orientação a Objetos | 15 |
| 2.2.2 Linguagem de Modelagem Unificada (UML) | 17 |
| 2.2.2.1 Diagrama de Caso de Uso | 17 |
| 2.2.2.2 Diagrama de Classes | 18 |
| 2.2.2.3 Diagrama de Atividades..... | 20 |
| 2.2.2.4 Diagrama de Interação | 21 |
| 2.3 PROCESSO UNIFICADO | 23 |
| 3 MATERIAIS E MÉTODOS | 25 |
| 3.1 MATERIAIS | 25 |
| 3.2 MÉTODO..... | 26 |
| 4 RESULTADO | 28 |
| 4.1 VISÃO GERAL DO SISTEMA | 28 |
| 4.2 MODELAGEM DO SISTEMA | 29 |
| 4.3 APRESENTAÇÃO DO SISTEMA | 42 |
| 4.4 IMPLEMENTAÇÃO DO SISTEMA..... | 48 |
| 5 CONCLUSÃO | 53 |
| REFERÊNCIAS | 54 |

1 INTRODUÇÃO

Este capítulo está dividido em considerações iniciais, objetivo geral, objetivos específicos, justificativa e por fim a estrutura geral do trabalho.

1.1 CONSIDERAÇÕES INICIAIS

O foco deste trabalho está voltado para a área de comercialização de seguros. A atividade de comercialização de seguros não é algo relativamente novo, iniciou em 1990 (TSS, 2015). Porém, o que se pode notar, com base na experiência da autora deste trabalho que atuou em empresa de seguros, é que existe pouco desenvolvimento tecnológico para as empresas que efetivamente realizam a venda de seguros, conhecidas como corretoras de seguros. Esse desenvolvimento é colocado em termos de sistemas de informação que contribuam para a realização das atividades rotineiras dessas empresas e como ferramenta de suporte à gestão e ao processo de tomada de decisão.

Uma corretora de seguros desempenha a função de intermediar a comunicação entre cliente e seguradora buscando os melhores serviços e condições aos clientes e oferecendo auxílio na contratação e utilização dos serviços. O trabalho da corretora inclui a realização de orçamentos, o fechamento de contratos, o atendimento aos imprevistos e sinistros e o pagamento de indenizações.

Considerando a quantidade e diversidade de atividades realizadas, essas empresas precisam de ferramentas que facilitem o seu trabalho. Partindo desse princípio, como resultado deste trabalho será desenvolvido um software para o controle e o fluxo de dados gerados pelas atividades e processos diários realizados por uma corretora de seguros. Visando, assim, agilizar o trabalho e organizar e centralizar as informações.

Neste trabalho é descrita a implementação de um aplicativo para ambiente *desktop* para gerenciamento de seguradoras.

1.2 OBJETIVOS

A seguir são apresentados o objetivo geral e os objetivos específicos da realização deste trabalho.

1.2.1 Objetivo Geral

Implementar um aplicativo para corretoras de seguros.

1.2.2 Objetivos Específicos

- Facilitar a busca de informações sobre segurados, produtores (vendedores de seguros), propostas, endossos, sinistros e apólices de seguros;
- Acompanhar a emissão de propostas de seguros, endossos e o pagamento de indenizações para segurados;
- Auxiliar no controle de renovações e cancelamentos de apólices de seguros.

1.3 JUSTIFICATIVA

O mercado de seguros em 2013 foi de 3% do Produto Interno Bruto (PIB) do mercado brasileiro, conforme indicam dados da Figura 1.

| MERCADO BRASILEIRO DE SEGUROS | | | |
|-------------------------------|-----------------|-------------|---------------|
| ANO | PRÊMIOS DIRETOS | PROVISÕES | PART. PIB (%) |
| 2001 | 24.211.622 | 9.778.502 | 1,86% |
| 2002 | 23.910.777 | 13.443.557 | 1,62% |
| 2003 | 30.717.421 | 22.035.380 | 1,81% |
| 2004 | 37.546.345 | 33.537.168 | 1,93% |
| 2005 | 42.561.865 | 46.856.177 | 1,98% |
| 2006 | 49.587.528 | 62.575.290 | 2,09% |
| 2007 | 58.443.093 | 81.812.550 | 2,20% |
| 2008 | 67.816.374 | 100.776.225 | 2,24% |
| 2009 | 76.611.206 | 137.425.491 | 2,36% |
| 2010 | 90.088.893 | 172.190.110 | 2,39% |
| 2011 | 105.031.660 | 215.652.788 | 2,54% |
| 2012 | 129.340.524 | 275.734.453 | 2,94% |
| 2013 | 145.348.303 | 318.617.798 | 3,00% |

(Valores em R\$ mil)

Figura 1 – Dados do mercado brasileiro de seguros
 Fonte: SUPERINTENDÊNCIA... (2014, p. 5).

Dados apresentado pela Superintendência Seguros Privados (SUSEP) (SUPERINTENDÊNCIA..., 2014, p. 5) indicam que entre 2001 e 2012 houve crescimento maior dos seguros pessoais, em comparação com seguros de danos. Contudo, essa tendência foi invertida em 2013. A Figura 2 apresenta esses dados.

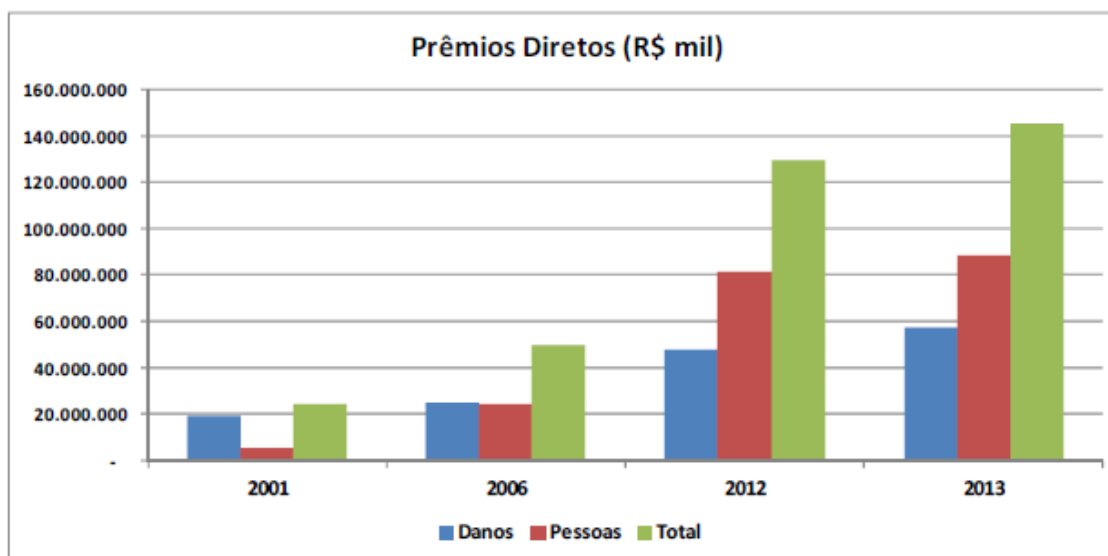


Figura 2 – Dados de seguros de pessoas e de danos
 Fonte: SUPERINTENDÊNCIA... (2014, p. 7).

Diante do crescimento e expressividade do setor de seguros como mostram os dados da Tabela 1 e Gráfico 1, surge a necessidade de aplicativos computacionais que atendam essa demanda. Assim, tendo em vista a necessidade de um sistema especializado no ramo de seguros, propõem-se o desenvolvimento de um software que auxilie os profissionais das seguradoras no controle do fluxo das informações. E, também, na automatização de processos rotineiros, na otimização do tempo gasto com o arquivamento de papéis e documentos, nos controles financeiros e gerenciamento de dados cadastrais. Dessa forma visou-se transformar os processos rotineiros em atividades automatizadas e integradas a um sistema que possibilite o gerenciamento dos processos de forma rápida.

1.4 ESTRUTURA DO TRABALHO

O trabalho está dividido em capítulos, de forma que no primeiro capítulo é apresentada a ideia geral, conceitos iniciais do trabalho, os objetivos e a justificativa. O Capítulo 2 apresenta a fundamentação teórica do estudo e irá explicar a estrutura e definições da tecnologia *Unified Modeling Language* (UML) e do Processo Unificado que foi utilizada durante o processo de desenvolvimento. O Capítulo 3 exibirá o estudo de caso do trabalho, os materiais e métodos utilizados para a realização da modelagem do software e os resultados obtidos através da mesma, por fim, será feita a conclusão do trabalho e a avaliação dos resultados no Capítulo 4.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo contém o referencial teórico do trabalho em seus principais aspectos e conceitos relacionados ao software desenvolvido. Inicialmente é apresentado o contexto geral sobre o mercado de seguros, algumas regras de negócio, produtos e serviços ofertados, organização de áreas e categorias desse mercado e sobre os papéis desempenhados pelas corretoras de seguros e seguradoras. A seção seguinte é sobre modelagem de software, partindo de uma visão geral e conceitual do assunto e em seguida é tratado especificamente sobre o padrão de modelagem UML e o Processo Unificado.

2.1 DEFINIÇÕES NO MERCADO DE SEGUROS

Denomina-se contrato de seguro aquele que estabelece para uma das partes, mediante recebimento de um prêmio da outra parte, a obrigação de indenizar a esta, uma determinada importância, no caso da ocorrência de um evento futuro e incerto, mas devidamente previsto nas cláusulas definidas no contrato, o documento que formaliza tal contrato chama-se apólice (PRECATO, 2015).

Entende-se por seguradora a parte que assume o risco do contrato e a indenização em caso de ocorrência de sinistro amparado pelo contrato de seguro (PRECATO, 2015). A parte contratante é denominada segurado e é por essa parte que a seguradora assume os riscos de uma possível indenização.

Sinistro pode-se definir como um evento ou acontecimento previsto em contrato, com o bem segurado (PRECATO, 2015).

Além desses conceitos são importantes os termos (TSS, 2015, p. 1):

a) Acidente ou Sinistro – evento imprevisto ou fortuito do qual resulta dano à pessoa ou bem segurado.

b) Apólice – instrumento do contrato de seguro por meio do qual o segurado repassa à seguradora a responsabilidade sobre os riscos que podem advir ao objeto de seguro.

c) Carência – período durante o qual a seguradora está isenta de pagamento dos riscos segurados.

d) Contrato de seguro ou Apólice – documento, geralmente expresso por uma apólice e composto de cláusulas, que define as condições e regras para ressarcir o segurado em caso de ocorrência dos eventos objetos do seguro e determina as obrigações do segurado.

e) Dano – prejuízo sofrido pelo segurado que é indenizável.

f) Evento – ocorrência ou acontecimento que pode ser garantida por uma apólice de seguro.

g) Inspeção ou Vistoria – exame do bem objeto do seguro. O resultado desse exame é registrado na forma de laudo.

h) Objeto de seguro – designação de qualquer interesse segurado, sejam coisas, pessoas, bens, responsabilidades, obrigações, direitos ou garantias.

i) Vigência - período de validade do contrato do seguro. Normalmente o período de validade para seguros no geral é de um ano.

Existe uma variedade de ramos de seguros que podem ser organizados em grupos distintos, segundo o site oficial da SUSEP os principais grupos de seguros são (SUPERINTENDÊNCIA, 2015):

- Patrimonial: seguros contra incêndios, roubos, queda de raio, vendaval, entre outros.
- Automóvel: seguros contra roubos, acidentes, perda total, responsabilidade civil contra terceiros e Danos Pessoais Causados por Veículos Automotores de Via Terrestre (DPVAT)..
- Pessoas: seguros de vida e acidentes pessoais, morte e invalidez.
- Rural: seguros agrícola, pecuário e penhor rural.
- Transporte: seguros de cargas, transporte nacional e internacional.
- Saúde: seguros de saúde, planos e contratos de saúde.

Para a contratação de um seguro é necessário que o negócio seja intermediado por um corretor de seguros, devidamente habilitado. O corretor é o responsável legal pelo segurado e representa o segurado diante da seguradora, defendendo seus interesses.

2.2 MODELAGEM DE SOFTWARE

Devido ao grande aumento da demanda e complexidade de softwares no geral, é necessário buscar alternativas para reduzir o tempo, custo do desenvolvimento e a manutenção do mesmo. A modelagem surge como uma opção para ajudar no processo de construção do software e pode ser entendida como uma representação simplificada de algo real.

A modelagem de software normalmente implica a construção de modelos gráficos que simbolizam os artefatos dos componentes de software utilizados e os seus inter-relacionamentos. Um modelo representa um conjunto de abstrações que fornece suporte a um desenvolvedor em um aspecto de desenvolvimento bem definido.

De acordo com Booch, Rumbaugh e Jacobson (2000), alguns princípios devem ser seguidos na modelagem:

- A escolha de qual modelo construir tem uma profunda influência em como um problema é atacado e como uma solução é delineada.
- Todo modelo pode ser expresso em diferentes níveis de precisão.
- Os melhores modelos estão conectados à realidade.
- Nenhum modelo único é suficiente. Todo sistema não trivial é melhor abordado por meio de um conjunto pequeno de modelos proximalmente independentes.

Uma forma comum de modelagem de programas orientados a objeto é a utilização da linguagem gráfica UML.

2.2.1 Modelagem Utilizando Orientação a Objetos

A Programação Orientada a Objetos (POO) tem sido a tecnologia de programação dominante e seus benefícios são amplamente reconhecidos (CHAVEZ, 2004). Com o aumento da procura por software cada vez mais elaborado e com melhor qualidade, o desenvolvimento passou a ser realizado seguindo técnicas como a Orientação de Objetos (OO). A OO visa prover maior reutilização de código, facilitar a manutenção e melhoria do sistema. Isso ocorre pelo uso de código mais amplamente testado que é obtido pelo reuso. O reuso também visa aumentar a

produtividade da equipe de trabalho, pela redução da quantidade de código produzido pelo uso de código já existente.

A seguir são apresentados alguns conceitos básicos sobre OO, são eles: objetos, atributos, estados, polimorfismo, herança, encapsulamento e interações.

Objetos

Define-se um objeto como um conceito, uma abstração, algo com limites nítidos e significado em relação ao que está sendo estudado ou que se quer representar (RUMBAUGH *et al.*, 1994). Cada objeto possui características e funcionalidades específicas representadas por dados e operações são realizadas com esses dados. Os objetos interagem com outros objetos por meio de troca de mensagens, compondo, assim, o sistema.

Estado

O estado de um objeto é um atributo, no entanto este atributo (estado) é tomado em relação aos demais objetos, definindo a condição em que objeto se encontra (O'DOCHERTY, 2005).

Encapsulamento

O encapsulamento refere-se a ocultar os atributos de um objeto. Os atributos são acessados por meio de operações. O encapsulamento evita que usuários do objeto acessem diretamente os atributos, evitando dependências em relação aos atributos (O'DOCHERTY, 2005).

Herança e polimorfismo

A herança é uma forma de reutilização de software em que novas classes são criadas a partir das classes existentes, absorvendo seus atributos e comportamentos e adicionando novas funcionalidades que as novas classes exigem. O polimorfismo permite que se escrevam programas de uma forma geral para tratar uma ampla variedade de classes relacionadas que já existem e as ainda a serem especificadas tornando fácil adicionar novos recursos ao software (DEITEL, 2003).

Interações

Segundo O'Docherty (2005), as interações entre os objetos podem ocorrer de duas formas: Associação e Agregação. Associação é quando um objeto é associado a outro a fim de compartilhar suas propriedades. Já Agregação é quando um ou mais objetos são juntados para formar outro objeto.

2.2.2 Linguagem de Modelagem Unificada (UML)

A UML é uma família de notações gráficas, apoiada por um metamodelo único, que ajuda na descrição e no projeto de sistemas de software, particularmente daqueles construídos utilizando conceitos de orientação a objetos (FOWLER, 2005).

Um bom modelo inclui os elementos que são importantes para representar uma visão do sistema de acordo determinado nível de abstração e com interesses definidos para a representação do modelo. Cada sistema pode ser descrito a partir de aspectos diferentes, utilizando modelos diferentes. Um modelo pode ser estrutural, enfatizando a organização do sistema, ou pode ser comportamental, enfatizando a dinâmica do sistema.

A UML inclui conceitos semânticos, notação e diretrizes. Destina-se a ser apoiada por ferramentas interativas de modelagem visual que têm geradores de código e editores de relatório. A especificação UML não define um processo padrão, mas destina-se a ser útil para um processo iterativo de desenvolvimento. (RUMBAUGH *et al.*, 1994). A seguir a descrição dos principais diagramas que compõem a UML.

2.2.2.1 Diagrama de Caso de Uso

Segundo Jacobson (2006), um caso de uso é um documento que descreve a sequência de eventos de um ator ao usar um sistema para realizar de forma completa um processo.

Um caso de uso é uma técnica de modelagem usada para descrever o que um novo sistema deve fazer. Um caso de uso se apresenta como uma lista completa das interações entre um usuário e o sistema para cumprir uma tarefa. Lista completa significa que o caso de uso descreve as interações desde o início da tarefa, até o fim. Podemos dizer que os componentes de um modelo de caso de uso são (BOOCH; RUMBAUGH; JACOBSON, 2000):

- **Ator** - é um papel que tipicamente estimula/solicita ações/eventos do sistema e recebe reações. Cada ator pode participar de vários casos de uso.
- **Caso de uso** - sequência de eventos feitos por um ator no uso do sistema.
- **Relacionamento de dependência, generalização e associação** – os relacionamentos entre os casos de uso.

Um diagrama de casos de uso organiza e modela o comportamento do sistema e mostra um conjunto de atores e de casos de uso. Na Figura 1 é exibido um ator principal e são vinculados a ele os principais casos de uso que serão executados. A Figura 3 representa um diagrama de casos de uso parcial de um processo de vendas pela Internet.

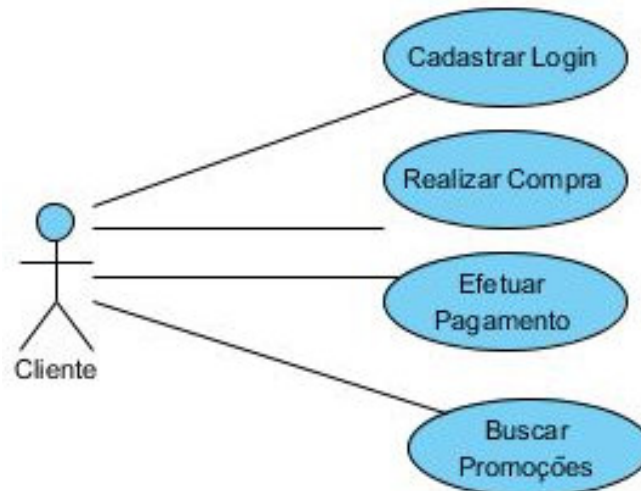


Figura 3 - Exemplo de diagrama de caso de uso
Fonte: Autoria Própria.

2.2.2.2 Diagrama de Classes

O diagrama de classes representa a estrutura do sistema, permite a visualização das classes, seus atributos e suas relações. O modelo de classes resulta de um processo de abstração no qual são identificados os objetos relevantes do sistema.

De acordo com Silva (2010), o comportamento de um objeto define o modo como ele age e reage a estímulos externos e a identidade de um objeto é um atributo que o distingue de todos os demais. Uma classe pode herdar dados e comportamento de outra classe, ou seja, herda todo o comportamento e funcionalidades da classe pai. Um diagrama de classes identifica a associação e cardinalidades entre as classes. Simplificando um diagrama exibe um conjunto de classes, interfaces e seus relacionamentos.

A Figura 4 apresenta duas classes distintas, cliente e pedido, com seus respectivos atributos. As classes se relacionam entre si por meio de associação.

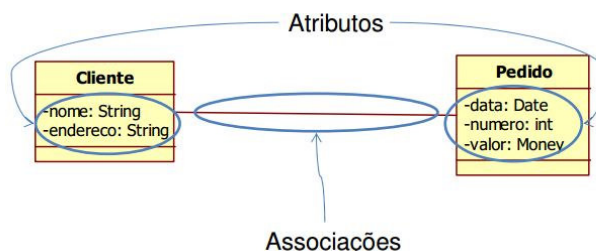


Figura 4 - Exemplificação de atributos e associações
 Fonte: Silva (2010, p. 5).

Na Figura 4 os itens destacados são:

- **Atributos:** Atributos são propriedades que descrevem as classes, atributos equivalem a relacionamentos de composição.
- **Associações:** São utilizadas para relacionar duas classes, só as classes que estão relacionadas são as classes cujos objetos podem se comunicar.

Depois de separadas as classes, juntamente com seus atributos e realizadas as associações entre as mesmas, é necessário definir a cardinalidade dessas associações.

A Figura 5 apresenta as notações utilizadas para as diferentes formas de multiplicidade. Cada tipo de multiplicidade possui uma simbologia correspondente. Essa simbologia auxilia no entendimento das interações entre as classes.

| Multiplicidade | Simbologia |
|---|------------|
| Zero ou um (opcional) | 0..1 |
| Um para um | 1..1 |
| Um para muitos | 1..n |
| Zero ou muitos | 0..n |
| Um a vinte (valor entre o intervalo estabelecido) | 1..20 |

Figura 5 - Tabela de multiplicidade
 Fonte: Lidel (2015, p. 4).

Na Figura 6 é apresentado um diagrama de classes, com a representação dos componentes e de sua estrutura: classes nomeadas, com atributos e operações, suas respectivas associações e cardinalidades.

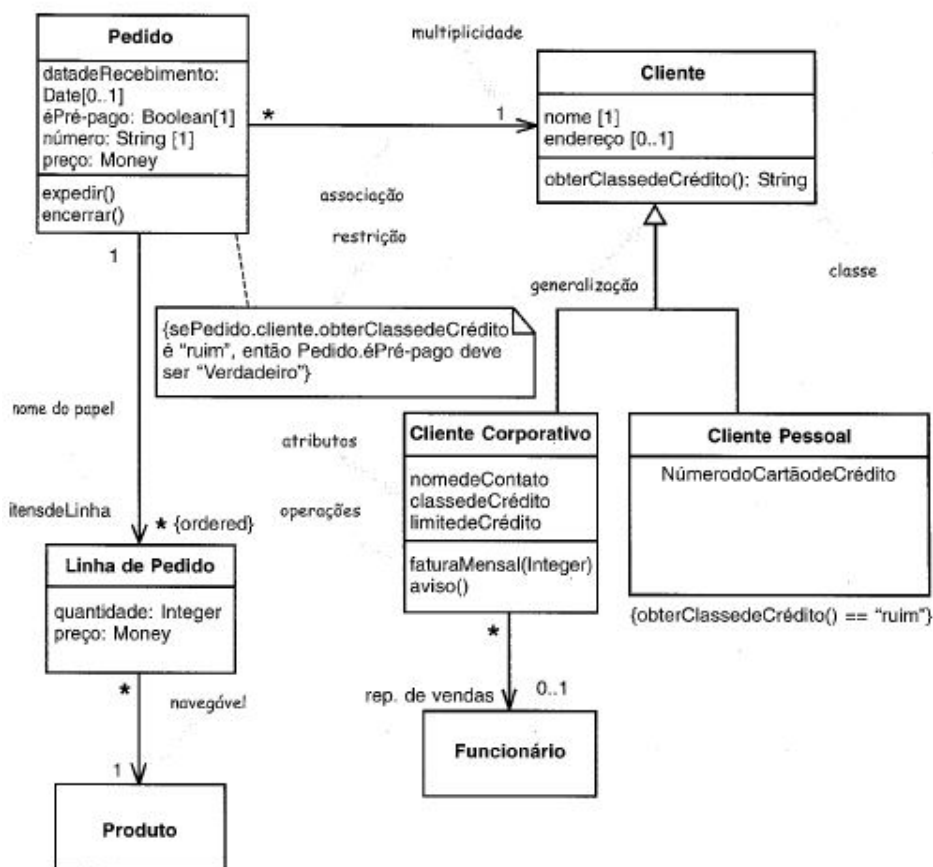


Figura 6 - Diagrama de classes
Fonte: Fowler (2005).

2.2.2.3 Diagrama de Atividades

Um diagrama de atividades tem por objetivo mostrar o fluxo que uma determinada tarefa deve seguir até sua conclusão. Para modelagem de aspectos dinâmicos de sistemas, um diagrama de atividade é um gráfico de fluxo de controle de uma atividade para outra. No exemplo de um diagrama de atividades apresentado na Figura 7, o início do diagrama é dado a partir de um nó inicial. Desse nó inicial os processos de atividades são apresentados de maneira sequencial, levando em conta as duas condições possíveis de resultados em alguns processos

que podem ser executados em paralelo, o diagrama é finalizado com um nó final, representando o fim do processo.

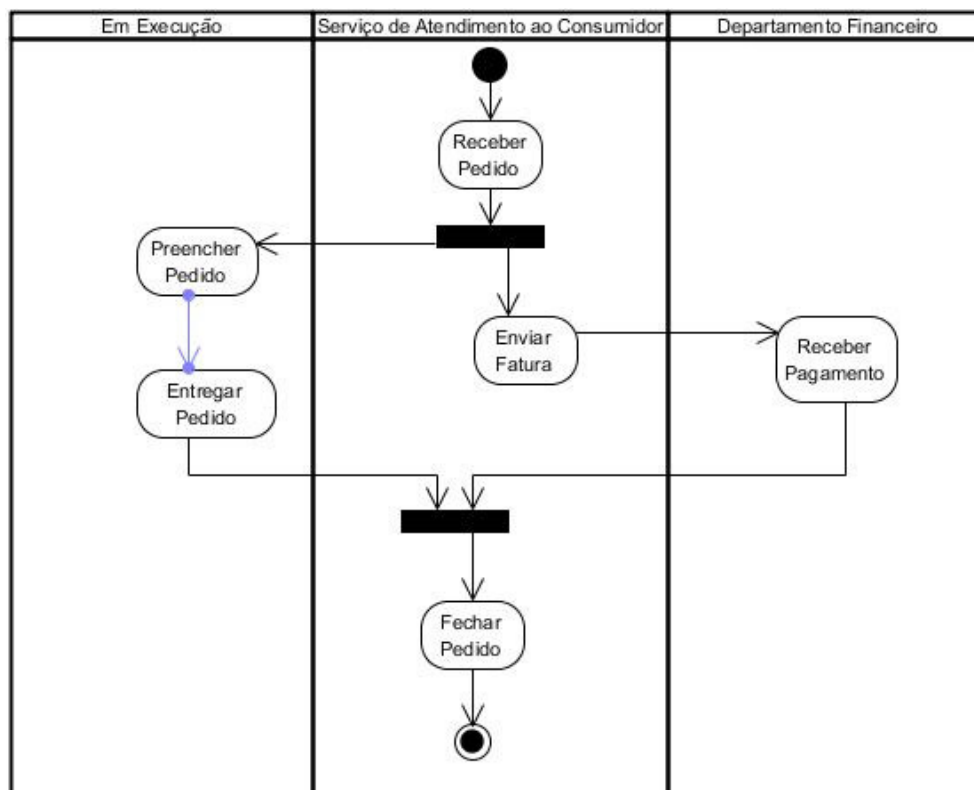


Figura 7 - Diagrama de atividades
Fonte: Fowler (2005).

2.2.2.4 Diagrama de Interação

O objetivo dos diagramas de interação é mostrar como um grupo de objetos colabora entre si em uma determinada operação. Divide-se em diagramas de sequência e diagramas de colaboração.

- **Diagrama de sequência:** representa como as mensagens entre os objetos são trocadas durante uma operação. Este tipo de diagrama também permite mostrar mensagens concorrentes, ou seja, mensagens que são processadas em paralelo sem um tempo limite definido. Na Figura 8 é apresentado um exemplo de um diagrama de sequência referente a um processo simples de geração de extrato bancário, os fluxos sequenciais das ações entre os atores e a troca de informações são bem exemplificados nesse tipo de diagrama.

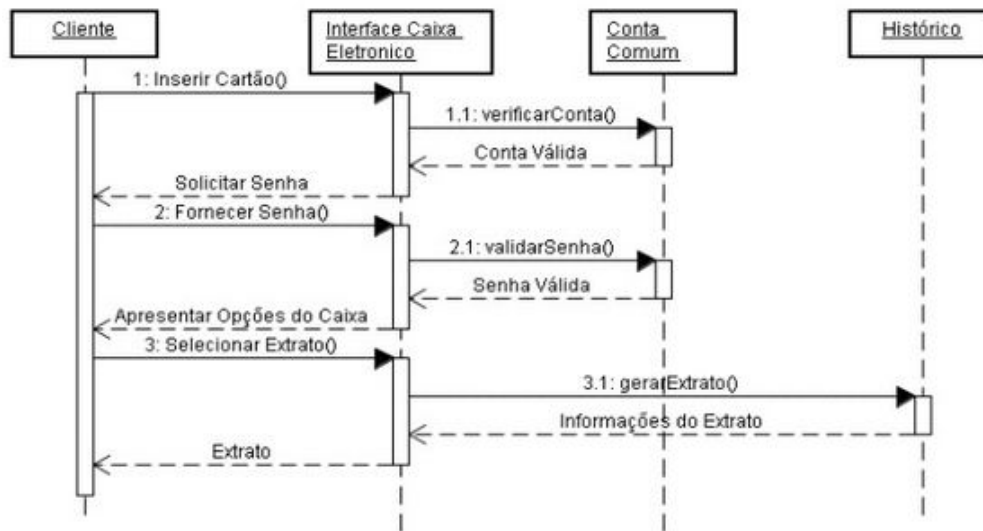


Figura 8 - Diagrama de sequência
Fonte: Senac (2015, p. 12).

- **Diagrama de comunicação:** utilizado quando se deseja mostrar maiores detalhes da colaboração entre os objetos, enfatizando a organização estrutural entre eles. A Figura 9 apresenta um exemplo de um diagrama de comunicação referente a uma tela de login. Nesse diagrama são mostradas as ligações entre os objetos, atores, as classes controladora e fronteira e as mensagens trocadas entre os objetos.

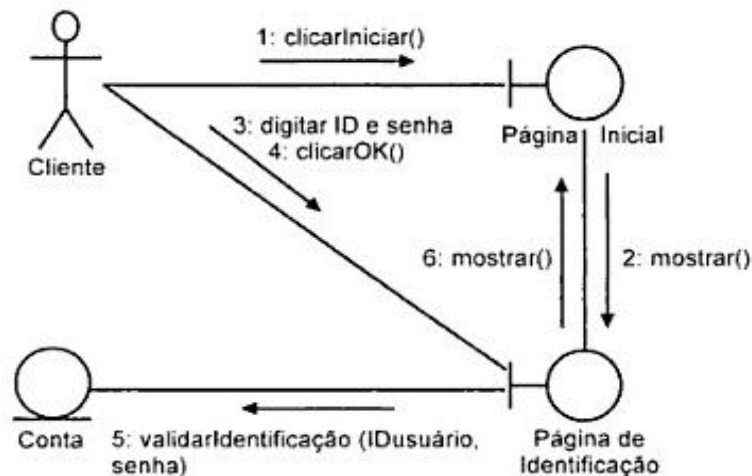


Figura 9 - Diagrama de comunicação
Fonte: Scott (2002).

2.3 PROCESSO UNIFICADO

O Processo Unificado (PU) é caracterizado por ser iterativo e incremental. O PU utiliza ciclos de vida incrementais, baseados em refinamentos e incrementos sucessivos dos requisitos a fim de melhor se adequar aos requisitos de negócio. Cada iteração inclui suas próprias atividades de análise de requisitos, projeto, implementação e testes (LARMAN, 2004).

Conforme pode ser observado na Figura 10, a divisão das tarefas é incremental ao longo das fases do projeto, formando um fluxo contínuo de refinamento e adaptação dos requisitos.

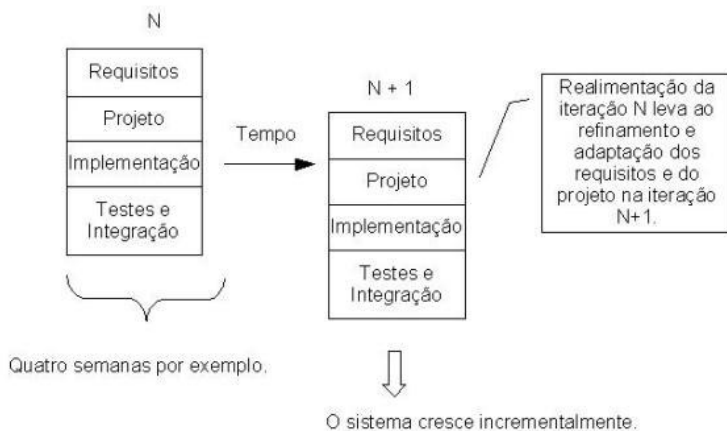


Figura 10 – Etapas do fluxo do Processo Unificado
Fonte: Larman (2004).

Segundo o autor Larman (2004) é altamente recomendado pelo PU que as iterações tenham tempo fixo pré-determinado e que se cumpra o prazo de cada iteração. Iterações pequenas entre duas a seis semanas são o ideal, pois são mais gerenciáveis e permitem rápida realimentação e adaptações.

O Processo Unificado organiza suas iterações em quatro fases principais (LARMAN, 2004):

- **Concepção:** o objetivo desta fase é levantar o escopo do projeto fornecendo uma visão inicial do problema, estimar de forma vaga esforço e prazos e determinar se o projeto é viável e merece uma análise mais profunda.
- **Elaboração:** na fase de elaboração todos os requisitos (ou a grande maioria) são levantados em detalhes. Em uma primeira iteração um ou dois requisitos, os de maior risco e valor arquitetural, são especificados em detalhes. Esses

requisitos são implementados e servem como base de avaliação junto ao usuário e desenvolvedores para o planejamento da próxima iteração.

- **Construção:** implementação iterativa dos elementos restantes de menor risco e mais fáceis e preparação para a implantação.
- **Transição:** testes finais e implantação.

A Figura 11 exibe um esboço das fases do processo unificado e da sequência das etapas a serem executadas durante o processo, a fase de elaboração é composta por várias iterações como pode ser observado nessa Figura.

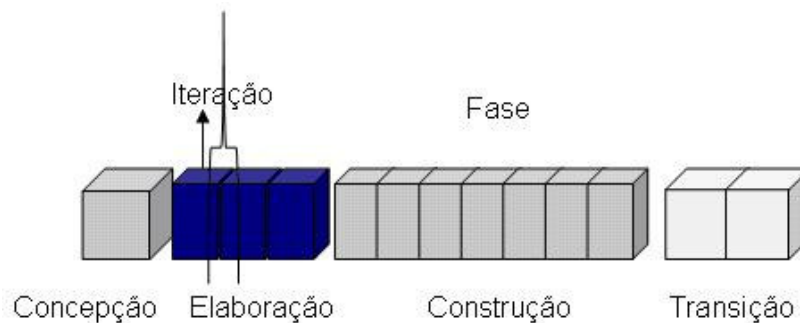


Figura 11 - Esboço do cronograma no Processo Unificado
Fonte: Larman (2004).

O Processo Unificado foi criado para ser um processo ágil de desenvolvimento e prega uma abordagem realística para a condução de um projeto. No Processo Unificado as atividades são repetidas em ciclos organizados. Há um plano de alto nível (chamado Plano de Fases) que estima a data de término do projeto e outros marcos de referência principais, mas ele não detalha os passos de granularidade fina para se atingir tais marcos. Um plano detalhado (Plano de Iterações) somente planeja a iteração a ser feita em seguida. O planejamento detalhado é feito de forma adaptativa, de iteração para iteração.

3 MATERIAIS E MÉTODOS

Este capítulo apresenta informações sobre as ferramentas e metodologias utilizadas na modelagem do software proposto.

3.1 MATERIAIS

O Quadro 1 apresenta as ferramentas e tecnologias utilizadas para a modelagem e a implementação do sistema.

| Ferramenta / Tecnologia | Versão | Referência | Finalidade |
|-------------------------|------------|---|---|
| Visual Paradigm for UML | 10.2 | http://www.visual-paradigm.com/ | Documentação da modelagem baseada na UML. |
| Case Studio 2 | 2.25 | http://www.casestudio.com | Modelagem do diagrama de entidades e relacionamentos do banco de dados. |
| Delphi | 7 | http://www.delphi.com/ | Ambiente de desenvolvimento e linguagem de programação. |
| Firebird | 2.5 | http://www.firebird.sql.org | Banco de dados. |
| IbExpert | 2012-02-21 | http://www.ibexpert.com | Administrador do banco de dados. |
| Rave Report | 11 | http://www.nevrona.com/ | Geração dos relatórios. |

Quadro 1 - Ferramentas e tecnologias de modelagem e implementação

Visual Paradigm for UML é uma ferramenta de *design* que suporta várias opções de modelagem como UML, *Business Process Model and Notation* (BPMN), *Entity-Relationship Diagram* (ERD) e *Data Flow Diagram* (DFD) e diagramas de requisitos SysML (VISUAL PARADIGM, 2015). Essa ferramenta oferece suporte a transformações específicas para códigos fonte de algumas linguagens de programação como C++ e Java.

CASE Studio 2 (TOAD, 2015) é uma ferramenta de modelagem de bases de dados que permite criar visualmente e manter ERDs, DFDs e gerar automaticamente *scripts Structured Query Language* (SQL) para bancos de dados distintos. Além disso, a ferramenta permite engenharia reversa, geração de documentos *HyperText Markup Language* (HTML) e *Rich Text Format* (RTF), Diagramas de Fluxos de Dados (DFD) e exportar para o formato *Extensible Markup Language* (XML), entre outros.

Embarcadero Delphi, ou simplesmente Delphi como é comumente conhecida, é uma linguagem de programação. Essa linguagem está na sua versão XE8 que é descrita como uma solução completa de desenvolvimento de software para um design rápido, codificação e extensão de aplicações para a plataforma Windows, Mac, iOS, Android e Internet of Things (IoT) (EMBARCADERO, 2015).

O Firebird é um banco de dados derivado do código do Borland InterBase 6.0. Esse SGBD é de código aberto e não possui licença dupla, portanto pode ser utilizado em qualquer tipo de aplicação, comercial ou não (CANTÚ, 2010).

IBExpert é Sistema Gerenciador de Banco de Dados (SGBD) para o banco de dados Firebird e todas as versões do InterBase a partir do 4.x (VOLACO, 2015)

Rave Report é desenvolvido e mantido pela empresa Nevrona Designs. A partir da Delphi 7 o Rave Reports tornou-se a ferramenta padrão para geração de relatórios nessa linguagem. O Rave possui um ambiente totalmente visual para criação de relatórios que é semelhante à IDE do Delphi, possuindo paleta de componentes e janelas que se assemelham ao Object Inspector e ao Object TreeView (RAVE, 2015).

3.2 MÉTODO

Ao longo do trabalho para a obtenção de todas as informações necessárias e a captação dos requisitos e funcionalidades do software a ser modelado, foi utilizada a experiência de trabalho da autora, pois a mesma já trabalhou em uma corretora de seguros. Tendo, portanto, conhecimento sobre os processos internos, regras de negócio referentes à comercialização de seguros e a rotina desse tipo de empresa.

Foram consultados documentos emitidos pelas seguradoras que foram gerados dentro do processo de venda e utilização do seguro tais como: orçamentos, propostas, apólices, endossos, avisos de sinistro, de forma a analisar e avaliar a melhor maneira de armazenamento desses documentos dentro do software e as reais necessidades dos usuários. Esses documentos ajudaram na compreensão dos dados e processos que os usuários precisam que o sistema gerencie e controle.

Primeiramente foi realizada uma análise do software a ser desenvolvido e do principal propósito de sua elaboração, essa análise deu origem ao documento

conhecido como visão geral do sistema, este documento explica e apresenta as principais características do software a ser modelado.

Após essa etapa foram realizadas análises e pesquisas mais específicas para identificação dos requisitos e funcionalidades principais, dando origem ao documento de requisitos. Depois de definidos os requisitos, os mesmos foram organizados e selecionados a partir destes os casos de uso do sistema, o documento gerado nesta fase é a tabela de casos de uso. Os "conceitos" do software foram identificados e definidos com base no documento de requisitos, dando origem à tabela de conceitos. Os requisitos que se encaixam como consultas e relatórios foram organizados em um novo documento, o de relatórios e consultas foram definidas as referências cruzadas.

Depois de criados todos os documentos acima foram elaborados os diagramas de atividades, diagrama de caso de uso, expansão dos casos de uso, diagrama de classes, diagrama de sequência e o diagrama de comunicação.

A implementação foi realizada tendo como base a modelagem gerada utilizando a linguagem de programação Delphi.

4 RESULTADO

A seguir será apresentada a visão geral do sistema, as principais regras de negócio e funcionalidades, as quais serão detalhadas através de modelos e diagramas. No capítulo ainda serão apresentadas as principais telas de apresentação do software e partes do código-fonte com o objetivo de exemplificar algumas tecnologias utilizadas para sua implementação.

4.1 VISÃO GERAL DO SISTEMA

O software implementado como resultado desse trabalho irá auxiliar no controle de informações e tomadas de decisão das corretoras, além de automatizar os processos e centralizar o acesso e busca de informações sobre apólices de seguros. Para a solução proposta foi considerados o contexto a seguir:

Qualquer pessoa interessada em contratar o serviço de uma seguradora é considerado um cliente em potencial. Os passos para a contratação de um seguro qualquer são simples. Primeiramente os possíveis clientes entram em contato com a corretora de seguros e solicitam um orçamento. O corretor realiza o cálculo do seguro conforme as condições pré-definidas com o cliente juntamente com a seguradora. Um cliente poderá contratar quantos seguros desejar a qualquer momento. Contudo cada seguro só poderá ser vinculado a apenas um único cliente.

Todos os seguros são divididos em categorias conhecidas como ramos. Cada ramo será composto por coberturas e cláusulas específicas ao ramo. Caso o cliente opte por contratar o seguro com a corretora, é elaborada uma proposta e a mesma é enviada diretamente a seguradora. A seguradora tem a opção de aceitar o risco e emitir a proposta de seguro gerando dessa forma um documento contratual chamado apólice ou ainda a seguradora pode não aceitar a proposta de seguro dessa forma não emitindo o contrato do seguro.

Durante o período de vigência do contrato ou apólice de seguro o usuário poderá alterar as condições do contrato, comprar e adicionar cláusulas ou ainda cancelar o contrato com a seguradora caso desejar. O cliente poderá acionar o seguro contratado a qualquer momento. O profissional responsável pela venda ou renovação do seguro dentro da corretora é conhecido como produtor. Podem existir

vários produtores dentro de uma corretora de seguros. Cada produtor tem a função de lançar todas as propostas contratadas e os sinistros relacionados a mesma dentro do sistema, além de acompanhar e controlar a emissão de renovações, seguros novos e sinistros.

O sistema gerenciará desde um simples cadastro de cliente, até a elaboração das propostas de seguro e, posteriormente, o acompanhamento da utilização dos serviços contratados pelo cliente.

O software manterá um controle da produção da corretora que será composta por seguros novos, renovações e endossos. Possibilitará o acompanhamento da emissão de propostas, sinistros e cancelamentos de apólices. Permitirá, ainda, aos usuários a emissão de relatórios gerenciais mantendo o controle de acesso e de escopo de informações por usuário.

O acesso dos usuários e o gerenciamento de informações do sistema devem ser feitos por meio do aplicativo *desktop* devidamente instalado em um servidor podendo ou não ser compartilhado em terminais.

4.2 MODELAGEM DO SISTEMA

A seguir está a lista de requisitos do sistema de gerenciamento de corretoras de seguros. Nessa listagem “F” significa funcional de requisito funcional.

- F01 – Cadastrar Clientes;
- F02 – Cadastrar Seguradoras;
- F03 – Cadastrar Produtores;
- F04 – Cadastrar Usuários;
- F05 – Cadastrar Tipos de Endosso;
- F06 – Cadastrar Profissões;
- F07 – Cadastrar Ramos de Atividade;
- F08 – Cadastrar Bancos;
- F09 – Cadastrar Franquias;
- F10 – Cadastrar Coberturas;
- F11 – Cadastrar Cláusulas;
- F12 – Cadastrar Ramos;
- F13 – Cadastrar Situação da Proposta;

- F14 – Registrar Nova Proposta;
- F15 – Registrar Aviso de Sinistro;
- F16 – Registrar Vistoria Prévia;
- F17 – Acompanhar Propostas;
- F18 – Acompanhar Sinistros;
- F19 – Acompanhar Renovações;
- F20 – Consultar Seguros por cliente;
- F21 – Emitir relatório de Seguros a serem renovados no mês;
- F22 – Emitir relatório de Seguros não renovados;
- F23 – Emitir relatório de Cancelamento de Seguros;
- F24 – Emitir relatório de Sinistros por Seguradora;

Baseados nos requisitos levantados são definidos os atores envolvidos nos processos e seus papéis. Após definidos os casos de uso, são descritas as funcionalidades básicas de cada caso de uso, bem como os atores responsáveis por cada uma. As referências cruzadas também são verificadas baseando-se nos requisitos que estejam diretamente vinculados.

No Quadro 2 são apresentados e descritos os principais casos de uso baseados na listagem de requisitos.

| Nome | Atores | Descrição | Referências Cruzadas |
|------------------------------------|--------------------------------|--|-----------------------------|
| F14–Registrar Nova Proposta | Secretária, Corretor, Produtor | O usuário deve cadastrar no sistema todas as propostas efetivadas pela corretora, informando todos os dados referentes a mesma. | F1, F2, F3, F4, F11 |
| F19–Acompanhar Renovações | Secretária, Corretor, Produtor | O usuário poderá pesquisar e verificar todas as propostas que estão próximas de seu vencimento e também alterar a situação das propostas conforme as renovações das mesmas forem ocorrendo. | F1, F2, F5, F3, F4, F11 |
| F17–Acompanhar Propostas | Secretária, Corretor, Produtor | O usuário poderá acompanhar e controlar a emissão de todas as propostas efetivadas pela corretora. Ele também poderá alterar a situação das propostas conforme a emissão das mesmas forem ocorrendo. | F1, F2, F3, F4, F5 |
| | | O usuário terá a opção de | F1, F6 |

| | | | |
|--|--------------------------------|--|---------------------|
| F15–Registrar Aviso de Sinistro | Secretária, Corretor, Produtor | cadastrar o aviso de sinistro selecionando o cliente, vinculando a proposta relacionada ao sinistro e em seguida informando os dados do aviso de sinistro. | |
| F18–Acompanhar Sinistros | Secretária, Corretor, Produtor | O usuário poderá acompanhar e controlar a finalização de todos os avisos de sinistros cadastrados. O usuário poderá ainda alterar a situação do sinistro. | F1, F2, F4, F9 |
| F16–Registrar Vistoria Prévia | Corretor, Produtor | O usuário terá a opção de registrar a vistoria prévia selecionando o cliente, informando os dados referentes a vistoria. | F8, F5, F1, F2, F18 |
| F20–Consultar Seguros por Cliente | Corretor | O sistema deverá permitir ao usuário a consulta de todos as propostas e seguros cadastrados separadamente por cliente. | F1, F5 |

Quadro 2 - Casos de uso

A partir dos casos de uso já identificados e descritos, foram analisados alguns aspectos referentes aos mesmos, levando em conta a listagem de requisitos gerais. Um conceito pode ser definido como um “objeto”, ou domínio, referente a algum processo ou atividade do software. É possível, geralmente, verificar os conceitos por meio dos casos de uso e selecionar os termos que representam informações transmitidas e recebidas pelo sistema.

O Quadro 2 exibe os conceitos encontrados a partir de uma leitura e análise sobre os requisitos e casos de uso do sistema. O Quadro está dividido em:

- *Conceito*: nome do conceito ou domínio.
- *I – A – E – C*: propriedades referentes aos conceitos sendo que (*I* - Inclusão, *A* – Alteração, *E* – Exclusão, *C* – Consulta).
- *Observação*: restrições sobre as propriedades relacionadas aos conceitos.
- *Referências cruzadas*: requisitos vinculados diretamente ao conceito.

| Conceito | I | A | E | C | Observação | Referências Cruzadas |
|--------------------------|---|---|---|---|--|----------------------|
| Tipos de Sinistro | X | X | X | X | Não podem ser excluídos tipos de sinistros que estejam vinculados a avisos de sinistros. | F08, F37, F05 |
| Seguradoras | X | X | X | X | Não podem ser excluídas seguradoras que já tenham | F05, F06, F07 |

| | | | | | | |
|-----------------------------|---|---|---|---|--|-------------------------|
| | | | | | propostas vinculadas a mesma. | |
| Ramos de Atividade | X | X | X | X | A exclusão só pode ser realizada se o ramo de atividade não estiver vinculado a nenhum cliente. | F08, F19, F37 |
| Ramos | X | X | X | X | Não poderão ser excluídos ramos que possuam propostas vinculadas ao mesmo. | F05, F07 |
| Endossos | | X | X | X | A inclusão deve ser tratada pelo caso de uso para este fim. | F05, F01, F02 |
| Tipos de Endosso | X | X | X | X | A exclusão só poderá ser feita caso não exista vínculo entre o tipo de endosso a ser excluído com o cadastro de alguma proposta de endosso no sistema. | F06, F01 |
| Tipos de Proposta | X | X | X | X | Não pode ser feita a exclusão de tipo de propostas que já tenham vínculo com propostas dentro do sistema. | F05, F07 |
| Franquias | X | X | X | X | Não poderão ser excluídas franquias que já estejam presentes no cadastro de propostas. | F22, F23, F32 |
| Vistoria Prévia | X | X | X | X | A inclusão deve ser tratada pelo caso de uso para este fim. | F25, F19 |
| Usuários | X | X | X | X | Não poderão ser excluídos usuários que já fizeram movimentações no sistema. | F04 |
| Propostas | X | | | X | A inclusão deve ser tratada pelo caso de uso para este fim. | F05, F06, F07 |
| Clientes | X | X | X | X | Não poderão ser excluídos clientes caso já exista uma proposta vinculado ao mesmo. | F05, F21 |
| Cláusulas | X | X | X | X | Não poderão ser excluídas cláusulas que possuam vínculos com propostas no sistema. | F05, F07, F06 |
| Coberturas | X | | | X | Não poderão ser excluídas coberturas que possuam vínculos com propostas no sistema. | F05, F07, F06, F31 |
| Situação da Proposta | X | X | X | X | Não poderá ser excluída ou alterada caso alguma proposta estiver vinculada. | F05, F07, F06, F32 |
| Bancos | X | X | X | X | Não podem ser alterados nem excluídos bancos que estejam relacionados a outros cadastros de propostas. | F05, F23, F08 |
| Profissões | X | X | X | X | A exclusão só poderá acontecer caso o registro não esteja presente em outro cadastro dentro do sistema. | F05, F06, F07, F23, F22 |
| Aviso de Sinistro | X | X | X | X | A inclusão deve ser tratada pelo caso de uso para este fim. | F08, F05, F01, F02 |
| Produtores | X | X | X | X | Não podem ser excluídos produtores que estejam | F03, F05, F06, F07, F22 |

| | | | | | | |
|----------------------------|---|---|---|---|---|------------------------------|
| | | | | | vinculados a propostas no sistema. | |
| Renovações | X | X | X | X | A inclusão deverá ser tratada pelo caso de uso referente. | F05, F06, F07, F01, F02, F11 |
| Tipos de Utilização | X | X | X | X | Não poderá ser excluído um tipo de utilização vinculado às propostas. | F06, F15, F16 |

Quadro 3 - Conceitos

O Quadro 4 apresenta as principais consultas e relatórios para a exibição de informações armazenadas, baseadas na análise de requisitos.

| Nome | Referências Cruzadas |
|---|-----------------------------|
| Emitir relatório de Seguros a serem renovados no mês. | F05, F06, F07 |
| Emitir relatório de Seguros não renovados. | F05, F06 |
| Emitir relatório de Cancelamento de Seguros. | F07, F05 |
| Emitir relatório de Sinistros por Seguradora. | F32, F05, F08, F38 |

Quadro 4 - Consultas e relatórios

O diagrama de atividades apresenta o fluxo da principal etapa do processo de registro de nova proposta, sendo que os demais fluxos dos processos se darão a partir do cadastro da proposta do seguro no sistema. Na Figura 12, é exibido o diagrama de atividades separado em duas entidades: usuário e gerenciador de corretoras, que correspondem aos atores do processo em questão.

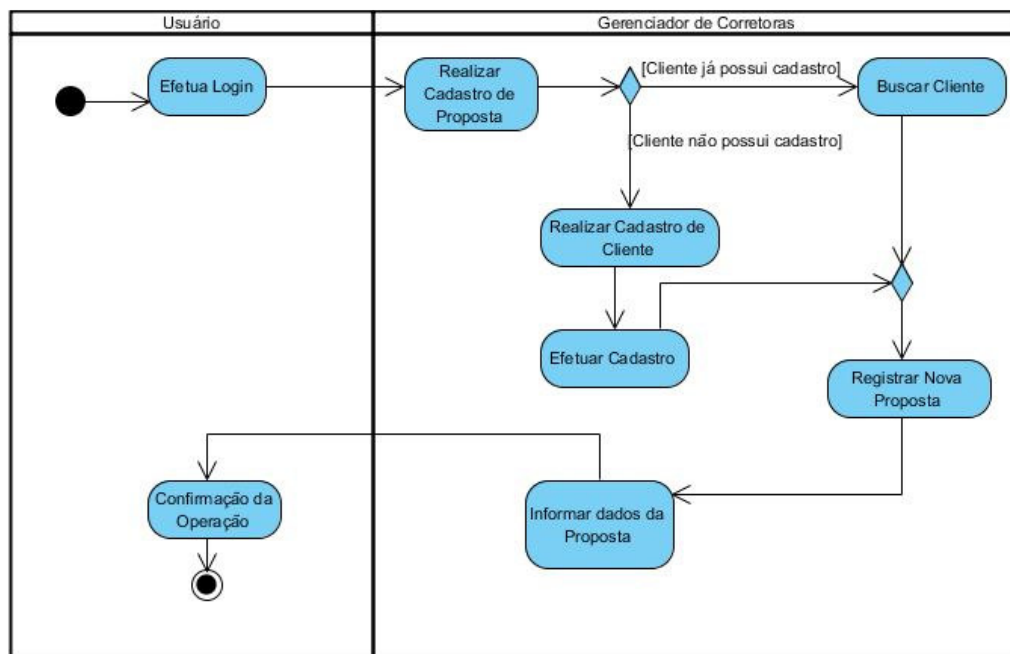


Figura 12 - Diagrama de atividades

Na Figura 13 é exibido o diagrama de caso de uso que foi elaborado com base nas informações obtidas pela listagem de requisitos. Nesse diagrama é possível visualizar claramente os atores e seus casos de uso correspondentes.

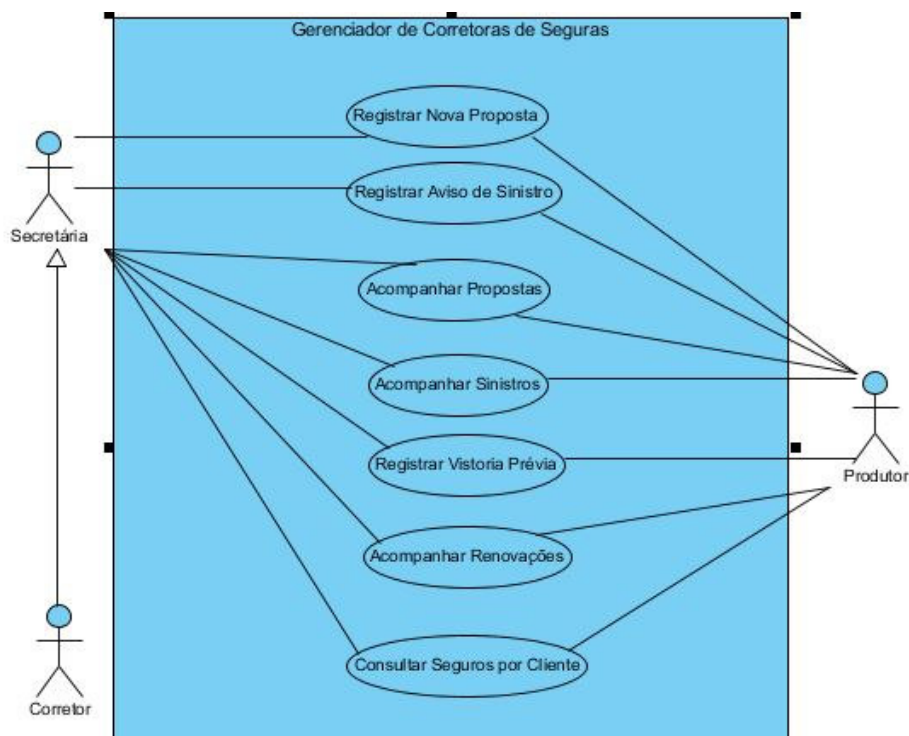


Figura 13 - Diagrama de caso de uso

A seguir são detalhados cada um dos casos de uso e seu respectivo fluxo principal, bem como seus fluxos alternativos, exceções e variantes. No Quadro 5 é detalhado o caso de uso registrar nova proposta.

| |
|--|
| <p>Caso de Uso: Registrar Nova Proposta O sistema deve permitir o cadastro de todas as propostas de seguro efetivadas na corretora. Conforme o tipo de proposta selecionada serão solicitadas informações específicas de cadastro de acordo com o ramo de seguro selecionado pelo usuário.</p> |
| <p>Atores: Corretor, Secretária, Produtor</p> |
| <p>Pré-condições: O respectivo cliente e seguradora devem estar devidamente cadastrados.</p> |
| <p>Pós-condições: O cadastro da proposta foi realizado com sucesso e o usuário recebe a confirmação do cadastro.</p> |
| <p>Fluxo Principal:</p> <ol style="list-style-type: none"> 1. [IN] O usuário faz <i>login</i> no sistema. 2. O usuário escolhe a opção de registro desejada. 3. [OUT] O Sistema apresenta as opções de ramos de seguro disponíveis para cadastro. 4. [IN] O Usuário seleciona o ramo desejado para o cadastro da proposta. 5. [OUT] O sistema apresenta a tela de cadastro da proposta. 6. [IN] O Usuário insere os dados da proposta (cliente, nº. do orçamento, data de |

| |
|--|
| emissão, tipo de proposta, nº. da apólice, seguradora). |
| 7. [OUT] O sistema gera a confirmação de cadastro e finaliza a operação. |
| Tratamento de Exceções: |
| 6a. O cliente não foi informado. 6a.1 O sistema solicita que seja informado o cliente; 6a.2 O usuário informa o cliente desejado. 6a.3 Retorna ao passo 1. 6b. A seguradora não foi informada. 6b.1 o sistema solicita que a seguradora seja selecionada; 6b.2 o usuário informa a seguradora. 6b.3 Retorna ao passo 1. |

Quadro 5 - Expansão do caso de uso: registrar nova proposta

No Quadro 6 está a expansão no caso de uso registrar aviso de sinistro.

| |
|---|
| Caso de Uso: Registrar Aviso de Sinistro O sistema deve permitir ao usuário registrar todos os sinistros ocorridos e vincular a proposta de seguro correspondente ao sinistro. |
| Atores: Secretária, Produtor, Corretor |
| Pré-condições: A proposta de seguro relacionada deve estar devidamente cadastrado no sistema. |
| Pós-condições: O aviso de sinistro é cadastrado no sistema. |
| Fluxo Principal: |
| <ol style="list-style-type: none"> 1. [IN] O usuário informa sua identificação. 2. O usuário escolhe a opção de registro desejada. 3. [OUT] O sistema exibe a tela de registro do aviso de sinistro. 4. [IN] O usuário informa os dados referentes ao aviso de sinistro (data do aviso, cliente, seguradora, número do sinistro e vincula a proposta ao aviso). 5. [OUT] O sistema grava o cadastro e exibe uma confirmação ao usuário. |
| Tratamento de Exceções: |
| 4a. Nenhuma proposta esta vinculada ao aviso de sinistro. 4a.1 O sistema informa que não é possível registrar o aviso de sinistros sem vincular uma proposta de seguro. 4a.2 O usuário vincula a proposta correspondente ao aviso do sinistro. 4a.3 Retorna ao passo 3. 5a. A proposta está cancelada. 5a.1 o sistema informa que não é possível registrar o aviso de sinistro para propostas que estão com o <i>status</i> de cancelada no sistema. 5a.2 o usuário confere a situação da proposta com a seguradora, caso a mesma esteja vigente ainda, ele altera a situação da proposta, caso a proposta esteja realmente cancelada o caso de uso é abortado. 5a.3 Retorna ao fluxo principal. |

Quadro 6 - Expansão do caso de uso: registrar aviso de sinistro

A expansão do caso de uso acompanhar renovações está no Quadro 7.

| |
|---|
| Caso de Uso: Acompanhar Renovações O sistema permitirá ao usuário acompanhar e controlar o vencimento das propostas de seguros cadastradas. O usuário poderá pesquisar todas as propostas cadastradas e alterar a situação da proposta conforme forem ocorrendo as renovações do seguros. |
|---|

| |
|---|
| Atores: Corretor |
| Pré-condições: A proposta deve estar devidamente cadastrada. |
| Pós-condições: A situação da proposta é alterada. |
| Fluxo Principal: <ol style="list-style-type: none"> 1. [IN] O usuário faz <i>login</i> no sistema. 2. O usuário escolhe a opção de registro desejada. 3. [OUT] O sistema exibe as propostas a serem renovadas para o usuário. 4. [IN] O usuário seleciona a proposta desejada e altera sua situação. 5. [OUT] O sistema atualiza a situação da proposta e sinaliza a mesma. |
| Tratamento de Exceções: <ol style="list-style-type: none"> 3a. Não existem nenhuma situação cadastrada para a proposta. <ol style="list-style-type: none"> 3a.1 O sistema informa que não existem situações de propostas cadastradas. 3a.2 O usuário efetua o cadastro de situação de propostas. 3a.3 Retorna ao passo 3. |

Quadro 7 - Expansão do caso de uso: acompanhar renovações

No Quadro 8 está a expansão do caso de uso registrar vistoria prévia.

| |
|--|
| Caso de Uso: Registrar Vistoria Prévia O sistema deve possibilitar ao usuário o registro de vistoria prévia realizadas e também o vínculo da mesma com o cliente. |
| Atores: Corretor |
| Pré-condições: O cliente e a seguradora devem estar devidamente cadastrados. |
| Pós-condições: A vistoria prévia é cadastrada no sistema. |
| Fluxo Principal: <ol style="list-style-type: none"> 1. [IN] O usuário faz <i>login</i> no sistema. 2. O usuário escolhe a opção de registro desejada. 3. [OUT] O sistema apresenta a tela de registro vistoria prévia. 4. [IN] O usuário informa todos os dados pertinentes à vistoria (número da vistoria, seguradora, cliente, data da vistoria). 5. [OUT] O sistema realiza o cadastro do registro e finaliza a operação. |
| Tratamento de Exceções: <ol style="list-style-type: none"> 4.a O cliente não é informado. <ol style="list-style-type: none"> 4.a1 O sistema informa que é necessário informar um cliente. 4.a2 O usuário informa o cliente desejado. 4.a3 Retorna ao passo 4. |

Quadro 8 - Expansão do caso de uso: registrar vistoria prévia

Na Figura 14 é apresentado o diagrama de classes do software. Nesse diagrama são apresentados apenas os atributos das classes.

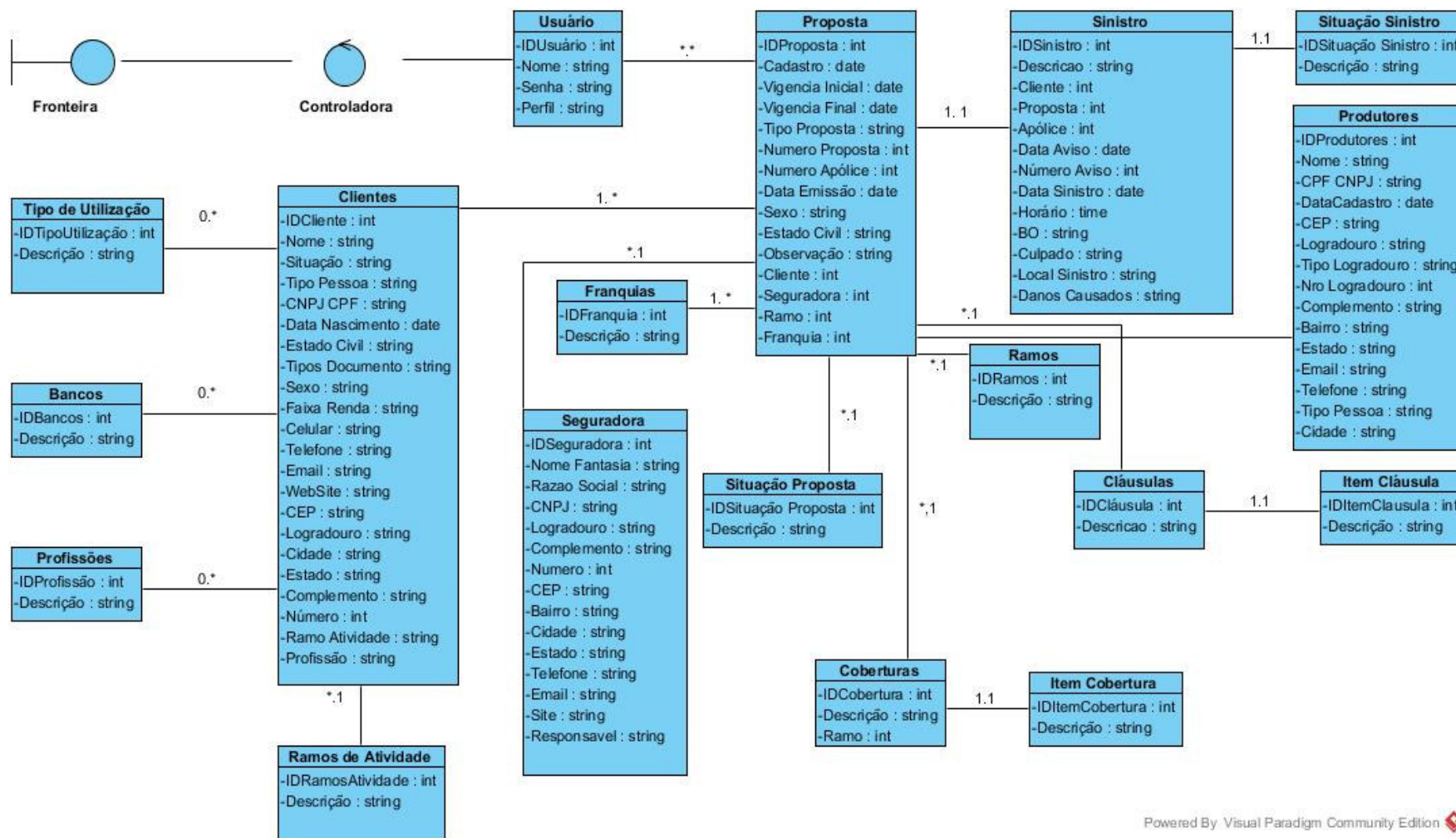


Figura 14 - Diagrama de classes
Fonte: autoria própria

A maioria das classes no sistema terá, basicamente, uma ligação direta ou indiretamente com a classe principal denominada Proposta. Será a partir dessa classe raiz que poderão ser elaboradas as demais funcionalidades dentro do sistema e a obtenção de informações.

A Figura 15 exibe as principais ligações e estrutura do cadastro de propostas. A classe principal está ligada a Controladora e esta por sua vez ligada a Fronteira, entendida também como interface.

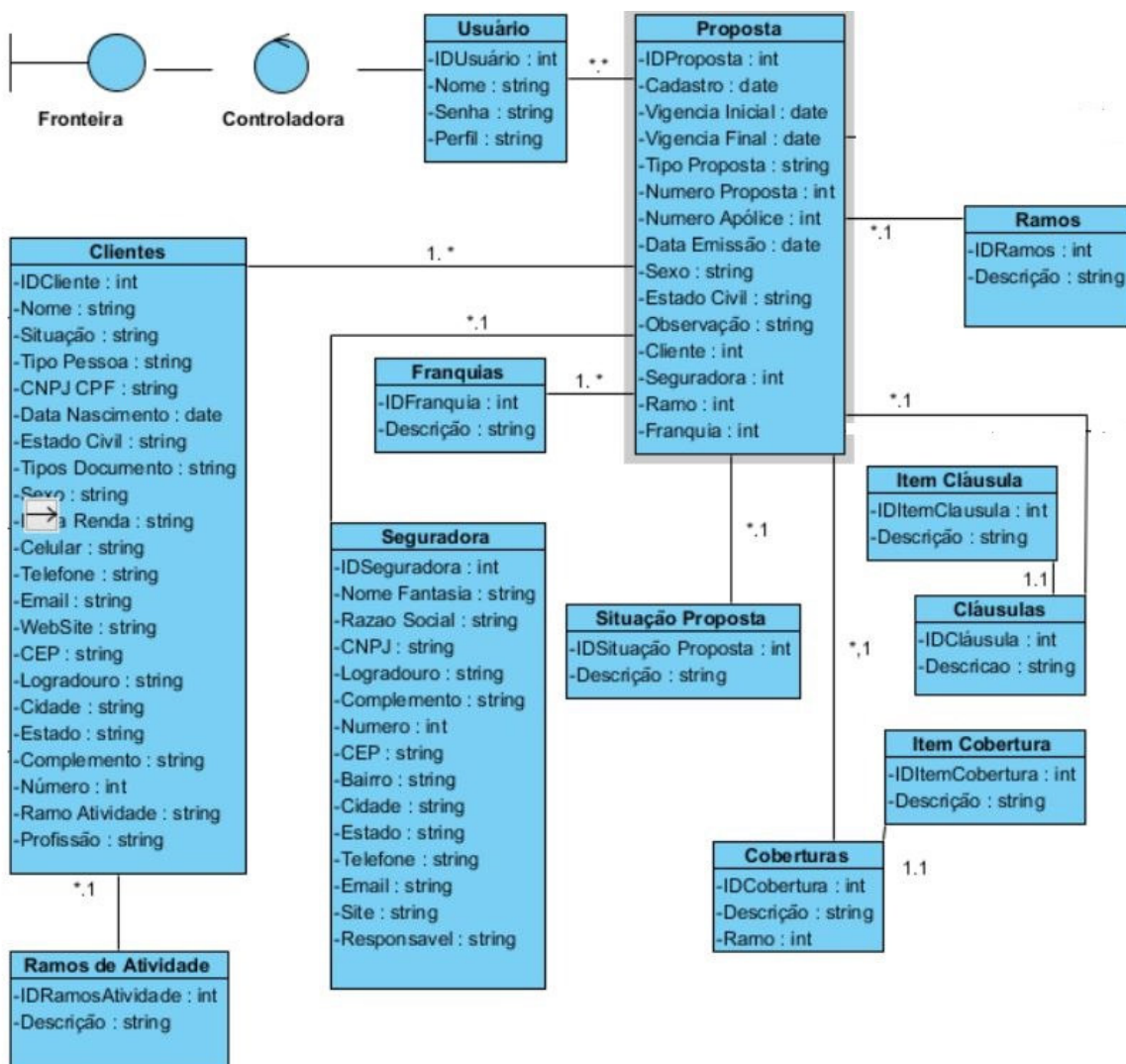


Figura 15 - Diagrama de classes (registrar nova proposta)
Fonte: autoria própria

Nas Figuras 16 a 19 são apresentados os diagramas de sequência para os principais casos de uso que tiveram sua expansão apresentada nos Quadros 5 a 8.

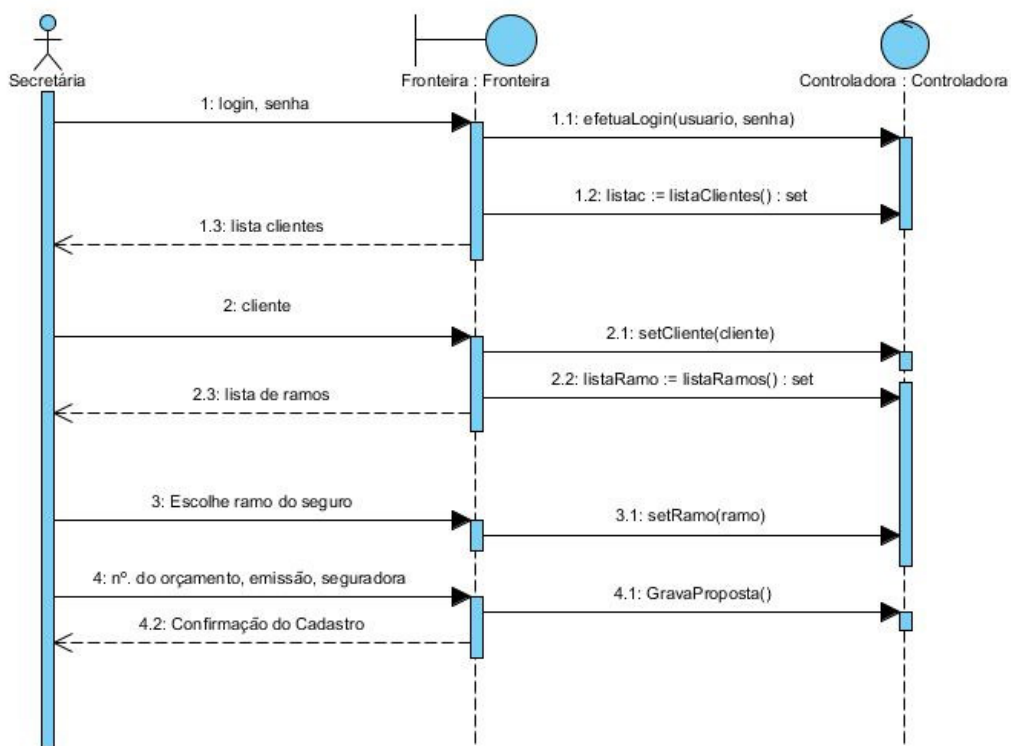


Figura 16 - Diagrama de seqüência do caso de uso registrar nova proposta

Fonte: autoria própria

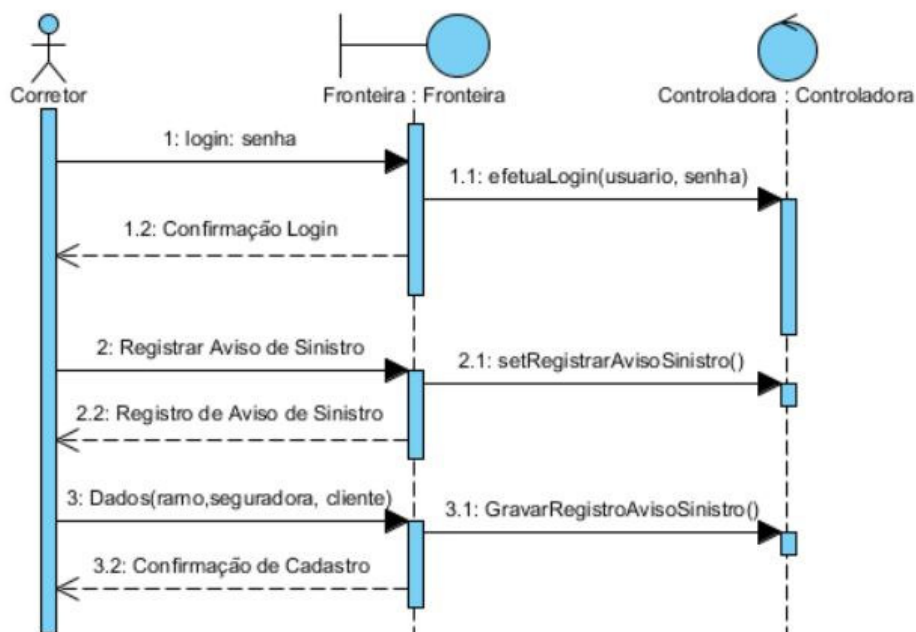


Figura 17 - Diagrama de seqüência do caso de uso registrar aviso de sinistro

Fonte: autoria própria

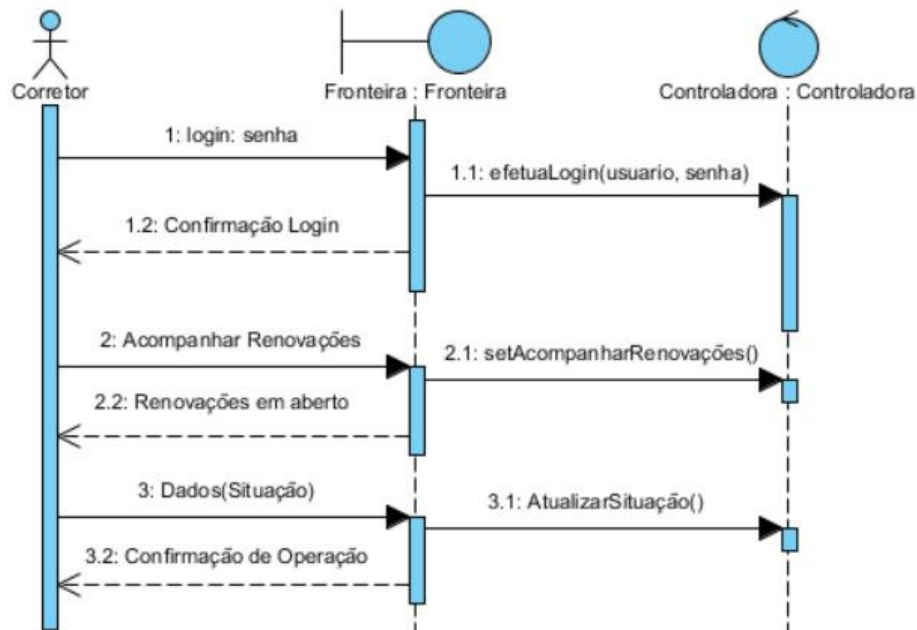


Figura 18 - Diagrama de seqüência do caso de uso acompanhar renovações
Fonte: autoria própria

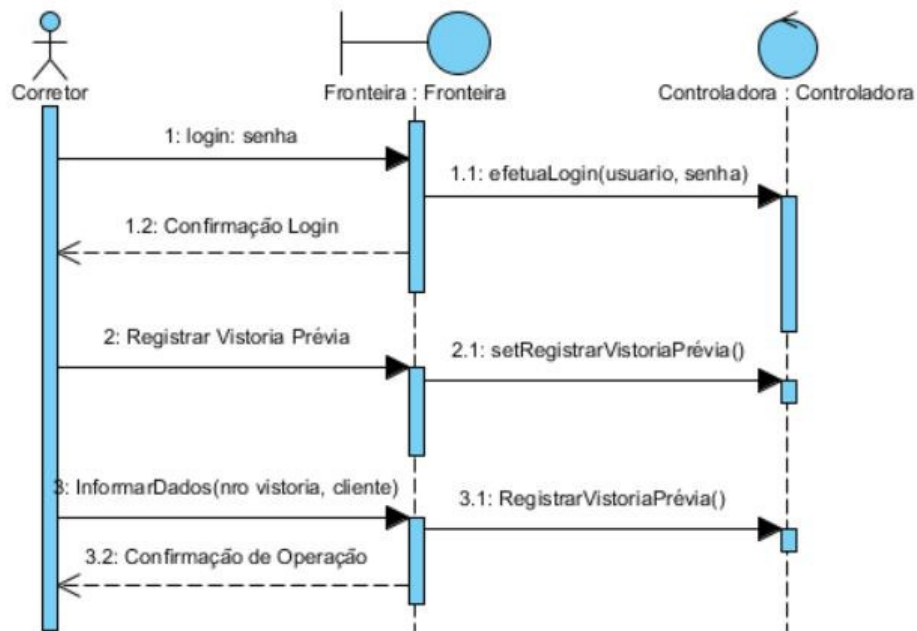


Figura 19 - Diagrama de seqüência do caso de uso registrar vistoria prévia
Fonte: autoria própria

O Diagrama de comunicação enfatiza a organização estrutural dos objetos que trocam mensagens. Esse diagrama corresponde a um formato alternativo para descrever interação entre objetos. Nas Figuras 20 a 23 são apresentados os

diagramas de comunicação para os principais casos de uso que tiveram sua expansão apresentada anteriormente.

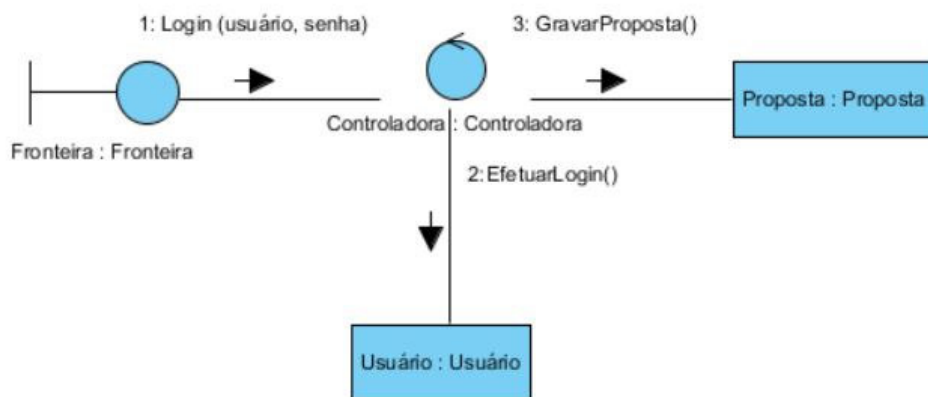


Figura 20 - Diagrama de comunicação do caso de uso registrar nova proposta
Fonte: autoria própria

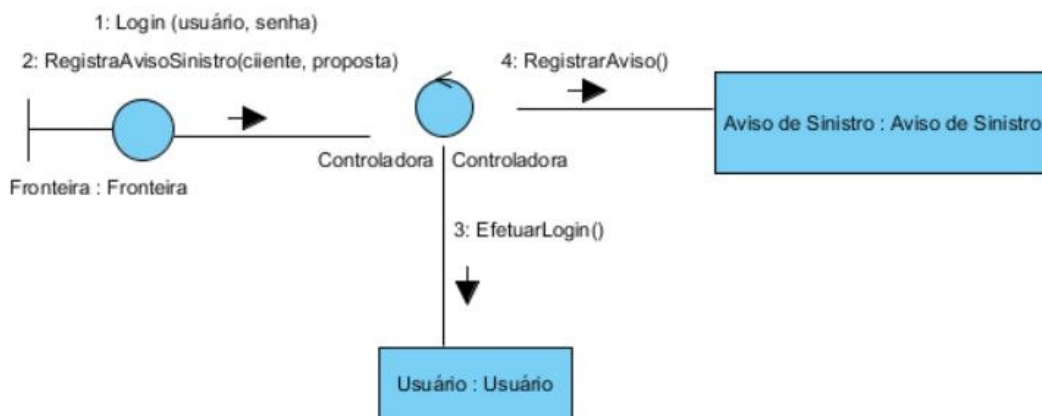


Figura 21 - Diagrama de comunicação do caso de uso registrar aviso de sinistro
Fonte: autoria própria

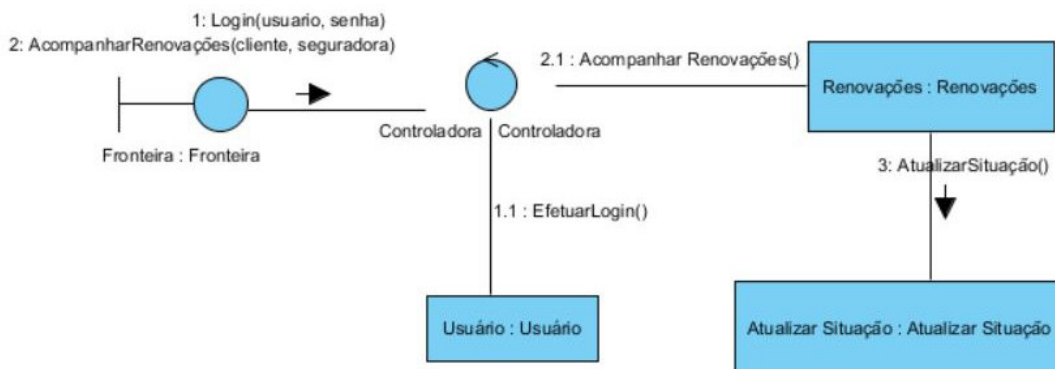


Figura 22 - Diagrama de comunicação do caso de uso acompanhar renovações
Fonte: autoria própria

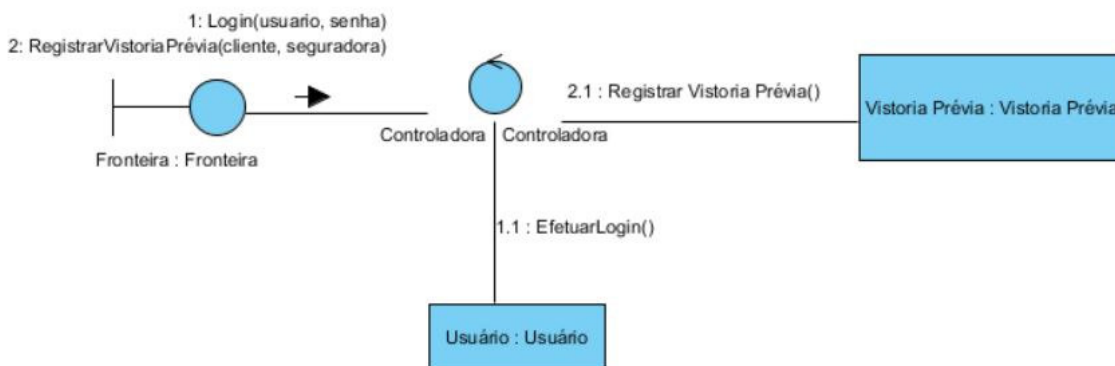


Figura 23 - Diagrama de comunicação do caso de uso registrar vistoria prévia
 Fonte: autoria própria

4.3 APRESENTAÇÃO DO SISTEMA

A inicialização do sistema é composta por três eventos distintos: ao executar o programa é exibida uma tela de apresentação enquanto o sistema é carregado, após isso é exibida a tela de *login* do sistema, a partir disso o usuário informa *login* e senha e, se corretos (no sentido de estarem armazenados no banco de dados), é apresentada a página inicial do sistema. A Figura 24 mostra a tela de *login* do sistema, nela serão validadas as informações de usuário e senha.

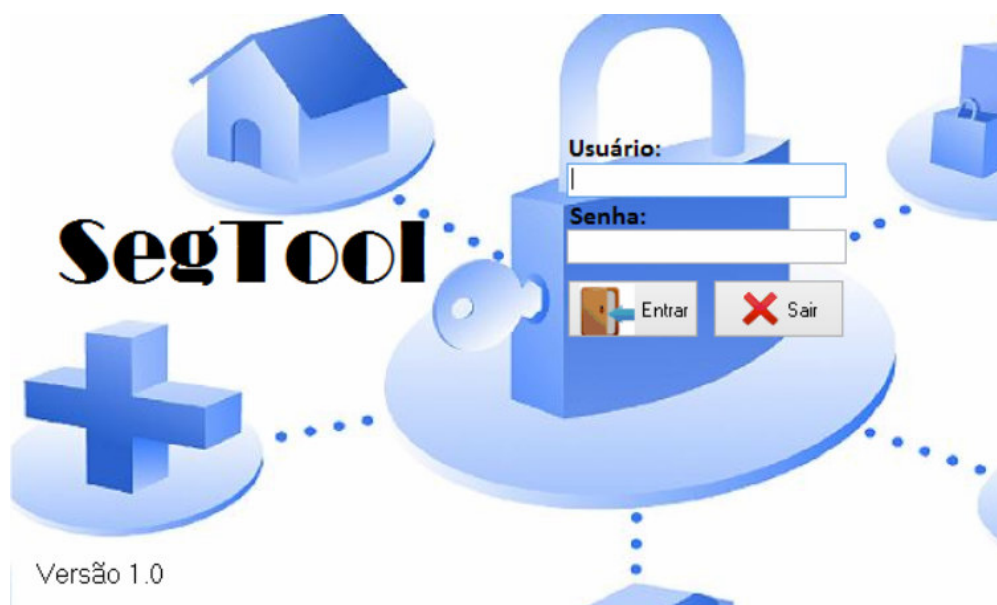


Figura 24 – Tela de login
 Fonte: autoria própria

A Figura 25 apresenta a tela de navegação inicial. A tela inicial é dividida em quatro menus principais: cadastros, movimentações, relatórios e ajuda. O menu

cadastros contém todos os cadastros do sistema agrupados por grupos e subgrupos. Dentro do menu movimentações estão os processos e as operações. No menu relatório são acessados os relatórios disponíveis. Por fim, no menu ajuda é apresentada uma tela informativa contendo os principais dados do software em uso. A tela inicial é composta por um menu de acesso rápido ao usuário contendo os principais cadastros e movimentações.



Figura 25 – Tela inicial do sistema
Fonte: autoria própria

A Figura 26 mostra um dos principais cadastros do sistema que é o cadastro de clientes. Todas as telas de cadastros estão divididas em duas abas principais: pesquisa e manutenção. Na aba pesquisa o usuário tem a possibilidade de pesquisar todos os clientes cadastrados ou filtrar o cliente desejado.

Cadastro de Clientes

Pesquisa | Manutenção

Código: 1

| Código | Data | Situação | Cliente | CNPJ/CPF | Data Nascimento | Estado Civil | Tipo Documento |
|--------|------------|----------|---------------------------------|-------------|-----------------|---------------|----------------|
| 1 | 01/06/2015 | Ativo | Aline Heloize Costa Vergo Polan | 00854545454 | 24/10/1991 | Não informado | RG |

1- Aba de Pesquisa.

2- Filtros da Pesquisa.

Figura 26 – Aba pesquisa do formulário de cadastro de clientes
 Fonte: autoria própria

Na Figura 27, dentro da aba manutenção o usuário poderá realizar as operações básicas de cadastro como inclusão, alteração e exclusão.

Cadastro de Clientes

Pesquisa | **Manutenção**

Dados Gerais | Endereço | Contato | Dados Bancários

1- Aba de Manutenção.

Código: Data: Pessoa: Nome/Razão Social:

CPF/CNPJ: Situação: Ramo de Atividade: Responsável:

Sexo: Data de Nascimento: Estado Civil: Faixa de Renda: Profissão:

Tipo de Documento: Número do Documento: Orgão Emissor: Data de Emissão:

Número CNH: Validade CNH: Categoria CNH: Primeira Habilitação:

2- Botões de inclusão, alteração, cancelamento e exclusão

Novo Cancelar Salvar Excluir Sair

Figura 27 – Aba manutenção do formulário de cadastro de clientes
 Fonte: autoria própria

Na maioria dos cadastros é realizada a validação de campos obrigatórios. Se o usuário clicar no botão “Salvar” sem informar os campos obrigatórios o sistema fará a validação e exibirá uma mensagem para o usuário, como mostra a Figura 28.

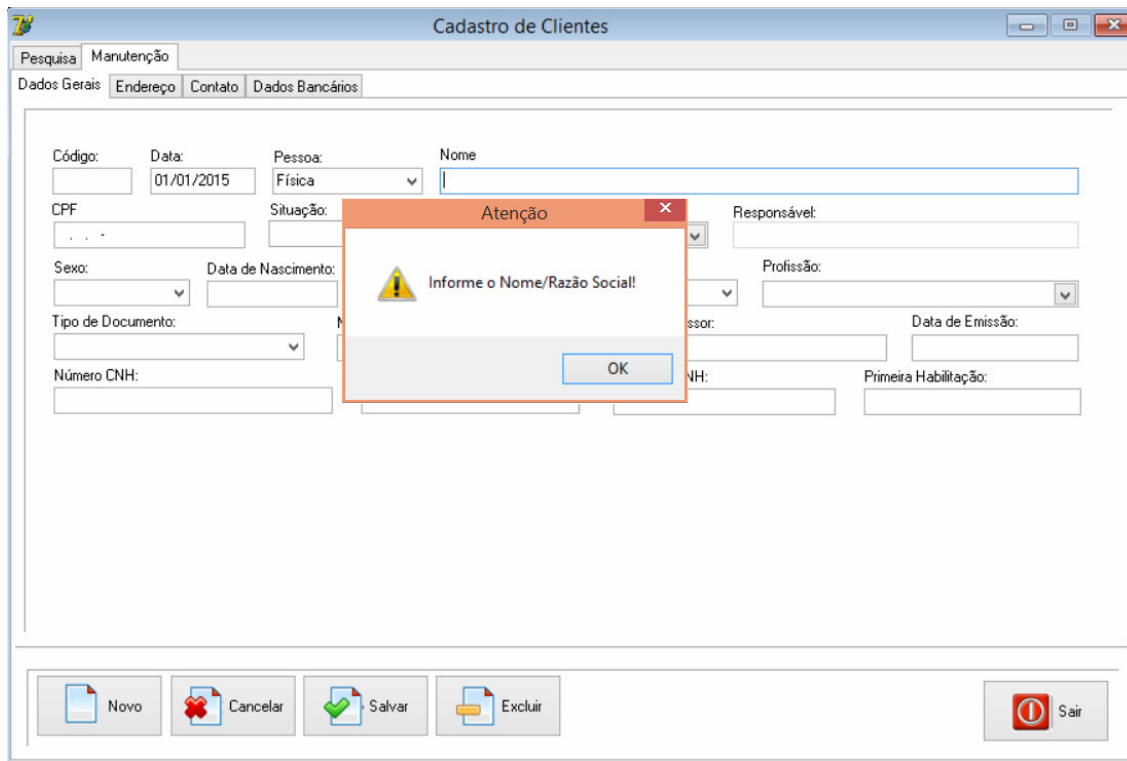


Figura 28 – Exemplo de validação de campos obrigatórios
Fonte: autoria própria

Algumas máscaras e *labels* dos campos de cadastro irão ser validados em tempo real no sistema. Por exemplo, na Figura 29, o campo ‘Pessoa’ definirá a máscara e o rótulo dos campos Nome/Razão Social e CPF/CNPJ automaticamente na tela.

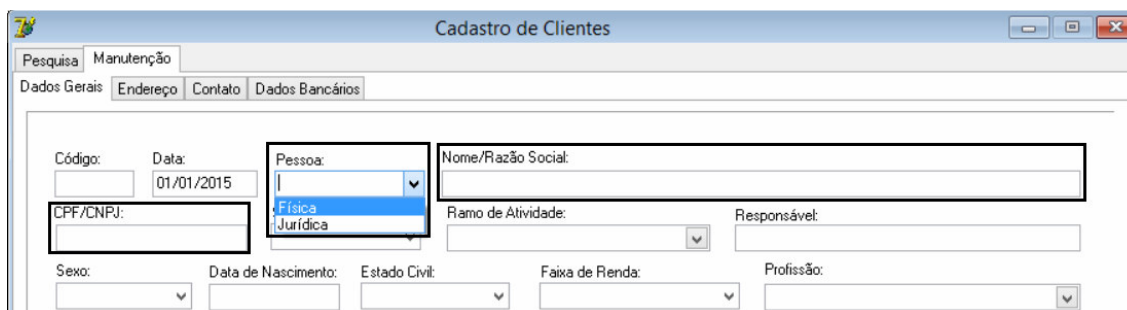
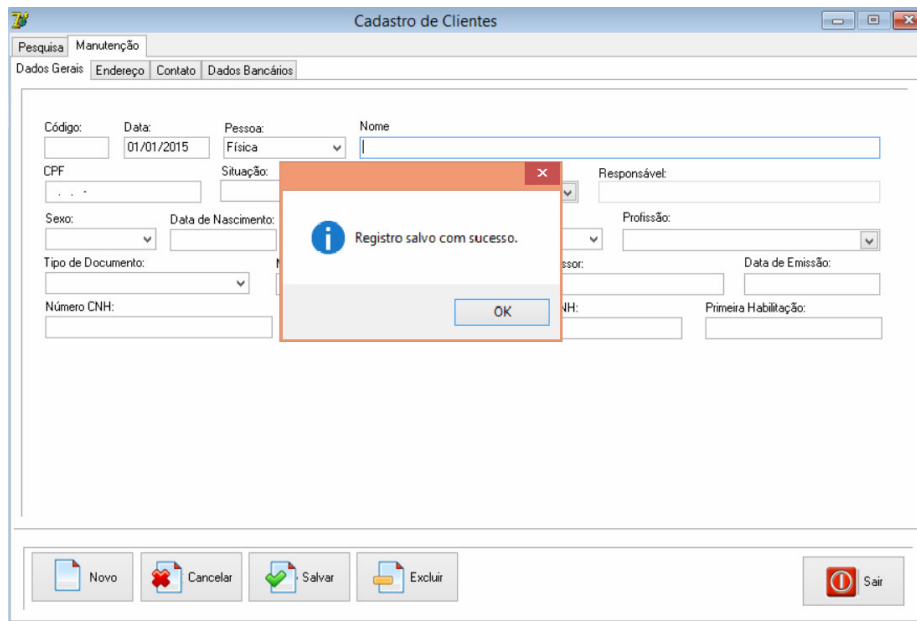


Figura 29 – Exemplo de validação de máscaras e labels
Fonte: autoria própria

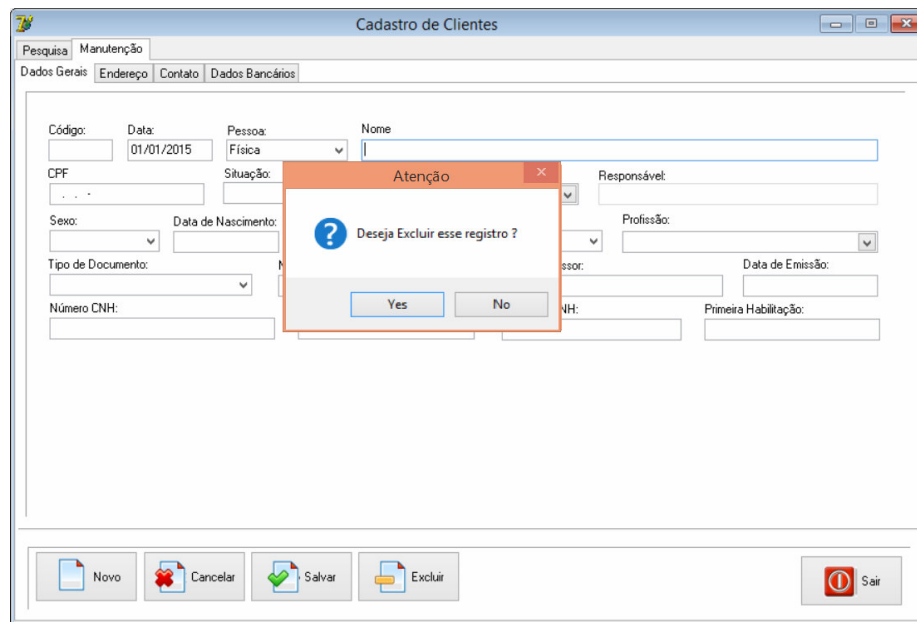
Caso todos os campos obrigatórios estejam preenchidos, o sistema realizará a gravação do cadastro e exibirá a confirmação do cadastro para o usuário, apresentado na Figura 30.



The screenshot shows the 'Cadastro de Clientes' application window. The 'Dados Gerais' tab is active, displaying various input fields for client registration. A modal dialog box is centered on the screen, titled 'Registro salvo com sucesso.' (Record saved successfully), with an information icon and an 'OK' button. The background form includes fields for 'Código', 'Data' (01/01/2015), 'Pessoa' (Física), 'Nome', 'CPF', 'Situação', 'Sexo', 'Data de Nascimento', 'Tipo de Documento', 'Número CNH', 'Responsável', 'Profissão', 'Data de Emissão', and 'Primeira Habilitação'. At the bottom, there are buttons for 'Novo', 'Cancelar', 'Salvar', 'Excluir', and 'Sair'.

Figura 30 – Formulário para inclusão de clientes
Fonte: autoria própria

Na Figura 31 é apresentada a funcionalidade de exclusão do cadastro. Ao executar essa operação, o usuário será alertado se deseja realizar a exclusão do cadastro realmente e em caso afirmativo, sistema removerá o cadastro.



The screenshot shows the 'Cadastro de Clientes' application window with the same form as in Figure 30. A modal dialog box is centered on the screen, titled 'Atenção' (Attention), with a question mark icon and the text 'Deseja Excluir esse registro?' (Do you want to delete this record?). The dialog has 'Yes' and 'No' buttons. The background form and bottom navigation buttons are visible but slightly dimmed.

Figura 31 – Formulário para exclusão de clientes
Fonte: autoria própria

Na Figura 32 é listada a tela de acompanhamento de propostas, através dessa tela o usuário poderá controlar a emissão todas as propostas e ainda atualizar a situação correspondente.

| Código | Vigência Inicial | Vigência Final | Tipo de Proposta | Número Proposta | Cliente | Seguradora | Ramo | Situação |
|--------|------------------|----------------|------------------|-----------------|-----------------------|--------------------|-----------|----------|
| 16 | 01/01/2015 | 01/11/2015 | Seguro Novo | 124587558 | João da Silva Pereira | Allianz Seguradora | Automóvel | |
| 7 | | | | | | | | |
| 1 | | | | | | | | |
| 2 | | | | | | | | |
| 3 | | | | | | | | |
| 8 | | | | | | | | |
| 9 | | | | | | | | |
| 10 | | | | | | | | |
| 12 | | | | | | | | |
| 13 | | | | | | | | |
| 14 | | | | | | | | |
| 15 | | | | | | | | |

Figura 32 – Formulário de acompanhamento das propostas
Fonte: autoria própria

4.4 IMPLEMENTAÇÃO DO SISTEMA

Para a implementação do software de gerenciamento de corretoras de seguros, foi utilizada a plataforma de desenvolvimento Embarcadero Delphi. E para a construção das interfaces do sistema e implementação das classes, funções e métodos que compõem o aplicativo foi utilizada a linguagem de programação Delphi 7. Para o armazenamento de dados foi usado o banco de dados Firebird.

A Listagem 1 mostra um trecho do código-fonte retirado da Unit_Menu pertencente ao formulário do menu do sistema. Dentro da Unit_Menu são declaradas todas as classes e métodos utilizados pelo formulário. Nesta Unit também são declaradas *procedures* para a chamada de todos os formulários do menu do sistema.

```

unit Unit_Menu;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, StdCtrls, Buttons;

type
  TForm_Menu = class(TForm)
    MainMenu: TMainMenu;
    Movimentaol: TMenuItem;
    Relatrios1: TMenuItem;
    Ajuda1: TMenuItem;
    Sobre1: TMenuItem;
    Label1: TLabel;
    Seguros1: TMenuItem;
    A1: TMenuItem;
    Residencial: TMenuItem;
    Empresarial: TMenuItem;
    Vidal: TMenuItem;
    Cadastros1: TMenuItem;
    Clientes1: TMenuItem;
    Produtores1: TMenuItem;
    Seguradoras1: TMenuItem;
    Usurios1: TMenuItem;
    Clientes2: TMenuItem;
    Profissoes1: TMenuItem;
    RamosdeAtividade1: TMenuItem;
    Produtores2: TMenuItem;
    Seguradoras2: TMenuItem;
    Coberturas1: TMenuItem;
    Clusulas1: TMenuItem;
    Franquias1: TMenuItem;
    iposdeEndosso1: TMenuItem;
    Usurios2: TMenuItem;
    AcompanhamentodePropostas1: TMenuItem;
    Bancos2: TMenuItem;
    Ramos1: TMenuItem;
    ipodeUtiliza1: TMenuItem;
    AcompanharSinistros1: TMenuItem;
    AcompanharRenovaes1: TMenuItem;
  end;

```

```

procedure btClausulasClick(Sender: TObject);
procedure btBancosClick(Sender: TObject);

private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form_Menu: TForm_Menu;

implementation

uses Unit_Corretora, Unit_Usuario, Unit_Coberturas, Unit_Ramos,
  Unit_Clausulas, Unit_RelacaoSegurado, Unit_Seguradora, Unit_Profissao,
  Unit_TipoUtilizacao, Unit_TipoEndosso, Unit_SegurosRenovar,
  Unit_RegistroSeguro, Unit_Cliente, Unit_SeguroVida,
  Unit_SeguroEmpresarial, Unit_RegistrarSeguro, Unit_Bancos,
  Unit_RamoAtividade, Unit_AcompanhamentoPropostas, Unit_RegistrarSinistro;

{$R *.dfm}

procedure TForm_Menu.FormCreate(Sender: TObject);
begin
  if not Assigned(Form_Menu) then
    Form_Menu := TForm_Menu.Create(nil)
end;

procedure TForm_Menu.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  Application.Terminate;
end;

procedure TForm_Menu.Bancos2Click(Sender: TObject);
begin
  if not Assigned(Form_Bancos) then
    Form_Bancos := TForm_Bancos.Create(nil)
  else
    ShowMessage('Formulário Criado');
    Form_Bancos.ShowModal;
end;

```

Listagem 1 – Unit_Menu

Na linguagem Delphi todas as telas do sistema são conhecidas como formulários e cada formulário está devidamente ligado a uma *unit* que basicamente é o código-fonte desse formulário. No exemplo da Listagem 1, o formulário de menu está ligado com a Unit_Menu. Dentro da Unit_Menu foi criado a *procedure* TForm_Menu.Bancos2Click(Sender: TObject). Para cada item do menu foi criada uma *procedure* correspondente. Nesse trecho de código a *procedure* verificará se o formulário Form_Bancos referente ao cadastro de bancos já existe, caso ainda não exista a *procedure* criará a tela em tempo de execução e a exibirá em seguida.

Para armazenar e centralizar componentes como TSQLConnection, TSQLTable, TSQLQuery, TSQLDataSet, TDataSetProvider, TClientDataSet e TDataSource foi utilizado um módulo de dados nomeado como Unit_DM. Esse módulo foi utilizado com o objetivo de facilitar o acesso e a configuração de todos os componentes da conexão do banco de dados e manipulação de todos os dados da aplicação. Todos os formulários do sistema fazem acesso aos componentes da Unit_DM para manipulação de dados. Na Listagem 2 é apresentado parte do código do módulo de dados do sistema.

```

unit Unit_DM;

interface

uses
  SysUtils, Classes, DBXpress, DB, SqlExpr, FMTBcd, IBCustomDataSet,
  IBTable, IBDatabase, IBQuery, ADODB, DBClient, Provider;

type
  TDM = class(TDataModule)
    SQLConnection: TSQLConnection;
    SQLTCorretora: TSQLTable;
    DSPCorretora: TDataSetProvider;
    CDSCorretora: TClientDataSet;
    SQLTCorretoraRAZAOSOC: TStringField;
    SQLTCorretoraNUMECNPJ: TStringField;
    SQLTCorretoraDATACADA: TDateField;
    SQLTCorretoraNUMERCEP: TStringField;
    SQLTCorretoraDESCLOGR: TStringField;
    SQLTCorretoraTIPOLOGR: TStringField;
    SQLTCorretoraNUMELOGR: TIntegerField;
    SQLTCorretoraCOMPLOGR: TStringField;
    SQLTCorretoraDESCBAIR: TStringField;
    SQLTCorretoraDESCCIDA: TStringField;
    SQLTCorretoraDESCESTA: TStringField;
    SQLTCorretoraDESCEMAI: TStringField;
    SQLTCorretoraDESCTELE: TStringField;
    CDSCorretoraRAZAOSOC: TStringField;
    CDSCorretoraNUMECNPJ: TStringField;
    CDSCorretoraDATACADA: TDateField;
    CDSCorretoraNUMERCEP: TStringField;
    CDSCorretoraDESCLOGR: TStringField;
    CDSCorretoraTIPOLOGR: TStringField;
    CDSCorretoraNUMELOGR: TIntegerField;
    CDSCorretoraCOMPLOGR: TStringField;
    CDSCorretoraDESCBAIR: TStringField;
    CDSCorretoraDESCCIDA: TStringField;
    CDSCorretoraDESCESTA: TStringField;
    CDSCorretoraDESCEMAI: TStringField;
    CDSCorretoraDESCTELE: TStringField;
    CDSCorretoraCODCORRE: TIntegerField;
    SQLTCorretoraCODCORRE: TIntegerField;
  end;

```

Listagem 2 – Unit_DM

Dentro do componente visual da Unit_DM é possível adicionar e manipular todos os componentes utilizados pelos formulários por meio de suas propriedades. O componente responsável pela conexão da aplicação com o banco de dados é o componente SQLConnection. O componente SQLTCorretora do tipo SQLTable será responsável por fazer as requisições de dados ao componente de conexão do banco. Os componentes DSPCorretora, CDSCorretora e DSCorretora serão responsáveis pelo tráfego de dados entre a aplicação e o banco de dados e o armazenamento de dados em memória em tempo de execução de operação. A Figura 29 apresenta os componentes visuais do módulo DM.

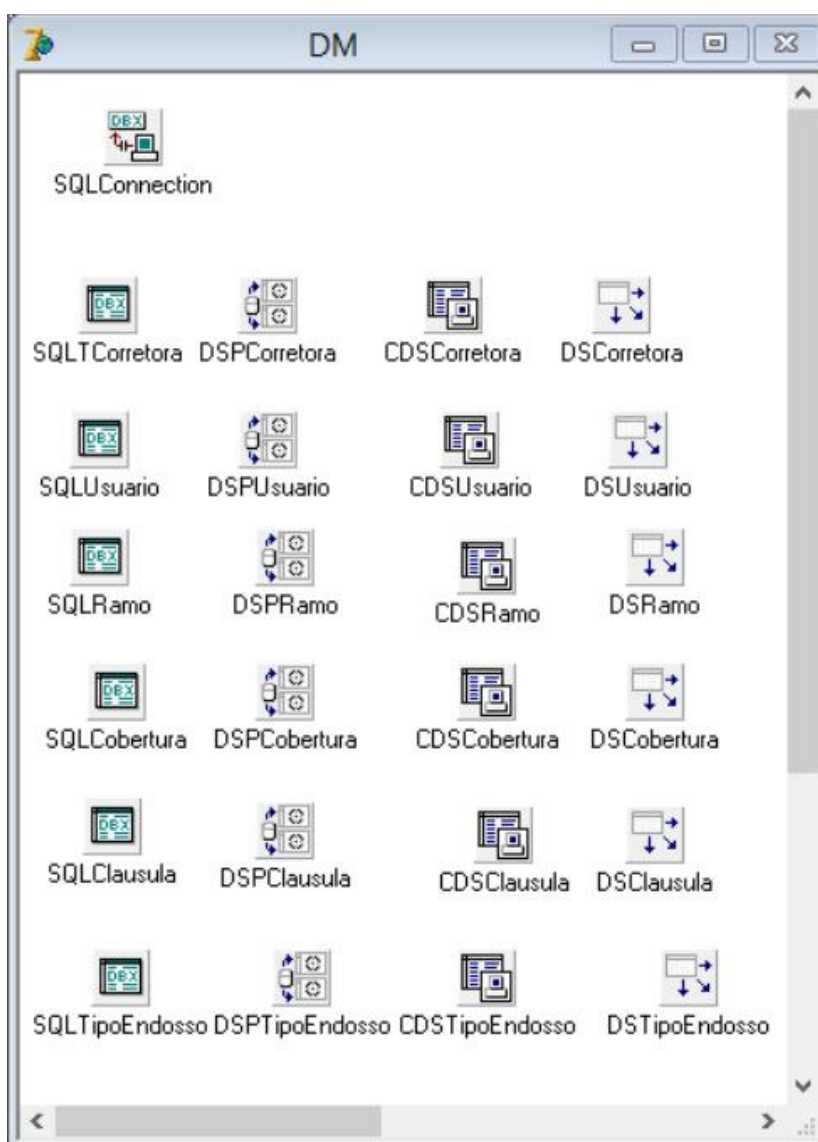


Figura 33 – Componentes visuais do módulo de dados DM
Fonte: obtido a partir de print screen da IDE linguagem Delphi

O trecho de código da Listagem 3 é referente à *procedure* de gravação do cadastro de produtores. Na primeira verificação é feita a checagem do estado do componente Dataset, se o mesmo está em modo de inserção ou edição, caso esteja o sistema irá chamar a função `DadosValidos()`. A função `DadosValidos()` verificará se todos os campos obrigatórios foram preenchidos em tela, se o resultado da função for verdadeiro (*true*), ou seja, todos os dados obrigatórios foram informados, o sistema gravará as informações no banco de dados e exibirá a mensagem de confirmação para o usuário.

```
procedure TForm_Produtores.btSalvarClick(Sender: TObject);
var Resultado: Boolean;
begin
  if DSConsulta.State in [dsInsert, dsEdit] then
  begin
    Resultado := DadosValidos();
    if Resultado = True then
    begin
      CDConsulta.Post;
      CDConsulta.ApplyUpdates(-1);
      CDConsulta.Close;
      ShowMessage('Registro Salvo com sucesso');
      edtData.setFocus;
    end
  end
else
  ShowMessage('Atenção: Favor executar uma operação.');
```

Listagem 3 – Trecho do código da Unit_Produtores

5 CONCLUSÃO

O estudo de caso deste trabalho abordou o desenvolvimento da análise e possibilitou a implementação de um sistema para a gestão de corretoras de seguros, concentrando-se nos principais processos realizados em uma corretora de seguros que são cadastro e acompanhamento de propostas, endossos e avisos de sinistros. O objetivo foi criar uma ferramenta para auxiliar corretoras na organização e controle de informações e na realização das atividades.

Para a realização da modelagem foi utilizada a linguagem UML e a ferramenta case Visual Paradigm para a construção dos diagramas. Na implementação do software foi utilizada a linguagem de programação Delphi e para o armazenamento de dados o banco de dados Firebird.

As abordagens apresentadas referentes aos modelos e diagramas UML, modelagem de software e métodos de desenvolvimento, explanadas ao longo do trabalho conceituaram as tecnologias e ferramentas utilizadas e auxiliaram no processo de desenvolvimento do software.

Conclui-se que ao final do trabalho que o software implementado atende às principais funcionalidades operacionais definidas como requisitos para o sistema. As telas foram desenvolvidas visando facilitar o uso do sistema. O aplicativo desenvolvido auxiliará as corretoras de seguros na gestão e controle de informações de maneira mais fácil que de forma manual com o uso de formulários e fichas de controle impressas.

Como trabalhos futuros visando complementar o que foi implementado, podem ser citados a realização da complementação dos requisitos, a implementação de novas funcionalidades, como o controle de comissões e a integração do sistema com outros sistemas externos das seguradoras.

REFERÊNCIAS

BLAHA, Michael; RUMBAUGH, James. **Modelagem e projetos baseados em objetos com UML 2**. 2. ed. Rio de Janeiro, RJ: Elsevier, 2006.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivan. **UML: guia do usuário**. Rio de Janeiro: ELSEVIER, 2000.

CANTU, Carlos H. **Conheça o Firebird em 2 minutos**. Disponível em: <http://www.firebirdnews.org/docs/fb2min_ptbr.html>. Acesso em: 25 jun. 2015.

CHAVEZ, Christina von Flach Garcia. **Um enfoque baseado em modelos para o design orientado a aspectos**. Rio de Janeiro: PUC - Rio, Departamento de Informática, 2004.

DEITEL, Harvey M. **Java como programar**. Tradução Carlos Artur Lang Lisboa. 4. ed. Porto Alegre: Bookman, 2003.

EMBARCADERO. **Delphi XE8**. Disponível em: <<http://www.embarcadero.com/br/products/delphi>>. Acesso em: 25 jun. 2015.

FOWLER, Martin. **UML essencial: um breve guia para a linguagem padrão de modelagem de objetos**, trad. João Tortello. 3. ed. Porto Alegre: Bookman, 2005.

LARMAN, Craig. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos**. 2. ed. Porto Alegre: Bookman.

LIDEL. **Diagrama de classes**. Disponível em: <http://www.lidel.pt/dco/722-636-8_des_sis_inf.pdf>. Acesso em: 20 mai. 2015.

O'DOCHERTY, Mike. **Object-oriented analysis and design: understanding system development with UML 2.0**. John Wiley & Sons Ltd., 2005.

PRECATO. **Precato seguros**. Disponível em: <<http://www.precatoseguros.com.br/site/glossario.php>>. Acesso em: 2 jun. 2015.

RAVE. **Rave Report no Delphi**. Disponível em: <<http://pt.scribd.com/doc/4941622/rave-report-no-delphi#scribd>>. Acesso em: 25 jun. 2015.

RUMBAUGH, James et al. **Modelagem e projetos baseados em objetos**. Rio de Janeiro, RJ: Campus, 1994.

SCOTT, Kendall. **O processo unificado explicado**. São Paulo : Bookman, 2002.

SENAC. <http://www.nevrona.com/>. Disponível em: <http://www.slideshare.net/CursoSENAC/anlise-orientada-a-objetos-diagrama-de-sequencia>>. Acesso em: 15 mar. 2015

SILVA, Viviane T. **Diagrama de classes**. 2010. Disponível em: <http://www2.ic.uff.br/~viviane.silva/2010.2/es1/util/aula7.pdf>>. Acesso em: 22 mai. 2015.

SUPERINTENDENCIA DE SEGUROS PRIVADOS (SUSEP). **2o Relatório de análise e acompanhamento dos mercados supervisionados**. Rio de Janeiro, 2014. Disponível em: <http://www2.susep.gov.br/menuestatistica/SES/principal.aspx>>. Acesso em: 10 abr. 2015

SUPERINTENDÊNCIA DE SEGUROS PRIVADOS (SUSEP). Disponível em: <http://www.susep.gov.br/>>. Acesso em: 10 abr. 2015.

TOAD. *Toad data modeler*. Disponível em: <http://www.casestudio.com/enu/default.aspx>>. Acesso em: 26 jun. 2015.

TSS. **Tudo sobre seguros**. Seguros compreensivos. Disponível em: <http://www.tudosobreseguros.org.br/sws/portal/pagina.php?l=217>>. Acesso em: 10 abr. 2015.

VISUAL PARADIGM. **Visual Paradigm**. Disponível em: <https://www.visual-paradigm.com/tutorials/>>. Acesso em: 26 jun. 2015.

VOLACO, Everson B. **Visão geral do IBExpert**. Disponível em: <http://www.devmedia.com.br/6-visao-geral-do-ibexpert/4828>>. Acesso em: 25 jun. 2015.