

CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA
Faculdade de Tecnologia de Mauá

Denis Damasceno dos Santos
Renan Becchelli Csapo

Desenvolvimento de um sistema para controle de apólices de seguro

Mauá
2025

Denis Damasceno dos Santos

Renan Becchelli Csapo

Desenvolvimento de um sistema para controle de apólices de seguro

Monografia apresentada à FATEC Mauá,
como parte dos requisitos para obtenção
do Título de Tecnólogo em Informática
para Negócios.

Orientador: Prof. Me. Ivan Carlos Pavão.

Mauá

2025

Autorizo a reprodução e a divulgação deste trabalho, no todo ou em parte, por qualquer meio convencional ou eletrônico, exclusivamente para fins de estudo e pesquisa, desde que a fonte seja citada.

Catálogo-na-Publicação – Biblioteca Fatec Mauá

005.117

S237d Santos, Denis Damasceno dos.

Desenvolvimento de um sistema para controle de apólices de seguro / Denis Damasceno dos Santos, Renan Becchelli Csapo. – 2025. – 47 p. : il. ; 30 cm.

Orientador: Prof. Me. Ivan Carlos Pavão.

Trabalho de conclusão de curso (Curso Superior de Tecnologia em Informática para Negócios) – Faculdade de Tecnologia de Mauá.

Inclui referências.

1. Sistemas de informação. 2. Corretora de seguros. 3. Engenharia de software. 4. Banco de dados. 5. C# (linguagem de programação). I. Csapo, Renan Becchelli. II. Pavão, Ivan Carlos. III. Título.

CDD 23. : Programação orientada a objetos 005.117
Elaborada por Tatiana Sambinelli CRB-8 SP-011003/O

Denis Damasceno dos Santos
Renan Becchelli Csapo

Desenvolvimento de um sistema para controle de apólices de seguro

Monografia apresentada à FATEC Mauá,
como parte dos requisitos para obtenção do
Título de Tecnólogo em Informática para
Negócios.

Aprovação em: 02/12/2025

Banca examinadora ou pareceristas:

Prof.(a) Me. Ivan Carlos Pavão
Orientador
FATEC Mauá

Prof.(a) Me. Luana Lourenço
Avaliador
FATEC Mauá

Prof.(a) Me. Renata Rodrigues
Avaliador
FATEC Mauá

" Dedicamos este trabalho aos nossos familiares e esposas, pelo amor, paciência e apoio incondicional em todos os momentos desta caminhada."

AGRADECIMENTOS

Agradecemos primeiramente aos nossos familiares e esposas, pelo apoio incondicional, paciência e incentivo em todos os momentos desta jornada acadêmica. A compreensão e o estímulo de cada um foram fundamentais para que pudéssemos conciliar os estudos, o trabalho e a realização deste projeto.

Estendemos nossos sinceros agradecimentos aos professores da Fatec de Mauá, que compartilharam seus conhecimentos e experiências, contribuindo significativamente para a nossa formação pessoal e profissional.

Aos colegas e amigos de curso, que estiveram presentes ao longo do caminho, pela parceria, pelas trocas de aprendizado e pelo companheirismo durante todo o período de desenvolvimento do trabalho.

Agradecemos especialmente ao nosso orientador, Professor Me. Ivan Carlos Pavão, pela orientação, dedicação e apoio durante todas as etapas deste projeto. Sua disponibilidade, conhecimento técnico e incentivo constante foram essenciais para a construção e conclusão deste Trabalho de Conclusão de Curso.

Por fim, expressamos nossa gratidão à Fatec de Mauá pela oportunidade de desenvolver esta pesquisa, pela estrutura oferecida e pelo compromisso com a formação de profissionais capacitados e comprometidos com a inovação e a qualidade

“Os computadores são incrivelmente rápidos, precisos e burros. Os seres humanos são incrivelmente lentos, imprecisos e brilhantes. Juntos, seus poderes ultrapassam os limites da imaginação.”

(Albert Einstein)

RESUMO

O presente trabalho tem como objetivo desenvolver um sistema de controle de apólices para uma corretora de seguros, visando otimizar os processos administrativos e a gestão das informações referentes a clientes, seguradoras e contratos. O estudo foi conduzido sob a perspectiva da engenharia de software, aplicando princípios de modelagem de dados e programação orientada a objetos para a construção de uma aplicação prática e funcional. A metodologia adotada caracteriza-se como pesquisa aplicada, de abordagem qualitativa e natureza exploratória e descritiva, desenvolvida com base no modelo incremental, que permitiu a implementação e validação progressiva dos módulos do sistema. O sistema foi construído utilizando a linguagem C# no ambiente Windows Forms, com integração ao banco de dados SQL Server, possibilitando o cadastro, a consulta, a alteração e o controle de vencimentos das apólices. Os resultados obtidos demonstraram que a aplicação atende aos requisitos propostos, garantindo maior confiabilidade, segurança e agilidade no gerenciamento das informações. A etapa de testes confirmou a estabilidade do sistema e sua adequação às necessidades de uma corretora de pequeno porte, contribuindo para a melhoria da organização e para a redução de falhas operacionais. Conclui-se que o sistema desenvolvido representa uma solução tecnológica eficiente e adaptável, reforçando a importância da utilização de ferramentas informatizadas na modernização dos processos empresariais.

Palavras-chave: sistemas de informação; corretora de seguros; engenharia de software; banco de dados; C#

ABSTRACT

This work aims to develop a policy management system for an insurance brokerage company, seeking to optimize administrative processes and improve the management of information related to clients, insurers, and contracts. The study was conducted from the perspective of software engineering, applying principles of data modeling and object-oriented programming to build a practical and functional application. The adopted methodology is characterized as applied research, with a qualitative, exploratory, and descriptive approach, based on the incremental model, which allowed for the progressive implementation and validation of the system's modules. The system was developed using the C# programming language in the Windows Forms environment, integrated with a SQL Server database, enabling the registration, consultation, updating, and monitoring of policy expirations. The results demonstrated that the application meets the proposed requirements, ensuring greater reliability, security, and efficiency in information management. The testing phase confirmed the stability of the system and its suitability for the operational needs of a small insurance brokerage, contributing to improved organization and reduced operational errors. It is concluded that the developed system represents an efficient and adaptable technological solution, reinforcing the importance of using computerized tools to modernize business processes.

Keywords: information systems; insurance brokerage; software engineering; database; C#.

LISTA DE SIGLAS

ABNT – Associação Brasileira de Normas Técnicas

API – Application Programming Interface (Interface de Programação de Aplicações)

BD – Banco de Dados

C# – Linguagem de Programação C Sharp

CRUD – Create, Read, Update and Delete (Criar, Ler, Atualizar e Excluir)

DER – Diagrama Entidade-Relacionamento

ER – Entidade-Relacionamento

IDE – Integrated Development Environment (Ambiente de Desenvolvimento Integrado)

MER – Modelagem Entidade-Relacionamento

POO – Programação Orientada a Objetos

SGBD – Sistema de Gerenciamento de Banco de Dados

SQL – Structured Query Language (Linguagem de Consulta Estruturada)

TI – Tecnologia da Informação

TCC – Trabalho de Conclusão de Curso

UML – Unified Modeling Language (Linguagem de Modelagem Unificada)

UI – User Interface (Interface do Usuário)

SUMÁRIO

1. INTRODUÇÃO	11
1.1 Problema de pesquisa	12
1.2 Objetivos	12
1.2.1 Objetivo Geral	13
1.2.2 Objetivos Específicos	13
1.3 Justificativa	13
1.4 Delimitação	14
1.5 Estruturação	15
2. DESENVOLVIMENTO	17
2.1 Referencial teórico	18
2.2 Procedimentos metodológicos	20
2.3 Análise de resultados	22
3. DESENVOLVIMENTO DO SISTEMA	25
3.1 Modelagem do Banco de Dados	26
3.1.1 Tabelas Principais do Sistema	27
3.1.2 Relacionamentos entre as Tabelas	27
3.1.3 Integridade e Normalização	28
3.1.4 Considerações Finais da Modelagem	29
3.2 Interface do Sistema	29
3.2.1 Tela de Login	30
3.2.2 Menu Principal	30
3.2.3 Tela de Cadastro	31
3.2.4 Tela de Consulta e Alteração	33
3.2.5 Geração de Relatórios	34
3.2.6 Padrões Visuais e Usabilidade	35
3.2.7 Considerações Finais sobre a Interface	35
3.3 Implementação das Funcionalidades	36
3.3.1 Login e Autenticação de Usuário	36
3.3.2 Cadastros e Alterações	37
3.3.3 Consultas	37
3.3.4 Relatórios	37
3.3.5 Estrutura de Conexão com o Banco de Dados	38
3.3.6 Testes e Validação	39

3.3.7 Considerações Finais da Implementação.....	39
3.4 Testes e Validação do Sistema	39
3.4.1 Tipos de Testes Aplicados	40
3.4.2 Critérios de Validação	40
3.4.3 Resultados dos Testes	41
3.4.4 Ajustes e Melhorias	41
3.4.5 Conclusão dos Testes	42
4. Conclusão	43
REFERÊNCIAS	45
GLOSSÁRIO	47

1. INTRODUÇÃO

No contexto atual, as empresas buscam constantemente soluções tecnológicas que otimizem seus processos e tornem suas operações mais ágeis e seguras. No segmento de corretagem de seguros, essa necessidade é ainda mais evidente, pois o controle de informações referentes a clientes, seguradoras, corretores, apólices e prazos de vencimento exige precisão e organização. A ausência de um sistema informatizado pode gerar falhas no controle, perda de dados e atrasos na renovação de contratos, comprometendo o relacionamento com o cliente e a credibilidade da corretora.

O gerenciamento manual de informações em planilhas ou registros físicos, ainda presente em muitas corretoras, é um método suscetível a erros humanos e limitações de acesso. Além disso, a falta de alertas automáticos sobre o vencimento de apólices pode acarretar prejuízos financeiros e perda de oportunidades de renovação. Nesse cenário, torna-se evidente a importância da informatização dos processos internos para garantir eficiência, segurança e confiabilidade das informações.

Diante dessa problemática, o presente trabalho tem como proposta o desenvolvimento de um sistema de controle de apólices e gestão de seguros, voltado para o ambiente desktop, utilizando a linguagem C# (Windows Forms) integrada ao banco de dados SQL Server. O sistema foi idealizado para permitir o cadastro, consulta, alteração e exclusão de registros de clientes, corretores, seguradoras e apólices, além da emissão de relatórios que facilitem a tomada de decisão e o acompanhamento de vencimentos.

O desenvolvimento do sistema foi motivado pela necessidade de criar uma ferramenta prática e eficiente que atenda às demandas de uma corretora de seguros de pequeno a médio porte, oferecendo uma interface intuitiva e funcionalidades que reduzam o retrabalho e o risco de inconsistências. Por meio da automação desses processos, busca-se melhorar o desempenho operacional e assegurar um controle eficaz das apólices vigentes.

Dessa forma, o trabalho propõe a criação de um Sistema de Controle de Apólices, que centraliza informações essenciais da corretora, permitindo acesso rápido, relatórios automatizados e alertas de vencimento. O sistema visa não apenas substituir métodos manuais, mas também proporcionar uma base sólida para futuras

melhorias, como a integração com plataformas web e o envio automático de notificações.

1.1 Problema de pesquisa

No cenário atual das corretoras de seguros, o volume crescente de informações relacionadas a clientes, corretores, seguradoras e apólices torna cada vez mais complexa a gestão manual desses dados. A utilização de planilhas e registros físicos, embora comum em empresas de pequeno e médio porte, apresenta limitações significativas quanto à organização, segurança e acesso às informações. Essa forma de controle dificulta o acompanhamento preciso de prazos de vencimento das apólices, podendo ocasionar atrasos na renovação de contratos, falhas operacionais e perda de oportunidades comerciais.

Diante desse contexto, surge a necessidade de investigar alternativas que possibilitem automatizar os processos de registro, consulta e monitoramento de apólices, de modo a garantir maior eficiência e confiabilidade nas operações da corretora.

Assim, formula-se a seguinte questão de pesquisa que orienta o desenvolvimento deste trabalho:

Como a implementação de um sistema informatizado, desenvolvido em C# e integrado a um banco de dados SQL Server, pode auxiliar no controle e na gestão das apólices de seguros em uma corretora, promovendo maior eficiência, organização e segurança das informações?

1.2 Objetivos

O presente trabalho tem como objetivo principal o desenvolvimento de um sistema informatizado para o controle e a gestão de apólices de seguros em uma corretora, visando à automatização de processos que, tradicionalmente, são realizados de forma manual e suscetíveis a erros. O sistema proposto foi concebido para centralizar informações essenciais, permitindo o registro, a consulta e o acompanhamento de dados de clientes, corretores, seguradoras e ramos de seguros, bem como o controle de prazos de vencimento das apólices.

A criação dessa ferramenta busca proporcionar maior agilidade nas rotinas administrativas, garantir a integridade dos dados e oferecer uma interface de fácil utilização, contribuindo para o aprimoramento da gestão interna da corretora e para a eficiência de seus processos operacionais.

1.2.1 Objetivo geral

Desenvolver um sistema desktop para o controle e a gestão de apólices de seguros, utilizando a linguagem de programação C# e o banco de dados SQL Server, de forma a otimizar o gerenciamento de informações e reduzir falhas decorrentes do controle manual.

1.2.2 Objetivos específicos

- Implementar um módulo de login com controle de acesso por usuário e senha;
- Desenvolver módulos de cadastro, consulta, alteração e exclusão de registros de clientes, corretores, seguradoras, ramos e apólices;
- Integrar o sistema a um banco de dados relacional (SQL Server) para armazenamento seguro e consistente das informações;
- Criar relatórios automatizados que permitam visualizar todas as apólices registradas e identificar aquelas com vencimento próximo (em até dez dias);
- Proporcionar uma interface gráfica intuitiva, de fácil navegação e adequada ao ambiente de trabalho da corretora;
- Assegurar segurança e integridade dos dados, evitando duplicidade de registros e inconsistências no banco de dados;
- Validar o funcionamento do sistema por meio de testes práticos que comprovem sua eficácia no ambiente simulado da corretora.

1.3 Justificativa

O desenvolvimento de sistemas informatizados voltados para o controle e a gestão de informações empresariais tem se mostrado uma necessidade crescente no

contexto organizacional contemporâneo. No setor de corretagem de seguros, essa demanda torna-se ainda mais evidente, considerando o grande volume de dados que precisam ser administrados diariamente, como cadastros de clientes, apólices, prazos de vencimento, seguradoras e ramos de atuação. A ausência de um sistema eficiente de gerenciamento pode gerar falhas de controle, retrabalho e perda de informações relevantes, impactando diretamente na qualidade do atendimento e na confiabilidade das operações.

Diante desse cenário, este trabalho justifica-se pela importância de propor uma solução tecnológica que contribua para a organização, segurança e agilidade no controle das apólices de seguros. A automatização dos processos internos por meio de um sistema desenvolvido em C#, com banco de dados SQL Server, possibilita otimizar o fluxo de trabalho da corretora e reduzir a dependência de controles manuais, que são mais suscetíveis a erros e inconsistências.

Além do aspecto técnico, o estudo também possui uma motivação pessoal e profissional. O tema foi escolhido a partir da vivência de um dos autores, Renan, que atua como proprietário de uma corretora de seguros e vivência, na prática, as dificuldades relacionadas ao controle manual de apólices e cadastros. Essa experiência direta no setor permitiu identificar as principais necessidades e desafios enfrentados pelas corretoras, servindo como base para o desenvolvimento de uma solução eficaz e aderente à realidade do mercado.

Sob a perspectiva acadêmica, o trabalho representa uma oportunidade de aplicar na prática os conhecimentos adquiridos ao longo do curso de Informática para Negócios, especialmente nas áreas de banco de dados, programação orientada a objetos, engenharia de software e gestão de processos. O projeto contribui não apenas para o aprimoramento das competências técnicas dos autores, mas também para o avanço de soluções que aproximam a tecnologia das demandas reais das empresas, promovendo eficiência, controle e tomada de decisão mais assertiva.

1.4 Delimitação

Esta pesquisa delimita-se ao desenvolvimento de um sistema desktop voltado ao controle de apólices de seguros, abrangendo o cadastro, consulta, alteração e exclusão de informações referentes a clientes, corretores, seguradoras, ramos e apólices. O sistema também contempla a geração de relatórios automatizados,

incluindo um módulo específico para identificação de apólices com vencimento próximo, em até dez dias.

O estudo foi realizado no contexto de uma corretora de seguros de pequeno porte localizada na região do Grande ABC, cuja rotina operacional serviu de base para o levantamento de requisitos e definição das funcionalidades do sistema. O projeto foi desenvolvido utilizando a linguagem de programação C# (Windows Forms) e o banco de dados SQL Server, de modo a oferecer uma solução prática e acessível para empresas do mesmo porte e segmento.

A pesquisa concentra-se nos aspectos técnicos e funcionais do sistema, não abrangendo a integração com plataformas web, aplicações móveis ou recursos avançados de inteligência artificial. O foco está na automação das atividades internas da corretora, especialmente no gerenciamento de cadastros e controle de vencimentos, visando demonstrar como a implementação de um sistema informatizado pode contribuir para a eficiência e confiabilidade dos processos administrativos.

1.5 Estruturação

Este trabalho está organizado em quatro capítulos, que apresentam de forma sequencial e lógica o desenvolvimento do estudo, desde a contextualização do problema até a conclusão dos resultados obtidos.

No Capítulo 1, apresenta-se a introdução do trabalho, contextualizando o tema e a relevância da pesquisa. São expostos o problema de pesquisa, os objetivos gerais e específicos, a justificativa, a delimitação do tema e a estrutura do trabalho, de modo a situar o leitor quanto ao propósito e à abrangência do estudo.

O Capítulo 2 aborda a fundamentação teórica, reunindo conceitos e referências que sustentam o desenvolvimento do sistema. São apresentados temas relacionados a sistemas de informação, engenharia de software, linguagem de programação C#, banco de dados SQL Server e metodologias de desenvolvimento de sistemas. Essa base teórica fornece o embasamento necessário para a compreensão técnica e científica do projeto.

O Capítulo 3 apresenta o desenvolvimento do sistema, detalhando as etapas de planejamento, análise de requisitos. Modelagem, implementação, testes e validação. Apresentamos as telas criadas no sistema, as funcionalidades de login, o

controle de apólices e a utilização da biblioteca iTextSharp, responsável pela geração de relatórios de formato PDF.

Por fim, o Capítulo 4 reúne as considerações finais, nas quais são analisados os resultados alcançados, as contribuições do sistema para a gestão de corretoras de seguros e as possíveis melhorias e expansões futuras, como o desenvolvimento de versões web ou mobile.

2. DESENVOLVIMENTO

O presente capítulo tem como objetivo apresentar o desenvolvimento do projeto proposto, descrevendo o processo de concepção, planejamento e implementação do sistema de controle de apólices para uma corretora de seguros. Nesta etapa, são abordados os fundamentos teóricos e técnicos que nortearam o desenvolvimento da aplicação, bem como as metodologias e tecnologias utilizadas na sua construção.

Diferentemente da introdução, que apenas contextualiza o tema e os objetivos da pesquisa, o desenvolvimento representa o núcleo do trabalho, onde são aprofundadas as discussões e demonstradas as etapas que compuseram a realização do projeto. Essa seção busca estabelecer a relação entre o embasamento teórico e a aplicação prática, evidenciando como o conhecimento adquirido ao longo do curso foi integrado à construção da solução proposta.

O desenvolvimento do sistema baseou-se nos princípios da engenharia de software, que envolvem a análise de requisitos, a modelagem de dados, o design da interface e a implementação do código-fonte. O processo foi conduzido de forma incremental, permitindo que cada módulo — como login, cadastros, relatórios e controle de vencimentos — fosse desenvolvido, testado e aprimorado de maneira progressiva. Essa abordagem favoreceu a organização das etapas e a validação das funcionalidades em tempo real, assegurando maior confiabilidade ao produto final.

A aplicação foi desenvolvida utilizando a linguagem de programação C#, em ambiente Windows Forms, e o banco de dados SQL Server como plataforma de armazenamento das informações. Essa escolha se justifica pela ampla utilização dessas tecnologias no mercado corporativo e pela facilidade de integração entre a aplicação e o banco relacional, o que proporciona maior segurança e desempenho na manipulação dos dados.

O sistema foi projetado para atender às demandas de uma corretora de seguros de pequeno porte, permitindo o gerenciamento completo das informações relacionadas a clientes, corretores, seguradoras, ramos e apólices. Além disso, inclui funcionalidades específicas para relatórios gerenciais e alertas automáticos de vencimentos, oferecendo à empresa uma visão clara sobre suas operações e auxiliando na tomada de decisões estratégicas.

Nos tópicos seguintes, serão detalhadas as etapas do desenvolvimento do projeto, abordando a análise de requisitos, a modelagem do banco de dados, o design

das telas, a implementação das funcionalidades e os testes realizados. Serão também apresentadas as ferramentas e metodologias empregadas, bem como os fundamentos técnicos que sustentam a construção do sistema.

2.1 Referencial teórico

O referencial teórico é o alicerce conceitual que sustenta o desenvolvimento deste trabalho, fornecendo base científica e técnica para a análise e implementação do sistema de controle de apólices desenvolvido para uma corretora de seguros. De acordo com Severino (2016, p. 185), “o que não se pode admitir em hipótese alguma é a transcrição literal de uma passagem de outro autor sem fazer a devida referência”, sendo, portanto, imprescindível que todas as ideias apresentadas estejam devidamente fundamentadas nas obras consultadas.

A área de engenharia de software constitui o principal campo teórico deste estudo. Segundo Pressman (2011, p. 17), “*engenharia de software é a aplicação de uma abordagem sistemática, disciplinada e quantificável para o desenvolvimento, operação e manutenção de software*”. Nessa perspectiva, o processo de desenvolvimento não deve ocorrer de forma improvisada, mas sim baseado em métodos estruturados que garantam qualidade, confiabilidade e facilidade de manutenção ao produto final.

Sommerville (2007) reforça essa visão ao afirmar que a engenharia de software envolve não apenas a criação de programas, mas todo o conjunto de atividades que asseguram que o software atenda às necessidades dos usuários, dentro de padrões de qualidade e segurança. Para o autor, a aplicação de boas práticas e a utilização de modelos de desenvolvimento são fundamentais para reduzir riscos e assegurar a entrega de sistemas eficazes.

No contexto organizacional, a automação de processos é um fator decisivo para a competitividade e a sustentabilidade das empresas. Laudon e Laudon (2014) definem os sistemas de informação como um conjunto de componentes inter-relacionados que coletam, processam, armazenam e distribuem informações para apoiar a tomada de decisão, a coordenação e o controle nas organizações. Essa definição está alinhada à proposta deste trabalho, cujo sistema visa centralizar e gerenciar as informações de uma corretora de seguros, reduzindo falhas e retrabalhos associados a controles manuais.

A modelagem de sistemas é outro elemento central no desenvolvimento de software, pois fornece a estrutura lógica para o planejamento e a organização do sistema. Conforme Fowler (2005), a utilização da UML (Unified Modeling Language) permite representar graficamente os componentes e as relações entre os elementos do sistema, facilitando o entendimento e a comunicação entre os membros da equipe de desenvolvimento. A UML inclui diagramas de casos de uso, classes, sequência e atividades, sendo amplamente empregada em projetos orientados a objetos.

Complementando, o *Object Management Group (OMG, 2017)*, responsável pela padronização da UML, descreve que essa linguagem é essencial para promover clareza na documentação e reduzir ambiguidades no processo de desenvolvimento. Dessa forma, sua aplicação contribui para a consistência do sistema e para a manutenção de um padrão de qualidade técnica.

Para a modelagem de dados, utilizou-se o modelo entidade-relacionamento (MER), que, segundo a DevMedia (2010), é “a forma mais comum de representar graficamente os dados de um sistema e os relacionamentos existentes entre eles”. A definição das entidades, atributos e relacionamentos foi fundamental para a construção do banco de dados do sistema, garantindo integridade e coerência nas informações manipuladas.

O banco de dados relacional escolhido foi o SQL Server, ferramenta desenvolvida pela Microsoft que oferece alto desempenho e segurança. A própria documentação oficial da Microsoft (2025) ressalta que o SQL Server é amplamente utilizado em sistemas corporativos pela sua robustez, escalabilidade e integração com outras tecnologias da empresa, como o .NET e o Visual Studio.

No desenvolvimento da aplicação, foi utilizada a linguagem C#, também da Microsoft, voltada à programação orientada a objetos e à criação de aplicações para o ambiente Windows. De acordo com a documentação do *Microsoft Learn* (2025), o C# é uma linguagem moderna, fortemente tipada e amplamente utilizada no desenvolvimento de aplicações empresariais. Essa escolha se deve à sua sintaxe clara, facilidade de integração com bancos de dados e suporte nativo ao paradigma orientado a objetos, características que se mostraram adequadas às necessidades deste projeto.

No que se refere às metodologias de desenvolvimento, o projeto adotou princípios das metodologias ágeis, especialmente o Scrum, que tem como propósito promover entregas contínuas e iterativas. Conforme Schwaber e Sutherland (2020, p.

5), criadores do Scrum, “o *framework Scrum é leve, simples de entender, mas difícil de dominar*”, sendo ideal para equipes pequenas que necessitam de flexibilidade e adaptação contínua às mudanças de requisitos. A abordagem incremental, utilizada neste trabalho, segue esses princípios, permitindo o desenvolvimento do sistema por etapas, com revisões constantes e aprimoramentos sucessivos.

A importância da engenharia de software, aliada à aplicação de metodologias ágeis e ao uso de tecnologias de programação modernas, constitui a base conceitual que orientou todas as decisões de desenvolvimento neste projeto. Assim, o referencial teórico aqui apresentado não apenas fundamenta as escolhas técnicas, como também demonstra a coerência entre as práticas de desenvolvimento adotadas e as recomendações da literatura especializada.

2.2 Procedimentos metodológicos

A metodologia adotada neste trabalho teve como propósito descrever, de maneira sistemática e detalhada, os procedimentos utilizados no desenvolvimento do sistema de controle de apólices para uma corretora de seguros. Os métodos empregados foram definidos de forma a garantir a coerência entre os objetivos propostos, a viabilidade técnica e a aplicação prática dos resultados obtidos.

O presente estudo caracteriza-se como uma pesquisa aplicada, uma vez que busca a solução de um problema real, identificado no contexto de uma corretora de seguros, por meio do desenvolvimento de um produto tecnológico — o sistema informatizado. De acordo com Gil (2019, p. 27), “a pesquisa aplicada tem como objetivo gerar conhecimentos para aplicação prática, dirigidos à solução de problemas específicos”. Dessa forma, o trabalho não se limita à análise teórica, mas aplica o conhecimento técnico e científico para criar uma ferramenta funcional.

Quanto à abordagem, trata-se de uma pesquisa qualitativa, pois o estudo não se baseia em dados numéricos ou análises estatísticas, mas na compreensão de processos e na observação direta do funcionamento do sistema desenvolvido. Segundo Godoy (1995, p. 58), “a pesquisa qualitativa preocupa-se com o significado, com a forma como as pessoas interpretam e dão sentido aos fenômenos”. No caso deste trabalho, o foco está no entendimento e na otimização dos processos internos da corretora de seguros.

Em relação aos objetivos, o trabalho tem natureza exploratória e descritiva. É exploratória por buscar compreender e mapear as necessidades do ambiente de negócio da corretora, e descritiva por apresentar as etapas de desenvolvimento e implementação do sistema, detalhando suas funcionalidades, estrutura de dados e recursos utilizados. Essa combinação permite identificar as demandas da organização e descrevê-las de forma técnica e objetiva.

As fontes de informação utilizadas foram predominantemente bibliográficas e documentais, por meio de consulta a livros, artigos científicos, publicações técnicas e sites especializados nas áreas de engenharia de software, banco de dados e metodologias de desenvolvimento ágil. Também foram consideradas observações empíricas sobre o funcionamento de uma corretora de seguros real, o que proporcionou uma visão prática sobre as rotinas administrativas que o sistema deveria atender.

O desenvolvimento do sistema foi estruturado com base na metodologia incremental, que consiste na construção do software em partes ou módulos funcionais, permitindo a validação progressiva de cada etapa. Segundo Pressman (2011), o modelo incremental favorece a entrega contínua de versões funcionais do sistema, possibilitando ajustes e aprimoramentos conforme as necessidades identificadas ao longo do processo de implementação.

Durante o desenvolvimento, foram utilizadas as seguintes ferramentas e tecnologias:

- Microsoft Visual Studio 2022, como ambiente de desenvolvimento integrado (IDE);
- Linguagem C#, para construção da aplicação desktop;
- Banco de dados SQL Server, para armazenamento e gerenciamento das informações;
- Microsoft SQL Server Management Studio (SSMS), para administração das tabelas e consultas SQL;
- Lucidchart, para elaboração dos diagramas UML e modelo entidade-relacionamento;
- Windows Forms, para o design das interfaces gráficas.

As etapas metodológicas seguiram o seguinte fluxo:

1. **Levantamento de requisitos**, com base na observação das rotinas da corretora e na identificação das funcionalidades essenciais ao sistema;
2. **Modelagem de dados**, com a criação do modelo entidade-relacionamento (MER) e definição das tabelas do banco de dados;
3. **Modelagem funcional**, utilizando diagramas de caso de uso e classes para representar a estrutura lógica do sistema;
4. **Implementação**, com o desenvolvimento dos módulos de login, cadastros, consultas, alterações e relatórios;
5. **Testes e validação**, para assegurar o funcionamento correto das funcionalidades e a integridade dos dados;
6. **Ajustes e refinamentos**, com melhorias na interface e na eficiência das consultas ao banco de dados.

O produto final obtido — o sistema de controle de apólices — representa a materialização do conhecimento técnico adquirido ao longo do curso de Informática para Negócios, aplicando princípios de engenharia de software, banco de dados relacional e programação orientada a objetos. Sua contribuição prática consiste em oferecer uma ferramenta de apoio à gestão de corretoras de seguros, capaz de otimizar processos, reduzir falhas e proporcionar maior controle das informações administrativas.

Assim, os procedimentos metodológicos adotados neste trabalho garantiram não apenas o rigor técnico no desenvolvimento do sistema, mas também a aplicabilidade dos resultados no contexto real de uma empresa do setor de seguros.

2.3 Análise de resultados

A análise dos resultados tem como objetivo examinar de forma crítica os efeitos do desenvolvimento do sistema proposto, buscando compreender em que medida as etapas metodológicas adotadas contribuíram para o alcance dos objetivos estabelecidos. Nessa perspectiva, os resultados obtidos são interpretados à luz do referencial teórico apresentado, permitindo verificar a coerência entre o conhecimento científico aplicado e a solução tecnológica desenvolvida.

Durante a execução do projeto, foi possível observar que a aplicação dos conceitos de engenharia de software e modelagem de sistemas possibilitou a

construção de uma aplicação estruturada, modular e de fácil manutenção. De acordo com Pressman (2011), o uso de métodos sistemáticos no desenvolvimento de software garante maior previsibilidade e controle sobre o produto final. Essa premissa foi confirmada no presente trabalho, uma vez que o modelo incremental adotado permitiu o acompanhamento contínuo das entregas e a realização de ajustes em tempo oportuno.

A análise também evidenciou que a utilização da abordagem qualitativa favoreceu a compreensão das necessidades reais do ambiente corporativo. Conforme destaca Godoy (1995), a pesquisa qualitativa busca interpretar fenômenos em seus contextos específicos, o que, neste caso, foi essencial para adaptar o sistema às particularidades do setor de corretagem de seguros. O contato direto com os processos da corretora permitiu identificar as dificuldades enfrentadas no controle de informações e direcionar as funcionalidades da aplicação para suprir essas demandas.

Sob a ótica da engenharia de software, os resultados demonstram que o planejamento e a organização das etapas de desenvolvimento, conforme proposto por Sommerville (2007), foram determinantes para garantir a qualidade técnica do sistema. A divisão do projeto em módulos — como login, cadastro de clientes, registro de apólices, relatórios e controle de vencimentos — favoreceu a clareza na codificação e facilitou os testes individuais de cada componente, aumentando a confiabilidade do produto final.

Em termos de gestão da informação, observou-se que a integração entre a aplicação em C# e o banco de dados SQL Server proporcionou maior eficiência na manipulação e recuperação dos dados. Essa integração vai ao encontro do que afirmam Laudon e Laudon (2014), ao destacarem que sistemas de informação bem projetados são capazes de transformar dados brutos em informações úteis para a tomada de decisão organizacional. No caso da corretora analisada, o sistema contribuiu para reduzir o tempo gasto com consultas manuais, minimizar erros de digitação e facilitar o acesso às informações das apólices ativas.

A adoção da metodologia incremental revelou-se eficaz para o contexto estudado, pois permitiu que o sistema fosse desenvolvido de maneira evolutiva, com a validação de cada etapa antes da implementação seguinte. Essa prática está alinhada ao que descreve Pressman (2011), ao afirmar que o modelo incremental

permite a entrega de versões funcionais do software em intervalos curtos, o que facilita o feedback contínuo e a correção de eventuais falhas.

Os resultados obtidos, portanto, demonstram que o uso de metodologias estruturadas, associado ao embasamento técnico e científico adequado, gera impactos positivos na eficiência dos processos administrativos e na confiabilidade das informações. O sistema desenvolvido mostrou-se capaz de atender às demandas de controle e gestão de apólices, oferecendo à empresa um instrumento de apoio à decisão e de melhoria operacional.

Dessa forma, a análise de resultados confirma a efetividade das estratégias adotadas durante o desenvolvimento do projeto, reforçando a importância da aplicação prática dos conhecimentos adquiridos ao longo da formação acadêmica. Além disso, evidencia-se que o produto final atende aos objetivos do estudo, contribuindo para o aperfeiçoamento dos processos internos e para a modernização das práticas administrativas no setor de seguros.

3. DESENVOLVIMENTO DO SISTEMA

Este capítulo apresenta o processo de desenvolvimento do sistema de controle de apólices para a corretora de seguros, desde as etapas iniciais de modelagem até a implementação final da aplicação. O objetivo é demonstrar de forma prática como os conceitos de engenharia de software e banco de dados, abordados no referencial teórico, foram aplicados para a criação de uma solução tecnológica funcional e alinhada às necessidades do negócio.

O sistema foi desenvolvido na linguagem C#, utilizando a plataforma Windows Forms do Microsoft Visual Studio 2022, e teve como base o banco de dados SQL Server para armazenamento e manipulação das informações. Essa escolha se justifica pela compatibilidade entre as ferramentas, pela estabilidade das tecnologias Microsoft no ambiente corporativo e pela facilidade de integração entre a aplicação e o banco de dados relacional.

A proposta central do sistema é otimizar o controle e a gestão de informações relacionadas às apólices de seguros, clientes e corretores, garantindo segurança, integridade e acessibilidade dos dados. Além disso, a aplicação busca oferecer recursos que auxiliem no acompanhamento dos vencimentos e na geração de relatórios, contribuindo para a organização e o planejamento das atividades da corretora.

O processo de desenvolvimento foi conduzido segundo a metodologia incremental, conforme proposto por Pressman (2011), que recomenda a construção do software em módulos independentes, validados progressivamente. Essa abordagem permitiu que as funcionalidades fossem implementadas e testadas de forma contínua, assegurando a estabilidade do sistema e a correção de possíveis falhas ao longo do projeto.

As principais etapas do desenvolvimento foram estruturadas da seguinte forma:

- Análise de requisitos: levantamento das necessidades do sistema, com base nas rotinas operacionais de uma corretora de seguros;
- Modelagem do banco de dados: definição das tabelas, relacionamentos e chaves primárias/estrangeiras, de acordo com o modelo entidade-relacionamento;
- Design da interface: criação das telas de navegação, login, cadastros, consultas e relatórios, com foco na usabilidade e clareza;

- Implementação: codificação das funcionalidades utilizando C# e integração com o banco de dados SQL Server;
- Testes e ajustes: verificação de desempenho, segurança e consistência dos dados;
- Validação final: análise do funcionamento geral do sistema e sua aderência aos objetivos propostos.
- Nos tópicos seguintes, são descritas detalhadamente as etapas de construção da aplicação, abrangendo a modelagem de dados, o desenvolvimento das interfaces gráficas, os principais trechos de código e as funcionalidades implementadas para cada módulo do sistema.

3.1 Modelagem do banco de dados

A modelagem do banco de dados constitui uma das etapas mais importantes do desenvolvimento do sistema, pois é responsável pela estruturação lógica das informações que serão armazenadas, manipuladas e consultadas pela aplicação. De acordo com Pressman (2011), a fase de modelagem de dados tem o papel de traduzir os requisitos do sistema em um modelo conceitual que possa ser implementado de forma eficiente e consistente em um ambiente relacional.

O processo de modelagem foi conduzido utilizando o modelo entidade-relacionamento (MER), que, segundo a DevMedia (2010), é a forma mais adequada de representar graficamente os dados de um sistema, evidenciando as entidades, atributos e relacionamentos existentes. O modelo criado para a corretora de seguros foi projetado de modo a garantir integridade referencial, evitar redundância e permitir consultas rápidas e seguras.

O banco de dados recebeu o nome de CorretoraDB e foi desenvolvido no Microsoft SQL Server, utilizando o SQL Server Management Studio (SSMS) como ferramenta de gerenciamento. A estrutura foi organizada em tabelas relacionadas entre si, conforme descrito a seguir.

3.1.1 Tabelas principais do sistema

Tblfuncionarios

Responsável por armazenar as informações dos funcionários que acessam o sistema, controlando login e senha.

Essa tabela é essencial para a segurança do sistema, pois controla o acesso dos usuários e permite validar as credenciais digitadas na tela de login.

tblClientes

Armazena os dados cadastrais dos clientes atendidos pela corretora.

Essa tabela é uma das centrais do sistema, pois se relaciona diretamente com as apólices e garante o histórico de cada cliente.

tblSeguradoras

Contém as informações das empresas seguradoras parceiras.

Essa estrutura permite o controle das seguradoras associadas, otimizando a consulta e vinculação das apólices a cada uma delas.

tblSeguros

Centraliza os dados referentes às apólices de seguros contratadas pelos clientes.

Essa tabela é o núcleo da aplicação, permitindo o controle de vigência, valores e vencimentos das apólices, além de possibilitar consultas e relatórios gerenciais.

tblRamos

Registra os tipos de seguros oferecidos (automóvel, vida, residencial, empresarial etc.).

Essa tabela pode ser relacionada com tblSeguros através do campo codRamo, padronizando os tipos de seguros cadastrados.

3.1.2 Relacionamentos entre as tabelas

O modelo entidade-relacionamento foi elaborado de forma a representar os vínculos lógicos entre as tabelas, conforme os relacionamentos abaixo:

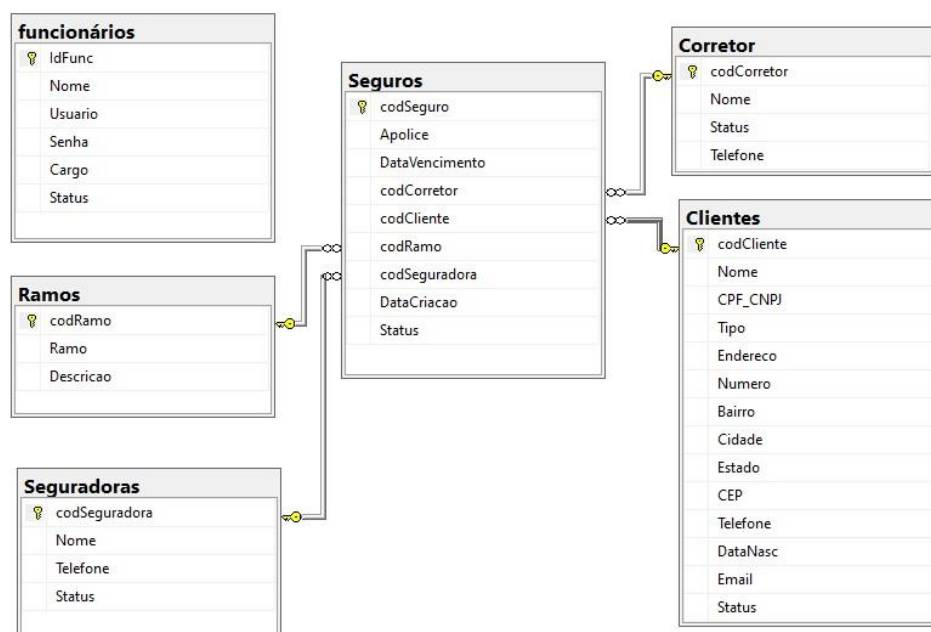
Um funcionário (tblfuncionarios) pode cadastrar múltiplos clientes e apólices, estabelecendo uma relação de um-para-muitos;

Cada cliente (tblClientes) pode possuir uma ou mais apólices (um-para-muitos);

Cada seguradora (tblSeguradoras) pode estar associada a diversas apólices (um-para-muitos);

A tblSeguros atua como tabela de junção, conectando clientes e seguradoras, e centralizando as informações contratuais.

Figura 1 – Diagrama de relacionamentos Banco de Dados



Fonte: Autores do Trabalho

3.1.3 Integridade e normalização

A estrutura do banco foi projetada seguindo os princípios da normalização de dados, conforme orientações de Sommerville (2007), garantindo que cada informação seja armazenada em apenas um local, evitando redundância e assegurando integridade referencial. As chaves primárias e estrangeiras foram devidamente definidas para manter a consistência das relações entre as tabelas.

Além disso, foram aplicadas restrições de integridade (constraints) para impedir a inserção de dados inválidos, como apólices sem cliente vinculado ou seguradora inexistente. O banco também foi configurado para suportar índices que otimizam as

consultas realizadas pelo sistema, especialmente nas telas de busca e geração de relatórios.

3.1.4 Considerações finais da modelagem

A modelagem do banco de dados da aplicação CorretoraDB foi desenvolvida com foco em simplicidade, integridade e escalabilidade, permitindo futuras expansões, como a inclusão de novas tabelas ou funcionalidades sem comprometer a estrutura existente. Essa etapa assegurou que a base de dados servisse de suporte confiável para todas as operações do sistema, desde o cadastro inicial até a emissão de relatórios e o controle de vencimentos de apólices.

3.2 Interface do sistema

A interface do sistema representa a camada de interação entre o usuário e as funcionalidades desenvolvidas. De acordo com Sommerville (2007), uma interface bem projetada deve priorizar a usabilidade, a clareza das informações e a eficiência na execução das tarefas. Nesse sentido, as telas do sistema CorretoraTCC foram elaboradas no ambiente Windows Forms do Microsoft Visual Studio 2022, buscando equilíbrio entre praticidade e estética, de forma a proporcionar uma navegação intuitiva e objetiva.

Durante o processo de desenvolvimento, as decisões de design foram orientadas pela necessidade de facilitar o uso por parte dos funcionários da corretora, que muitas vezes não possuem formação técnica em informática. Dessa forma, as telas foram organizadas de modo funcional, com menus bem estruturados, campos de preenchimento direto e botões claramente identificados por rótulos e ícones.

As interfaces foram agrupadas conforme suas finalidades, abrangendo as seguintes categorias: login e autenticação, menu principal, cadastros, consultas, alterações e relatórios. A seguir, cada uma dessas interfaces é descrita, acompanhada das figuras correspondentes que ilustram sua aparência e funcionamento.

3.2.1 Tela de login

A primeira tela exibida ao iniciar o sistema é a Tela de Login, cuja função é validar o acesso do usuário mediante a verificação das credenciais cadastradas na tabela `tblfuncionarios`. Essa etapa garante a segurança da aplicação, impedindo o uso indevido por pessoas não autorizadas.

O formulário contém dois campos de entrada — Usuário e Senha — e dois botões principais: ENTRAR (`btentrar`) e CANCELAR (`btcancelar`). Quando o usuário insere as informações e confirma, o sistema executa uma consulta ao banco de dados para validar o acesso, abrindo o formulário Menu ou retorna a mensagem “Acesso negado”.

Figura 2 – Tela de Login do Sistema

Fonte: Autores do Trabalho

3.2.2 Menu principal

Após a autenticação, o sistema direciona o usuário para o Menu Principal, construído com o componente `MenuStrip`. Ele funciona como o ponto central de navegação, organizando as principais funcionalidades da aplicação.

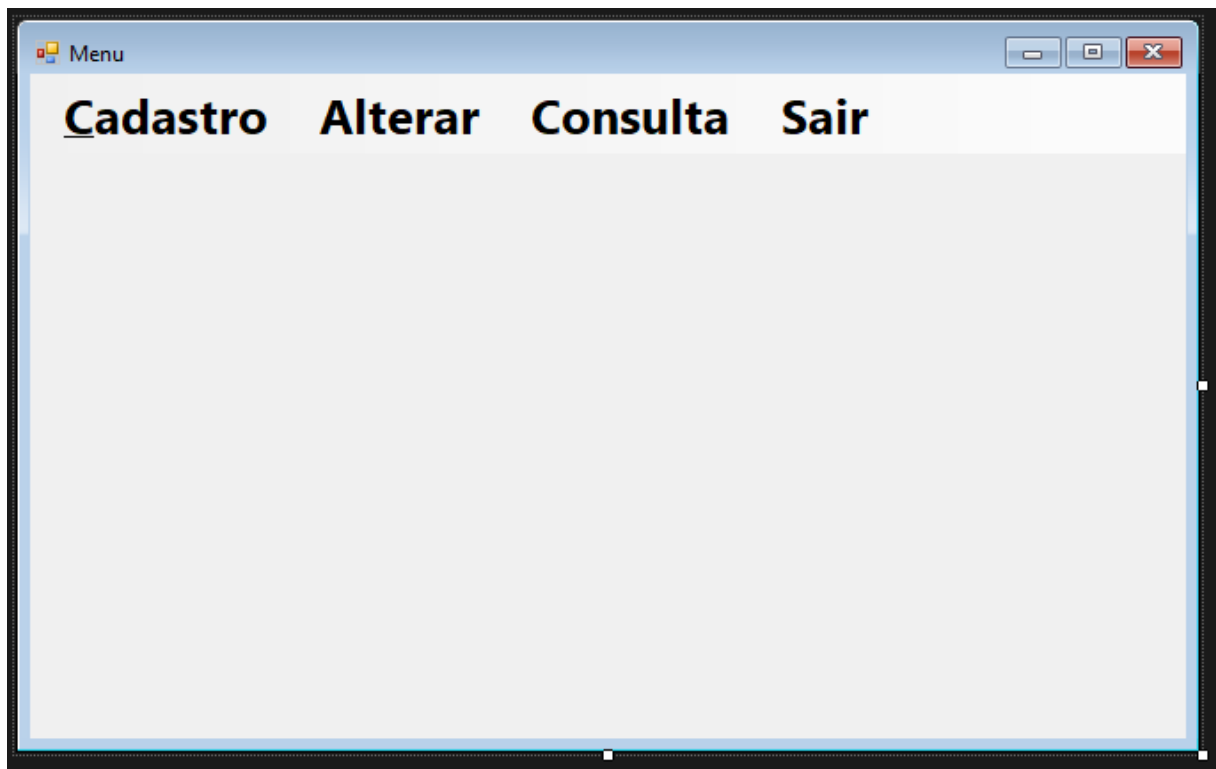
O menu contém as seguintes opções:

- **Cadastro** → acesso aos formulários de funcionários, clientes, seguradoras e apólices;
- **Alterar** → edição de registros já cadastrados;

- **Consultar** → busca de informações específicas no banco de dados;
- **Sair** → encerramento da sessão e retorno à tela de login.

A disposição dos itens foi projetada para simplificar a navegação, permitindo que o usuário acesse rapidamente as rotinas mais utilizadas.

Figura 3 – Menu Principal do Sistema



Fonte: Autores do Trabalho

3.2.3 Tela de cadastro

As telas de cadastro têm como objetivo registrar novas informações no banco de dados. Foram criadas interfaces específicas para o cadastro de funcionários, clientes, seguradoras, ramos e seguros, cada uma delas com campos próprios e validação de dados obrigatórios.

Os formulários seguem o mesmo padrão visual, contendo label, textbox, combobox e botões como Gravar, Limpar, e Voltar. Essa padronização facilita o uso e mantém a coerência visual entre os módulos.

Figura 4 – Tela de Cadastro de Clientes

The screenshot shows a Windows-style window titled 'CadClientes'. Inside, the form is titled 'Cadastro Cliente'. It contains the following fields and controls:

- Nome / Razão Social:** A text input field.
- CPF / CNPJ:** A text input field.
- Endereço:** A text input field.
- Numero:** A text input field.
- Cidade:** A text input field.
- CEP:** A text input field.
- Bairro:** A text input field.
- Estado:** A text input field.
- RG:** A text input field.
- Data de Nascimento:** A date picker showing 'sábado , 8 de novembro de 20' with a dropdown arrow.
- Telefone:** A text input field.
- E-mail:** A text input field.
- Buttons:** Three buttons at the bottom: 'Gravar', 'Limpar', and 'Voltar'.

Fonte: Autores do Trabalho

Figura 5 – Tela de Cadastro de Seguros

The screenshot shows a Windows-style window titled 'CadSeguros'. Inside, the form is titled 'Cadastro Seguros'. It contains the following fields and controls:

- Nº Apólice:** A text input field.
- Data de Vencimento:** A date picker showing '08/11/2025' with a calendar icon and a dropdown arrow.
- Cliente:** A dropdown menu.
- Corretor:** A dropdown menu.
- Seguradora:** A dropdown menu.
- Ramo:** A dropdown menu.
- Buttons:** Three buttons at the bottom: 'Gravar', 'Limpar', and 'Voltar'.

Fonte: Autores do Trabalho

3.2.4 Tela de consulta e alteração

As telas de consulta permitem ao usuário visualizar informações armazenadas no banco de dados e localizar registros de forma rápida por meio de filtros. Já as telas de alteração possibilitam a edição de dados existentes, garantindo a atualização das informações sem a necessidade de exclusão de registros.

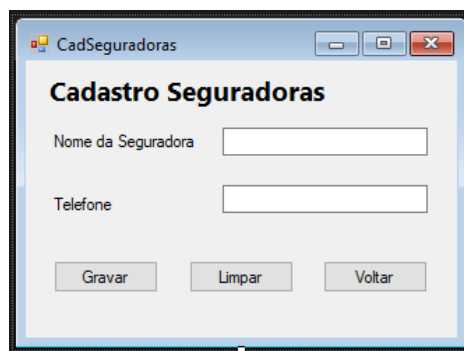
Essas telas podem utilizar componentes como DataGridView para exibir os resultados das consultas em formato tabular ou preenchimento das Textbox com os dados que retornam da pesquisa, e TextBox para pesquisa por nome, CPF/CNPJ, número da apólice ou seguradora.

Figura 6 – Tela de Consulta de Clientes

A imagem mostra uma janela de software com o título 'Form3' e uma aba 'Consulta Cliente'. O formulário contém os seguintes campos e controles:

- Consulta Por:** Um menu suspenso com uma seta para baixo.
- CPF / CNPJ:** Um campo de texto.
- Consultar:** Um botão.
- Nome / Razão Social:** Um campo de texto.
- Endereço:** Um campo de texto.
- Numero:** Um campo de texto.
- Cidade:** Um campo de texto.
- CEP:** Um campo de texto.
- Bairro:** Um campo de texto.
- Estado:** Um campo de texto.
- RG:** Um campo de texto.
- Data de Nascimento:** Um campo de texto com o valor '08/11/2025' e um ícone de calendário.
- Telefone:** Um campo de texto.
- E-mail:** Um campo de texto.
- Limpar:** Um botão.
- Voltar:** Um botão.

Fonte: Autores do Trabalho

Figura 7 – Tela de Consulta de SeguradorasA imagem mostra uma janela de software intitulada "CadSeguradoras". No topo, há uma barra de título com o ícone de uma pasta e os botões de minimizar, maximizar e fechar. O conteúdo principal da janela tem o título "Cadastro Seguradoras" em negrito. Abaixo do título, há dois campos de entrada de texto: "Nome da Seguradora" e "Telefone". Na base da janela, há três botões de ação: "Gravar", "Limpar" e "Voltar".

Fonte: Autores do Trabalho

3.2.5 Geração de relatórios

A geração de relatórios foi projetada para ser funcional, o que facilita o acesso às informações consolidadas da corretora. Nesse relatório, o usuário pode visualizar dados sobre as apólices ativas, vencimentos próximos e histórico de clientes.

Entre as opções disponíveis, há o relatório de apólices com vencimento em até 10 dias, que fornece uma visão preventiva das renovações a serem realizadas. Essa funcionalidade é essencial para a gestão eficiente das apólices, evitando atrasos e fortalecendo o relacionamento com os clientes.

A interface também conta com botões que permitem gerar e exportar relatórios, tornando o processo ágil e acessível mesmo para usuários com pouca experiência em informática. O design segue o padrão visual das demais telas do sistema, garantindo consistência e usabilidade.

Figura 8 – Relatório de Apólices com Vencimento Próximo

Relatório de Seguros						
Período: 30/10/2025 a 30/10/2026 Status: Todos						
Apólice	DataVencimen to	Cliente	Corretor	Seguradora	Ramo	Status
2250	28/08/2026 00:00:00	Giovanna da Silva Esposo	Renan Becchelli Csapo	Porto	Automovel	Ativo
221155	08/10/2026 00:00:00	Renan Becchelli Csapo	Renan Becchelli Csapo	Porto	Automovel	Ativo
254544	30/10/2026 00:00:00	Renan Becchelli Csapo	Renan Becchelli Csapo	Porto	Residencia	Ativo

Gerado em: 30/10/2025 08:57:08

Fonte: Próprios Autores do Trabalho

3.2.6 Padrões visuais e usabilidade

Todas as telas seguem uma padronização visual que inclui cores suaves, botões arredondados, tipografia legível e disposição simétrica dos componentes. O logotipo da corretora, inserido na parte superior das janelas, reforça a identidade visual do sistema.

Além disso, os elementos de interface foram posicionados com base em princípios de ergonomia e acessibilidade, conforme recomendações de Pressman (2011), garantindo que as ações principais estejam sempre visíveis e que o usuário possa realizar operações com o mínimo de cliques possível.

3.2.7 Considerações finais sobre a interface

A construção das interfaces do sistema CorretoraTCC seguiu uma abordagem centrada no usuário, buscando unir simplicidade visual, clareza funcional e eficiência operacional. O resultado é uma aplicação que atende às necessidades administrativas de uma corretora de seguros, proporcionando uma experiência de uso intuitiva e agradável.

A próxima seção abordará os aspectos técnicos e funcionais da implementação do sistema, detalhando os códigos utilizados para a integração com o banco de dados, os métodos de gravação, consulta e geração de relatórios.

3.3 Implementação das funcionalidades

A implementação do sistema CorretoraTCC foi realizada na linguagem de programação C#, utilizando a plataforma Windows Forms no ambiente Microsoft Visual Studio 2022, com integração direta ao banco de dados SQL Server. Segundo Pressman (2011), a fase de implementação é aquela em que o modelo conceitual é transformado em código executável, sendo uma das etapas mais críticas do ciclo de desenvolvimento, pois materializa a solução projetada.

O processo de implementação seguiu o modelo incremental, permitindo que cada módulo do sistema — como autenticação, cadastros, consultas e relatórios — fosse desenvolvido e testado de forma independente, garantindo maior confiabilidade e controle sobre o produto final.

A seguir, são apresentadas as principais funcionalidades desenvolvidas e seus respectivos comportamentos no sistema.

3.3.1 Login e autenticação de usuário

A tela de login tem como finalidade controlar o acesso ao sistema, verificando o usuário e a senha informados pelo funcionário. Os dados são validados com base nas informações armazenadas na tabela `tblfuncionarios` do banco de dados.

O processo de autenticação segue os seguintes passos:

O usuário informa o nome de usuário e a senha nos campos correspondentes;

O sistema cria uma conexão com o banco de dados por meio da classe `SqlConnection`;

A consulta SQL é executada utilizando `SqlDataAdapter`, retornando o resultado em um `DataTable`;

Caso o resultado contenha uma linha correspondente, o login é validado e abre o Formulário Menu;

Caso contrário, uma mensagem de “Acesso negado” é exibida.

Essa rotina garante a segurança da aplicação, restringindo o acesso apenas aos funcionários devidamente cadastrados.

3.3.2 Cadastros e alterações

Os módulos de cadastro permitem registrar e atualizar informações sobre funcionários, clientes, seguradoras e apólices.

Cada formulário possui campos validados, botões de controle (btGravar, btlimpar, btVoltar) e mensagens de confirmação.

As informações inseridas nos formulários são enviadas ao banco por meio de comandos SQL do tipo INSERT, UPDATE e DELETE.

Os cadastros seguem a mesma estrutura de lógica e conexão, alterando apenas os campos e tabelas envolvidas. Isso garante padronização e simplicidade na manutenção do código.

3.3.3 Consultas

A funcionalidade de consulta foi implementada por meio do componente DataGridView, que exibe os registros cadastrados de forma tabular.

Somente o formulário ConsClientes apresenta os resultados em Textbox onde os dados são exibidos nos respectivos campos.

O usuário pode realizar filtros por nome, CPF_CNPJ, status, seguradora ou a opção Todos conforme a tabela desejada.

Essa abordagem permite que as consultas sejam executadas de maneira dinâmica e segura, evitando injeções de SQL e erros de entrada de dados.

3.3.4 Relatórios

A funcionalidade de geração de relatórios foi implementada para automatizar o processo de consolidação e exportação de dados apresentados na interface. O sistema utiliza uma consulta SQL baseada no campo *dataVencimento* da tabela tblRamos, retornando todas as apólices com vencimento previsto nos próximos 10 dias.

Para permitir que essas informações fossem exportadas de maneira padronizada e profissional, foi incorporada à biblioteca iTextSharp, responsável pela criação de relatórios em formato PDF diretamente a partir dos dados do sistema. Essa biblioteca, amplamente utilizada em projetos desenvolvidos em C#, oferece recursos

para a construção de documentos personalizados, com controle total sobre estrutura, formatação e layout.

Durante a implementação, a classe `iTextFont` foi empregada para definir o estilo tipográfico utilizado no relatório, assegurando legibilidade, compatibilidade com caracteres acentuados e uma apresentação visual coerente com o padrão adotado pela corretora.

Com isso, o sistema passou a oferecer relatórios em formato digital, que podem ser salvos, compartilhados ou impressos conforme a necessidade. A automação dessa funcionalidade contribuiu para a agilidade operacional, redução do uso de papel e melhoria no controle interno das informações, reforçando o papel do sistema como ferramenta de apoio à gestão da corretora.

3.3.5 Estrutura de conexão com o banco de dados

Para centralizar as operações de comunicação com o banco, foi criada uma classe de conexão, que é reutilizada em todo o projeto.

Essa classe utiliza o método `SqlConnection` e implementa boas práticas de reuso e segurança, simplificando a manutenção futura.

```
private SqlConnection conexao;  
private SqlDataAdapter adapter;  
private DataTable tblClientes;  
private SqlCommand comando;  
private string strsql, strconex;  
  
string strconex = "Data Source=RENAN\\SQLEXPRESS;Initial  
Catalog=CorretoraDB;Integrated Security=SSPI";
```

Essa estrutura modulariza o código, facilitando o uso da conexão em diferentes formulários do sistema.

3.3.6 Testes e validação

Após a implementação de cada módulo, foram realizados testes unitários e funcionais para verificar a integridade dos dados e o correto funcionamento das operações.

Os testes incluíram:

- Tentativas de login com dados incorretos;
- Inserção e exclusão de registros duplicados;
- Consultas com diferentes filtros;
- Geração de relatórios com base em períodos de tempo.

De acordo com Sommerville (2007), a etapa de testes é essencial para assegurar a confiabilidade e a qualidade do software, garantindo que ele atenda aos requisitos definidos durante a análise.

3.3.7 Considerações finais da implementação

A implementação do sistema CorretoraTCC evidenciou a integração entre a teoria e a prática, aplicando conceitos de engenharia de software, modelagem de dados e programação orientada a objetos.

O resultado foi uma aplicação funcional, segura e escalável, capaz de apoiar as operações administrativas de uma corretora de seguros e otimizar o controle das informações de seus clientes e apólices.

3.4 Testes e validação do sistema

Após a fase de implementação, foi realizada a etapa de testes e validação, com o objetivo de garantir o correto funcionamento das funcionalidades do sistema CorretoraTCC e a confiabilidade das informações armazenadas no banco de dados. De acordo com Sommerville (2007), a atividade de testes é essencial para detectar erros e assegurar que o software desenvolvido atenda aos requisitos definidos na etapa de análise. Pressman (2011) complementa que os testes não devem apenas confirmar o funcionamento esperado, mas também identificar situações que possam comprometer a estabilidade e a segurança da aplicação.

O processo de testes foi conduzido de maneira incremental, acompanhando o ciclo de desenvolvimento do projeto. Cada módulo — login, cadastros e consultas —

foi testado individualmente antes da integração final, o que possibilitou detectar e corrigir falhas de forma mais rápida e controlada.

3.4.1 Tipos de testes aplicados

Os testes realizados podem ser classificados em três categorias principais:

Testes Unitários: Foram aplicados sobre blocos específicos de código, como funções de conexão, comandos SQL e métodos de gravação e leitura de dados. Esses testes permitiram validar o comportamento de cada rotina isoladamente, garantindo que os módulos funcionassem conforme o esperado.

Testes Funcionais: Nessa etapa, avaliou-se o comportamento do sistema sob a perspectiva do usuário final. Cada funcionalidade foi testada considerando as possíveis entradas, saídas e respostas do sistema.

Foram verificados, por exemplo:

- Login com dados válidos e inválidos;

- Inserção, alteração e exclusão de registros;

- Consultas com filtros por cliente, seguradora e número de apólice;

- Geração de relatórios e cálculos de vencimentos.

Testes de Integração:

Após a verificação individual dos módulos, realizou-se a integração entre as partes do sistema, assegurando que as telas, o banco de dados e as funcionalidades interagissem corretamente

Esse tipo de teste foi essencial para validar o fluxo completo de uso — desde o login até a emissão de relatórios.

3.4.2 Critérios de validação

A validação teve como propósito confirmar se o sistema atendia aos requisitos levantados durante a análise. Segundo Gil (2019), a validação consiste em verificar se o produto obtido está em conformidade com os objetivos propostos pela pesquisa, garantindo a aplicabilidade e a relevância dos resultados.

Foram adotados os seguintes critérios para validação:

Usabilidade: o sistema deve ser de fácil compreensão e utilização pelos funcionários da corretora, mesmo sem conhecimento técnico avançado;

Confiabilidade: as informações registradas devem ser armazenadas e recuperadas sem erros;

Segurança: apenas usuários cadastrados podem acessar o sistema e alterar os dados;

Eficiência: as consultas e relatórios devem ser executados com desempenho satisfatório, sem lentidão perceptível;

Aderência aos requisitos: todas as funcionalidades previstas no escopo do projeto devem estar implementadas e operacionais.

3.4.3 Resultados dos testes

Os testes realizados demonstraram que o sistema atendeu aos requisitos funcionais e não funcionais definidos. As principais observações foram:

O login funcionou adequadamente, restringindo o acesso apenas a usuários registrados na tabela `tblfuncionarios`;

Os cadastros e alterações foram executados corretamente, com inserção e atualização imediata dos dados no banco de dados SQL Server;

As consultas retornaram resultados precisos e filtrados conforme os critérios informados pelo usuário;

O relatório de apólices com os parâmetros: Período: 30/10/2025 a 30/10/2026 e Status: Todos apresentou as informações de forma clara e dentro dos parâmetros esperados;

O sistema manteve estabilidade e consistência mesmo após múltiplas operações consecutivas de leitura e escrita.

Durante os testes, também foram realizados ensaios com dados fictícios e situações de erro proposital, como campos vazios e login incorreto, para avaliar o tratamento de exceções.

Esses testes mostraram que o sistema reage de forma segura, apresentando mensagens de alerta sem comprometer o funcionamento geral.

3.4.4 Ajustes e melhorias

Com base nos resultados obtidos, foram aplicados pequenos ajustes para otimizar o desempenho e a usabilidade do sistema, incluindo:

Melhoria na formatação dos relatórios para melhor visualização dos dados;

Adição de mensagens de confirmação e alerta;
Ajustes no layout das telas de cadastro e consulta;
Implementação de tratamento de exceções para evitar falhas na conexão com o banco de dados.

Essas melhorias contribuíram para um produto final mais estável e intuitivo, atendendo às expectativas levantadas no início do projeto.

3.4.5 Conclusão dos testes

A etapa de testes e validação confirmou a eficácia da metodologia aplicada e a coerência entre os objetivos definidos e os resultados alcançados. O sistema mostrou-se estável, funcional e seguro, demonstrando que o desenvolvimento foi conduzido de forma técnica e planejada, conforme os princípios da engenharia de software destacados por Sommerville (2007) e Pressman (2011).

Dessa forma, conclui-se que o software desenvolvido cumpre sua finalidade de auxiliar a gestão administrativa de uma corretora de seguros, otimizando o controle das apólices, reduzindo erros operacionais e proporcionando maior eficiência na tomada de decisões.

4. Conclusão

O presente trabalho teve como objetivo o desenvolvimento de um sistema de controle de apólices para uma corretora de seguros, visando à otimização de processos administrativos e à organização das informações referentes a clientes, seguradoras e contratos.

A pesquisa foi fundamentada em princípios de engenharia de software, modelagem de dados e programação orientada a objetos, integrando o conhecimento teórico adquirido ao longo do curso com uma aplicação prática voltada ao mundo real.

Durante o processo de elaboração, foram aplicadas metodologias que permitiram a construção de um software funcional, estruturado e seguro. A adoção do modelo incremental de desenvolvimento, conforme proposto por Pressman (2011), possibilitou a entrega gradual de módulos e a realização de testes contínuos, garantindo maior confiabilidade e flexibilidade durante o processo de implementação. Além disso, o uso da abordagem qualitativa contribuiu para compreender as necessidades específicas do ambiente corporativo e direcionar as funcionalidades do sistema às demandas da corretora de seguros.

O banco de dados relacional desenvolvido no SQL Server e a aplicação criada em C# (Windows Forms) mostraram-se eficientes para o controle das informações e para a execução das operações de cadastro, consulta e geração de relatórios. A integração entre esses componentes proporcionou ao sistema estabilidade e desempenho satisfatório, atendendo aos critérios de segurança, usabilidade e integridade definidos na metodologia.

Os resultados obtidos confirmaram a eficácia do sistema no apoio às atividades administrativas, especialmente no que se refere ao gerenciamento de apólices e controle de vencimentos, permitindo maior agilidade na consulta de dados e redução de erros manuais. Além de atender ao objetivo prático da pesquisa, o projeto também reforçou a importância da aplicação dos conhecimentos de engenharia de software e banco de dados na solução de problemas reais do mercado.

Conforme salientam Sommerville (2007) e Laudon e Laudon (2014), sistemas de informação bem projetados desempenham papel estratégico nas organizações, ao promover eficiência operacional e suporte à tomada de decisão. Nesse sentido, o sistema CorretoraTCC representa uma contribuição significativa para o

aprimoramento das rotinas de uma corretora, fornecendo uma ferramenta de gestão acessível e adaptada às suas necessidades.

Como proposta para trabalhos futuros, sugere-se a ampliação do sistema com a implementação de novos módulos, como controle financeiro, emissão de boletos e notificações automáticas de vencimentos por e-mail, além da migração para ambiente web ou aplicativo móvel, o que ampliaria o alcance e a flexibilidade de uso da aplicação.

O trabalho alcançou os objetivos propostos, unindo teoria e prática de forma integrada e demonstrando a relevância da tecnologia da informação como ferramenta de modernização e eficiência nos processos empresariais. O desenvolvimento do sistema de controle de apólices reafirma o papel do profissional de Informática para Negócios como agente de inovação, capaz de transformar necessidades organizacionais em soluções tecnológicas concretas e eficazes.

REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. ABNT NBR 10520: informação e documentação – citações em documentos – apresentação. Rio de Janeiro: ABNT, 2023.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. ABNT NBR 6023: informação e documentação – referências – elaboração. Rio de Janeiro: ABNT, 2018.

DEV MEDIA. MER e DER: Modelagem de Bancos de Dados. Disponível em: <https://www.devmedia.com.br/mer-e-der-modelagem-de-bancos-de-dados/14332>. Acesso em: 8 out. 2025.

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

GIL, Antonio Carlos. **Métodos e técnicas de pesquisa social**. 7. ed. São Paulo: Atlas, 2019.

GODOY, Arilda Schmidt. **Introdução à pesquisa qualitativa e suas possibilidades**. Revista de Administração de Empresas, São Paulo, v. 35, n. 2, p. 57–63, mar./abr. 1995.

LAUDON, Kenneth C.; LAUDON, Jane P. **Sistemas de informação gerenciais**. 12. ed. São Paulo: Pearson, 2014.

MICROSOFT. Documentação oficial do SQL Server. Microsoft Learn. Disponível em: <https://learn.microsoft.com/pt-br/sql/>. Acesso em: 8 out. 2025.

OBJECT MANAGEMENT GROUP (OMG). Unified Modeling Language (UML): Superstructure Specification, Version 2.5.1. Needham: OMG, 2017. Disponível em: <https://www.omg.org/spec/UML>. Acesso em: 8 out. 2025.

PRESSMAN, Roger S. **Engenharia de Software: uma abordagem profissional**. 7. ed. São Paulo: McGraw-Hill, 2011.

SCHWABER, Ken; SUTHERLAND, Jeff. The Scrum Guide: o guia definitivo do Scrum. 2020. Disponível em: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Portuguese-BR.pdf>. Acesso em: 8 out. 2025.

SOMMERVILLE, Ian. **Engenharia de Software**. 8. ed. São Paulo: Pearson Addison-Wesley, 2007.

SEVERINO, Antônio Joaquim. **Metodologia do trabalho científico**. 24. ed. São Paulo: Cortez, 2016.

LINHA DE CÓDIGO. Scrum em 2 minutos. Disponível em:
<http://www.linhadecodigo.com.br/artigo/2375/scrum-em-2-minutos.aspx>.
Acesso em: 8 out. 2025.

DESENVOLVIMENTO ÁGIL. Scrum – Guia de Referência. Disponível em:
<http://www.desenvolvimentoagil.com.br/scrum/>. Acesso em: 8 out. 2025.

GLOSSÁRIO

Algoritmo: Conjunto finito de instruções lógicas e ordenadas utilizadas para resolver um problema ou executar uma tarefa específica em um programa de computador.

Apólice: Documento que formaliza o contrato de seguro entre a seguradora e o segurado, definindo as coberturas, prazos e valores envolvidos.

Banco de Dados Relacional: Modelo de organização de dados baseado em tabelas interligadas por chaves primárias e estrangeiras, que permite o armazenamento estruturado e o acesso eficiente às informações.

C#: Linguagem de programação orientada a objetos desenvolvida pela Microsoft, utilizada principalmente para o desenvolvimento de aplicações na plataforma .NET.

Cliente: Pessoa física ou jurídica que contrata um serviço de seguro por meio da corretora, sendo titular das apólices registradas no sistema.

DataGridView: Componente gráfico do Windows Forms que permite exibir, editar e manipular dados em formato de tabela dentro de uma aplicação C#.

Engenharia de Software: Área da computação responsável pela aplicação de princípios, métodos e técnicas para o desenvolvimento sistemático, controlado e eficiente de software de qualidade.

Interface: Elemento visual de um sistema que possibilita a interação entre o usuário e as funcionalidades da aplicação.

Login: Processo de autenticação que valida a identidade do usuário por meio de um nome e senha cadastrados no sistema.

MenuStrip: Componente gráfico utilizado em aplicações Windows Forms para organizar menus e comandos de navegação na interface principal.

Metodologia Incremental: Modelo de desenvolvimento de software que realiza a construção do sistema de forma progressiva, por meio da entrega de versões parciais e funcionais até o produto final.

Modelagem Entidade-Relacionamento (MER): Representação gráfica das entidades, atributos e relacionamentos de um banco de dados, utilizada na etapa de projeto lógico.

Programação Orientada a Objetos (POO): Paradigma de programação que organiza o código em classes e objetos, promovendo a reutilização, encapsulamento e modularidade do software.

Relatório de Vencimentos: Funcionalidade do sistema responsável por listar as apólices que estão próximas da data de vencimento, facilitando o controle e a renovação dos contratos.

Scrum: Framework de metodologia ágil utilizado para o gerenciamento de projetos de software, baseado em ciclos curtos de entrega chamados *sprints*.

SQL Server: Sistema de gerenciamento de banco de dados relacional (SGBDR) desenvolvido pela Microsoft, amplamente utilizado em aplicações corporativas.

Tabela: Estrutura fundamental de um banco de dados, composta por linhas e colunas, onde são armazenados os registros de informações.

UML (Unified Modeling Language): Linguagem de modelagem padronizada utilizada para representar graficamente a estrutura e o comportamento de sistemas orientados a objetos.

Usuário: Pessoa autorizada a acessar o sistema mediante autenticação, podendo realizar operações como cadastro, consulta e emissão de relatórios.

Windows Forms: Plataforma gráfica do .NET Framework utilizada para o desenvolvimento de aplicações desktop no ambiente Windows.