

N. CLASS.	M005 J
CUTTER	C1775
ANO/EDIÇÃO	2015

CENTRO UNIVERSITÁRIO DO SUL DE MINAS - UNIS-MG
BACHARELADO EM SISTEMAS DE INFORMAÇÃO
GRACIANA FERNANDES CAMBRAIA

**SISTEMA DE INFORMAÇÃO PARA CÁLCULO DE
COMISSÕES DE VENDA DE CURSOS E LIVROS:**
Desenvolvimento de uma aplicação para gestão de vendas

Varginha/MG
2015

FEPESMIG

GRACIANA FERNANDES CAMBRAIA

**SISTEMA DE INFORMAÇÃO PARA CÁLCULO DE
COMISSÕES DE VENDA DE CURSOS E LIVROS:
Desenvolvimento de uma aplicação para gestão de vendas**

Monografia apresentada ao curso de Bacharelado em Sistemas de Informação do Centro Universitário do Sul de Minas – UNIS/MG como pré-requisito para obtenção do grau de bacharel, sob orientação do Profa. Ma Andrea Cristina Oliveira Alves.

**Varginha/MG
2015**



GRACIANA FERNANDES CAMBRAIA

PROPOSTA DE IMPLEMENTAÇÃO DE UM SISTEMA PARA CÁLCULO DE

COMISSÕES DE VENDA DE CURSOS E LIVROS:

Desenvolvimento de uma aplicação para gestão de vendas

Monografia apresentada ao curso de Bacharelado em Sistemas de Informação do Centro Universitário do Sul de Minas – UNIS/MG, como pré-requisito para obtenção do grau de bacharel pela Banca Examinadora composta pelos membros:

Aprovado em 12/11/2015

Prof. Ma Andrea Cristina Oliveira Alves

Prof. Especialista César Fernandes Ribeiro Filho

AGRADECIMENTOS

Agradeço aos meus colegas, professores e a minha família por terem ajudado na construção deste trabalho.

RESUMO

A gestão da informação é fundamental dentro das instituições de ensino auxiliando no seu sucesso. Os dados devem ser processados gerando informações que auxiliem a tomada de decisão. No setor educacional isso é verificado, uma quantidade enorme de dados que não são utilizados de forma adequada e eficaz. O presente projeto visa o desenvolvimento e implementação de um sistema de informação em uma instituição de venda de cursos e livros com a finalidade de melhorar os processos e ajudar a gestão na tomada de decisão. O software tem o objetivo de receber o lançamento das vendas de cursos e livros e através das alíquotas pré-estabelecidas ele calcula a comissão e lança o valor para o vendedor. Estes dados podem ser buscados através de um relatório que irá auxiliar os gestores da instituição na tomada de decisão. No desenvolvimento do projeto, foram necessários a utilização de métodos de desenvolvimento relacionados à área de Engenharia de Software, da Linguagem C# e do Banco de Dados SQL. Automatizar todo esse processo foi muito importante, pois ajudou na gestão da informação dentro da instituição.

Palavras-chave: Sistemas. Banco de dados. Administração.

ABSTRACT

Information management is fundamental within the educational institutions aiding in their success. Data must be processed generating information that can help decision making. In the educational sector this is checked, a huge amount of data that is not used properly and effectively. This project aims at the development and implementation of an information system in a sale of the institution of courses and books in order to improve processes and help management in decision making. The software is designed to receive the launch of sales courses and books and through the pre-established rates it calculates the commission and cast the value to the seller. These data can be fetched in a report that will assist the institution managers in decision making. In developing the project, it took the use of method development related to Software Engineering area, the language C # and SQL Database. Automate this whole process was very important because it helped in the management of information within the institution.

Keywords: Systems. Database. Administration.

SUMÁRIO

1 INTRODUÇÃO.....	9
1.1 Tema específico	9
1.2 Problema de pesquisa.....	9
1.3 Problematização.....	10
1.4 Hipóteses	10
1.5 Objetivo geral	10
1.6 Objetivos específicos:.....	10
1.7 Justificativa.....	11
2 REVISÃO DA LITERATURA.....	12
2.1 Os benefícios da Informatização	12
2.2 A importância dos Sistemas de Informação	12
2.3 Sistemas de Informação em Ambientes Educacionais	14
2.4 Plataformas de Desenvolvimento	14
2.4.1 Plataforma.Net.....	14
2.5 Linguagem de Programação	15
2.5.1 Linguagem C#	16
2.5.2 A linguagem C# e suas características	17
2.5.3 Evolução da linguagem	18
2.6 Banco de Dados.....	20
2.7 Sistema de Gerenciamento de Dados – SGBD.....	21
2.8 MySQL.....	23
3. MATERIAL E MÉTODOS.....	24
3.1 Definição do projeto	25
3.2 Coleta de dados.....	25
3.3 Stakeholders/Usuários	26

3.4 Pesquisa bibliográfica e definição de ferramentas	26
4 RESULTADOS	28
4.1 Levantamento de requisitos	28
4.2 Requisitos Funcionais do Sistema	28
4.3 Requisitos Não Funcionais do Sistema	30
4.4 Modelagem e diagramas	31
4.5 Diagrama de entidade relacionamento	32
4.6 Resumo das Características do Sistema	33
5 CONCLUSÃO	40
REFERÊNCIAS	41

1 INTRODUÇÃO

Com o avanço da tecnologia cada vez mais se vê necessária à automatização dos processos. Isso se dá através da implantação de sistemas informatizados que irão agilizar o trabalho que antes era feito de forma manual. Várias empresas têm observado que a tecnologia veio para somar e tem sido vista como um diferencial competitivo em vários setores.

Outro aspecto relevante é que os dados passaram a fazer parte do ativo da empresa desempenhando um papel fundamental, pois quando analisados dentro de um contexto ajudam seus gestores na tomada de decisão.

A organização objeto deste estudo atua no setor educacional onde são feitas vendas de cursos e livros que preparam o cliente para prestar concursos públicos e proporciona seu crescimento através de Pós-Graduação. O intuito é auxiliar o cliente em sua formação e direção profissional. É um setor que tem crescido no Brasil devido à busca por estabilidade e capacitação profissional para o mercado de trabalho.

Este trabalho visa justamente desenvolver um software para essa empresa do setor educacional que comercializa cursos e livros, auxiliando-a no cálculo das comissões pagas a seus vendedores. O objetivo é melhorar os processos, trazendo rapidez e precisão a um trabalho feito atualmente de forma manual.

1.1 Tema específico

Criação e implementação de um software para cálculo de comissão de vendas.

1.2 Problema de pesquisa

Como resolver o problema de cálculo manual de comissões de vendas de uma instituição de ensino utilizando a linguagem C#?

1.3 Problematização

Atualmente na instituição de ensino analisada nesse projeto, as comissões das vendas de livros e cursos são calculadas manualmente, uma a uma, usando calculadora de acordo com o percentual de comissão pago, o que despense de tempo e trabalho. Esse trabalho busca uma nova alternativa através de um software que irá calcular automaticamente essas comissões, auxiliando de forma mais rápida o gerenciamento do setor comercial da empresa. Foram feitos estudos para analisar a viabilidade de se desenvolver este software na linguagem C# (C Sharp).

1.4 Hipóteses

É possível utilizar a linguagem C# por ser totalmente orientada a objetos, fácil de aprender e utilizar, sendo robusta e com boa performance.

Pode-se utilizar dos conceitos práticos e teóricos apresentado na engenharia de software na implementação de sistemas para qualquer ramo de atividade.

1.5 Objetivo geral

Desenvolver um software baseado na Linguagem C# e banco de dados MySQL para informatizar o cálculo da comissão das vendas de cursos e livros e emissão de relatórios específicos de uma instituição de ensino.

1.6 Objetivos específicos:

Desenvolver um software com as seguintes qualidades:

- Capaz de realizar os processos de cálculos de forma mais rápida, segura e confiável;
- Capaz de gerar relatórios específicos;

- Fazer uso de linguagem de programação adequada para possibilitar uma performance desejável.
- Utilizar um sistema de gerenciamento de bancos de dados que proporcione integridade e segurança aos dados.

1.7 Justificativa

A motivação para o desenvolvimento de uma aplicação computacional para o cálculo de comissões surgiu da necessidade observada na organização que será objeto deste estudo que atua no ramo de venda de cursos e livros para concursos. Os colaboradores recebem salário fixo mais comissões e estas comissões são calculadas de forma manual levando muito tempo e o resultado não é confiável.

Existem hoje em dia alguns softwares com o intuito de calcular as comissões de vendas, mas são genéricos e não se adaptam para necessidade citada acima, além do alto custo. Eles são desenvolvidos para determinados setores e com dados pré-estabelecidos.

Portanto ao criar um software específico para o cálculo comissões pode-se introduzir os dados necessários para gerar o resultado esperado de acordo com a atividade da empresa. O setor comercial terá mais facilidade para gerar relatórios e ajudar na tomada de decisões, além de ter seu trabalho facilitado.

2 REVISÃO DA LITERATURA

2.1 Os benefícios da Informatização

A presente pesquisa se dará em uma empresa no setor educacional de venda de cursos e livros para concursos, onde os colaboradores recebem comissão pelas vendas, as quais são calculadas manualmente. Esse processo é dispendioso e não é seguro, portanto sugerimos o desenvolvimento de um software para suprir esta necessidade.

Na busca de melhorar e agilizar os processos, softwares tem sido desenvolvidos para auxiliar pessoas e empresas em suas atividades diárias. Geralmente são criados de forma generalizada para suprir necessidades gerais e a pessoa ou empresa tem que se adaptar a ele. Mas o diferencial de tais softwares é a sua especificidade, ou seja, como irão tratar de problemas específicos.

Devido ao aumento do uso de Tecnologias e o uso específico dos softwares, os programadores precisam buscar conhecer o problema e suas especificidades e através disso buscar parâmetros para o desenvolvimento do software.

Para isso, segundo Laudon e Laudon (1999) deve-se observar o ciclo de vida deste sistema de informação:

- Definir o problema: entender a necessidade do projeto para a empresa;
- Estudar como os processos são feitos e buscar soluções viáveis;
- Desenvolver o projeto através das prerrogativas encontradas;
- Instalar e testar;
- Suporte após instalação.

O desenvolvimento específico de um software se faz necessário através de um conhecimento do funcionamento atual do processo usado. Desta forma analisaremos o cenário atual e suas peculiaridades com a finalidade de solucionar o problema proposto.

2.2 A importância dos Sistemas de Informação

A tecnologia avança rapidamente, todos os dias, novos sistemas, softwares e equipamentos são desenvolvidos, mas ainda existem empresas e pessoas que não tem acesso a todo este benefício. Isto se deve muitas vezes por uma falta de visão dos benefícios que a informatização pode trazer que na maioria das vezes é compreendida como despesa e não como investimento que traz redução de custos e várias facilidades. Ao contrário do que se pensa utilizar softwares de gestão não é mais um privilégio de alguns e os benefícios gerados são muitos como: rapidez nos processos, controle de estoque, controle financeiro e gestão centralizada da informação. Considerando-se a importância da informatização que buscam cada vez mais atender de forma rápida e eficiente as necessidades de informação de seus usuários fornecendo subsídios para o desenvolvimento das atividades fundamentais.

Segundo Rowley (1994), “a informática tem exercido uma influência fundamental no funcionamento das organizações e serviços de informação”.

Com o advento do computador e das novas tecnologias e sua conseqüente introdução, “resultou em padronização, aumento da eficiência, cooperação e melhores serviços” (ROWLEY, 1994) oferecidos.

Entretanto, para que esses diferenciais sejam atingidos é preciso que se tenha claro os objetivos a que se propõe a informatização.

Dentre as muitas e variadas razões que justificam a opção por um sistema informatizado de gerenciamento, Rowley (1994), aponta:

- Os computadores possibilitam a redução do número de tarefas repetitivas. Em geral, os dados serão inseridos uma única vez e, daí em diante, poderão ser acessados e modificados;
- Os sistemas informatizados podem ser mais baratos e mais eficientes;
- Podem propiciar a introdução de serviços que não existiam antes;
- Controle adicional de todas as funções que se consegue com a ajuda de informações gerenciais mais abrangentes que justificam um processo decisório mais eficaz.

Portanto o sistema informatizado trará melhorias e rapidez nos processos, fazendo com que as informações sejam armazenadas de maneira eficaz e sua busca seja rápida quando necessária.

De uma forma ampla, a finalidade da informatização é agilizar, aumentar a eficiência e a precisão na recuperação da informação. Figueiredo (1992), diz que a informatização deve ser pensada e “implementada” a partir do momento em que:

- os procedimentos manuais se tornem inadequados;
- é possível ampliar a gama de serviços oferecidos;
- a cooperação se torna imprescindível.

Não apenas para suprir essas necessidades, mas a informatização vem para agregar valor à empresa, já que seu *feedback* rápido proporciona a empresa sair na frente de seus concorrentes como um diferencial competitivo.

2.3 Sistemas de Informação em Ambientes Educacionais

O setor educacional tem crescido no Brasil e se tornado bastante competitivo, por isso as empresas tem percebido a necessidade de gerenciar melhor seu negócio. Antes focavam mais no setor pedagógico, como corpo discente, docente e metodologias. Agora perceberam a necessidade da integração entre os setores administrativo, financeiro e pedagógico, e estão apostando em Sistemas de Informação que proporcionem isso.

Segundo Auler (1988) o meio externo influencia diretamente a organização, portanto criar uma visão estratégica e utilizar ferramentas administrativas e tecnológicas modernas é um diferencial para as empresas do setor educacional.

2.4 Plataformas de Desenvolvimento

2.4.1 Plataforma.Net

Segundo Don Box (2003), a Plataforma.NET é centrada ao redor de uma *Common Language Runtime* (CLR, conceito similar ao da Java Virtual Machine, JVM) e um conjunto de bibliotecas que pode ser empregado em uma grande variedade de linguagens, as quais podem trabalhar juntas, já que todas são compiladas para uma mesma linguagem intermediária, a Microsoft Intermediate Language (MSIL). Assim, é possível desenvolver aplicativos mesclando C# e Visual Basic ou qualquer outra linguagem suportada. Don Box (2003)

Segundo Maya (2015) o Visual Studio.Net, dedicado a plataforma .Net é extremamente completa, oferecendo diversos serviços e facilidades ao desenvolvedor. Para aqueles desenvolvedores que desenvolvem aplicativos com banco de dados o Visual Studio.Net oferece um recurso que adiciona ao formulário todos os componentes e código para manipular as principais funções que um banco de dados precisa, como consulta, inserção, remoção, e atualização. Além disso, o Visual Studio.Net é simples de usar, possui um ambiente bastante organizado, assim como o antigo ambiente que o Visual Basic 6 oferecia e era extremamente elogiado por isso.

O Visual Studio.Net além de oferecer funcionalidades também oferece constantes atualizações, assim em média a cada dois anos uma nova versão, ainda mais completa, é lançada pela Microsoft. Uma desvantagem é que é pago e não funciona em outros sistemas operacionais a não ser no Windows. Já sua versão gratuita é bastante limitada e o software desenvolvido com ela não pode ser comercializado.

2.5 Linguagem de Programação

A função da linguagem de programação é servir como um meio de comunicação entre os humanos e os computadores. Existem dois tipos de linguagens de programação: as de baixo nível e as de alto nível.

As linguagens de baixo nível são interpretadas diretamente pelo computador, tendo um resultado rápido, porém é muito difícil e incômodo se trabalhar com elas.

Segundo Lima e Reis (2002) pode-se notar que é uma linguagem bastante complicada. Já as linguagens de alto nível são mais fáceis de trabalhar e de entender, as ações são representadas por palavras de ordem (exemplo faça, imprima, etc.) geralmente em inglês, foram feitos assim para facilitar a memorização e a lógica. Elas não são interpretadas diretamente pelo computador, sendo necessário traduzi-las para linguagem binária utilizando-se de um programa chamado compilador.

Para Farrer (1985), quando programamos em uma linguagem de programação de alto nível primeiramente criamos um arquivo de texto comum contendo a lógica do programa, ou seja, é onde falamos ao computador como deve ser feito o que queremos. Este arquivo de texto é chamado de código-fonte, cada palavra de ordem dentro do código-fonte é chamada de instrução. Após criarmos o código-fonte devemos traduzir este arquivo para linguagem

binária usando o compilador correspondente com a linguagem na qual estamos programando. O compilador irá gerar um segundo arquivo que chamamos de executável ou programa, este arquivo gerado é interpretado diretamente pelo computador.

Existem algumas linguagens de programação que não necessitam de compiladores, como o PHP, uma linguagem dedicada à produção de websites dinâmicos. As instruções em PHP são compiladas e executadas ao mesmo tempo.

Cada linguagem de programação é diferente da outra, contendo palavras-chave próprias. Outros exemplos de linguagens de alto nível são C++, Java, C#, Delphi (Pascal), PHP, Visual Basic, etc. Como a proposta deste trabalho é implementar utilizando a linguagem C#, a mesma será analisada de forma mais profunda no item a seguir.

2.5.1 Linguagem C#

O desenvolvimento do C# (pronuncia-se: C Sharp), teve início no final do século XX (1999). A convite da Microsoft, Anders Hejlsberg formou uma equipe de programadores para desenvolver uma nova linguagem de programação. A primeira linguagem criada pela equipe de Hejlsberg recebe o nome de Cool. Mais tarde, na virada de século, a linguagem Cool foi rebatizada para C#.

Para Lima e Reis (2002), o C# surgiu com a finalidade de possibilitar a criação de soluções executáveis sobre a plataforma .NET Framework, com o intuito de flexibilizar o desenvolvimento de aplicativos. Dessa maneira, o desenvolvedor não cria soluções para um dispositivo eletrônico específico, e sim para a plataforma .NET Framework.

Para a codificação de sistemas baseados em .NET Framework, a Microsoft lançou o Visual Studio. Esta ferramenta é uma IDE (*Integrated Development Environment*) que conta com um editor de código, compilador, com modelos de projetos de exemplos, designers e assistente de códigos.

Msdn (2012) diz que os algoritmos escritos em C# são executados no .NET Framework, um componente para Windows que inclui um sistema de execução virtual da aplicação por meio do *Common Language Runtime* (CLR) e um conjunto de bibliotecas de classes, além de contar com uma solução comercial da Microsoft, que faz parte da infraestrutura de linguagem comum (CLI). O CLI é tido como um padrão internacional para a

criação e execução de ambientes de desenvolvimento onde as linguagens e as bibliotecas trabalham em sincronia.

Ainda de acordo com Msdn (2012), a compilação em C# é processada em uma Intermediate Language (IL) que está em conformidade com a especificação CLI. O código IL e seus recursos, tais como bitmaps e strings, são armazenados no disco rígido em um arquivo executável chamado de Assembly. No momento da execução de um programa em C#, o Assembly é carregado para o CLR. Uma vez aprovado os requisitos de segurança, o CLR executa a compilação Just in Time (JIT) para converter o código IL em linguagem de baixo nível.

2.5.2 A linguagem C# e suas características

A criação do C# foi baseada nas linguagens de programação C, C++ e Java. Deste modo, programadores habituados com uma dessas linguagens são capazes de desenvolver soluções sem maiores dificuldades, pois ela reúne recursos de cada uma destas linguagens base e adiciona novos recursos exclusivos.

Em relação à linguagem C, o C# tem foco na compilação de soluções de alto nível, já a linguagem C é voltada para o desenvolvimento de baixo nível.

Msdn (2012) ressalta que, o processo de compilação do C# é mais simples e mais flexível comparado ao C++ ou Java, pois elimina-se a necessidade de arquivos de cabeçalhos separados e não há a necessidade da declaração de métodos e tipos em uma ordem específica. Um arquivo contendo códigos de programação podem ser definidos quaisquer números de classes, estruturas, interfaces e eventos que forem necessários.

Segundo Lima e Reis (2002), as características essenciais do C# são:

- **Simplicidade:** os projetistas de C# costumam dizer que essa linguagem é tão poderosa quanto o C++ e tão simples quanto o Visual Basic;
- **Completamente orientada a objetos:** em C#, qualquer variável tem de fazer parte de uma classe;
- **Fortemente tipada:** isso ajudará a evitar erros por manipulação imprópria de tipos, atribuições incorretas, etc.;
- **Gera código gerenciado:** assim como o ambiente .NET é gerenciado, assim também é o C#;

- **Tudo é um objeto:** System.Object é a classe base de todo o sistema de tipos de C#;
- **Controle de versões:** cada assembly gerado, seja como EXE ou DLL, tem informação sobre a versão do código, permitindo a coexistência de dois assemblies homônimos, mas de versões diferentes no mesmo ambiente;
- **Suporte a código legado:** o C# pode interagir com código legado de objetos COM e DLL's escritas em uma linguagem não gerenciada;
- **Flexibilidade:** se o desenvolvedor precisar usar ponteiros, o C# permite, mas ao custo de desenvolver código não gerenciado, chamado "unsafe".
- **Linguagem gerenciada:** os programas desenvolvidos em C# executam num ambiente gerenciado, o que significa que todo o gerenciamento de memória é feito pelo runtime via o GC (Garbage Collector), e não diretamente pelo programador, reduzindo as chances de cometer erros comuns a linguagens de programação onde o gerenciamento da memória é feito diretamente pelo programador.

2.5.3 Evolução da linguagem

A primeira versão do C# começou a ser usado a partir do Visual Studio 2002. Sua principal característica era a capacidade de escrever códigos gerenciáveis.

Lima e Reis (2002) acrescentam que a linguagem C# faz parte do conjunto de ferramentas oferecidas na plataforma .NET e é definida como uma linguagem simples, robusta, orientada a objetos, fortemente tipada e altamente escalável a fim de permitir que uma mesma aplicação possa ser executada em diversos dispositivos de hardware, independentemente destes serem PCs, handhelds ou qualquer outro dispositivo móvel. Embora seja possível usar várias linguagens na plataforma .NET, C# é aquela que deve sempre ser considerada em primeiro lugar, pois oferece poder, facilidade, flexibilidade e é a linguagem nativa para a plataforma .NET. O C# resolve o abismo entre as linguagens de baixo nível e alto nível.

A segunda versão da linguagem (2.0) foi introduzida com o Visual Studio 2005 e Visual Studio 2005 Express. De acordo com Msdn (2005) as novidades são: tipos genéricos, iteradores, classe parcial, tipos anuláveis, métodos anônimos, qualificador de alias de namespace, classes estáticas, modificadores externos para declarar métodos declarados

externamente, possibilidade de definir diferentes níveis de acessibilidades aos métodos gets e sets, maior flexibilidade no tipo de retorno e parâmetros ao passar para um delegate, buffer de tamanho fixo.

Com o Visual Studio 2008 e Visual Studio 2008 Express, o C# 3.0 (“C# Orcas”), de acordo com Msdn (2007) introduziu várias extensões de linguagem que se baseiam no C# 2.0. A versão 3.0 conta ainda com métodos de extensão, expressões lambda, tipos anônimos, tipos implícitos de arrays, expressões de consulta e árvores de expressões.

Na sua versão 4.0, contando com o Visual Studio 2010 e Visual Studio 2010 Express, os novos recursos agregados, de acordo com Burrows (2010), compreendem: DLR, parâmetros opcionais e de argumentos nomeados, propriedades indexadas, suporte COM.

Atualmente, o C# está na versão 5.0 acompanhada com o Visual Studio 2012 e Visual Studio 2012 Express. O Website kunal-chowdhury (2012) diz que são duas as novas funcionalidades apresentadas na quinta versão da linguagem: Async e Caller Information. Async é utilizada para indicar se o método, expressão lambda, ou método anônimo são assíncronos. Caller Information para obter informações sobre a chamada a um método, obter o caminho e número de linhas do código-fonte e o nome do membro que o chamou. Caller Information é utilizado para a detecção, depuração e criação de ferramentas de diagnósticos.

A criação do C# foi baseada em C, C++ e Java. Dessa maneira, o desenvolvedor possui um apoio referencial no aprendizado da linguagem, não precisando estudar novos conceitos e paradigmas de programação opostos. Desde o seu lançamento, o C# permite o desenvolvimento de aplicações executadas exclusivamente sobre a plataforma .NET Framework.

Dentre algumas das características essenciais do C#, em ordem cronológica de versão, destacam-se: (1.0) totalmente orientada a objetos, gerenciamento automático de memória; (2.0) tipos genéricos, métodos anônimos, classes estáticas; (3.0) tipos anônimos, expressões de consulta e árvore de expressões; (4.0) propriedades indexadas, suporte COM e DLR; (5.0) Async e Caller Information.

A linguagem C# veio acompanhada de uma IDE comercial de desenvolvimento: Visual Studio. A partir de 2005, contou ainda com uma IDE gratuita, denominada Visual Studio Express. Em ordem cronológica, o Visual Studio conta com as seguintes versões: 2002, 2005, 2008, 2010 e 2012.

Segundo Lima e Reis (2002) a utilização do Visual Studio como plataforma de desenvolvimento e o C# garantem algumas vantagens nas construções de aplicativos, tais

como: programação orientada a objetos, uso de eventos nos controles, validação de dados e tratamento de erros.

2.6 Banco de Dados

O avanço tecnológico da informática tem gerado uma grande quantidade de dados, os quais precisam ser armazenados e transformados em informações.

Em se tratando de empresas esses dados são considerados com um diferencial competitivo, pois quando coletados de maneira correta geram informações relevantes para tomada de decisão, portanto o armazenamento e processamento dessas informações se tornam tão importantes e passíveis de atenção.

Segundo Date (2000) um Banco de Dados é uma coleção de dados relacionados e armazenados em algum dispositivo com um significado inerente, projetado para um propósito específico.

Segundo Korth (1994), um banco de dados “é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico”, ou seja, sempre que for possível agrupar informações que se relacionam e tratam de um mesmo assunto, posso dizer que tenho um banco de dados.

Algumas das principais características de um banco de dados são:

- Compartilhamento de dados comuns entre aplicações;
- Arquivar dados sem duplicação;
- Facilidade de consulta;
- Facilidade de manutenção.

Antes da implementação de um Banco de Dados é necessário pensar no processo de planejamento, conhecer qual a finalidade do mesmo e buscar junto aos usuários um melhor caminho para que o Banco de Dados se torne eficiente.

Por isso, de acordo com Machado e Abreu (1996) a sua estruturação passa por um Projeto de três fases: Projeto Conceitual, Projeto Lógico e Projeto Físico.

- **Projeto Conceitual:** descreve a realidade do ambiente do problema, constituindo-se em uma visão global dos principais dados e relacionamentos. Dados os requisitos do sistema,

o esquema conceitual é uma descrição concisa dos requerimentos do usuário, usando um modelo de dados. Ela serve para checar se todos os pedidos dos usuários estão sendo atendidos e se não há conflitos entre eles. Seu objetivo é descrever as informações contidas em uma realidade, as quais irão estar armazenadas em um banco de dados.

- **Projeto Lógico:** é a descrição das estruturas de representação da base de dados. Tem seu início a partir do Modelo Conceitual. O modelo conceitual é transformado em um modelo de dados de implementação. Ele resulta no esquema do banco de dados no modelo de dados de implementação usado pelo SGBD.
- **Projeto Físico:** Irá partir do Modelo Lógico e descreve as estruturas físicas de armazenamento de dados, tais como: tamanho do campo, índices, tipos de preenchimento destes campos, nomenclaturas, etc., projetadas de acordo com os requisitos de processamento e uso dos recursos computacionais. Este modelo detalha estudo dos métodos de acesso do SGBD, para elaboração dos índices de cada informação colocada nos Modelos Conceitual e Lógico.

2.7 Sistema de Gerenciamento de Dados – SGBD

Para gerenciar estes dados temos os SGBD – Sistemas Gerenciadores de Banco de Dados; um software que permite manipular os dados de forma rápida e eficaz, pois permite recuperar as informações de forma praticamente instantânea. Segundo Velloso (2003, p. 87) um Sistema de Banco de dados é composto essencialmente de:

- Uma base de dados;
- Um Sistema Gerenciado de Banco de Dados;
- Linguagem de exploração;
- Programas voltados a necessidades objetivas.

Segundo Date (2000) vejamos algumas vantagens e desvantagens de um SGBD:

São algumas vantagens de um SGBD:

- Independência de dados;
- Controle de redundância de dados;

- Garantia de integridade de dados;
- Privacidade de dados;
- Facilidade de criação de novas aplicações;
- Segurança de dados;
- Controle automático de relacionamento entre registros;
- Otimização de espaço e armazenamento.

Já as desvantagens podem ser enumeradas como:

- Custo: A implementação de um sistema de SGBD pode ser cara e demorada, especialmente em grandes organizações. Requisitos de formação podem ser bastante onerosos.
- Segurança: Mesmo com salvaguardas no lugar, pode ser possível para alguns usuários não autorizados acessar o banco de dados. Em geral, o acesso de banco de dados é uma proposição de tudo ou nada. Uma vez que um usuário não autorizado fica no banco de dados, eles têm acesso a todas as tabelas, e não apenas algumas. Dependendo da natureza dos dados envolvidos, essas quebras na segurança também pode representar uma ameaça à privacidade individual. Cuidados também devem ser tomados regularmente para fazer cópias de backup das tabelas e armazená-las por causa da possibilidade de incêndios e terremotos que poderiam destruir o sistema.

Existem muitos SGBD sendo utilizados atualmente, porém, segundo uma visão de mercado, eles podem ser divididos em dois grandes grupos:

- Os comerciais, que cobram um valor de licença para serem usados;
- Os livres, que possuem distribuição livre, inclusive do código fonte, não implicando em nenhum custo adicional.

Segundo Heuser (2001) dentre os bancos de dados comerciais, três possuem um destaque maior devido a sua relevância: o Oracle, que é um dos mais famosos do mundo, sendo indicado principalmente para aplicações médias e grandes, onde a possibilidade de crescimento é muito importante; o Microsoft SQLServer, que vêm ganhando importância; e o DB2 da IBM, que começa a ganhar destaque.

Nos bancos de dados de distribuição livre se destacam o MySQL, uma das primeiras iniciativas na área, que possui um grande alcance no mercado e vem evoluindo a cada ano, e o

PostgreSQL, que pode ser utilizado desde aplicações pequenas até médias, com uma série de recursos que permite um crescimento razoável para soluções que a adotam. Vejamos um pouco mais sobre o MySQL, o SGBD que será utilizado na realização deste projeto.

Será analisado o MySQL por ser fácil de administrar e por exigir baixo recurso de software.

2.8 MySQL

Criado na Suécia, por David Axmark, Allan Larsson e o filandês Michael Widenius, iniciaram o projeto em 1980. É um sistema de gerenciamento de dados que usa a linguagem SQL (Linguagem de Consulta Estruturada, do inglês Structured Query Language) como interface (Medeiros, 2006).

Possui uma interface simples e está disponível para Sistema Operacional Open Source (Linux, MAC Os, etc) e para proprietários (Microsoft Windows), sendo conhecido por sua facilidade de uso, além de ser gratuito. É mais fácil de programar, com funções mais simples podendo ser modificado, voltado para aplicações de missão crítica de alto volume de dados. Considerado o mais rápido do mercado tornando-se uma alternativa barata e viável. Pode ser usado em aplicações para internet e desktop (Medeiros, 2006).

Segundo Date (1991) vejamos algumas características do MySQL:

- Portabilidade (suporta praticamente qualquer plataforma atual);
- Compatibilidade (existem drivers ODBC, JDBC e .NET e módulos de interface para diversas linguagens de programação, como Delphi, Java, C/C++, Python, Perl, PHP, ASP e Ruby)
- Excelente desempenho e estabilidade;
- Pouco exigente quanto a recursos de hardware;
- Facilidade de uso;
- É um Software Livre com base na GPL;
- Contempla a utilização de vários Storage Engines como MyISAM, InnoDB, Falcon, BDB, Archive, Federated, CSV, Solid...
- Suporta controle transacional;
- Suporta Triggers;
- Suporta Cursors (Non-Scrollable e Non-Updatable);

- Suporta Stored Procedures e Functions;
- Replicação facilmente configurável;
- Interfaces gráficas (MySQL Toolkit) de fácil utilização cedidos pela MySQL Inc.

O MySQL foi a opção escolhida, dentre outras, por proporcionar um rápido processamento de dados e um tempo curto de resposta sem exigir muito do hardware.

3. MATERIAL E MÉTODOS

3.1 Definição do projeto

O desenvolvimento da pesquisa se deu através da revisão bibliográfica, coleta de dados e levantamento de requisitos, com o objetivo de melhorar a gestão do negócio por meio do desenvolvimento de um software para cálculo das comissões de vendas de cursos e livros.

3.2 Coleta de dados

A coleta de dados é algo importante para poder compreender o sistema atual e compreender o que se espera do novo projeto. Segundo Mascarenhas (2012), o levantamento de requisitos funcionais e não funcionais nos auxiliam a colher informações úteis para o sistema.

Para o levantamento de dados a técnica utilizada foi a entrevista não estruturada realizada com a diretora comercial e os colaboradores da instituição com a finalidade de entender o processo atual e colher informações necessárias para criação do software as quais chamamos de requisitos.

Vejamos alguns pontos abordados:

- Quantidade de cursos vendidos por segmento;
- Valor recebido de comissão nos últimos 12 meses por cada colaborador;
- Média mensal de vendas por colaborador;
- Quantidade de cancelamentos por período;
- Relatórios específicos de vendas.

Esses são alguns pontos que poderão auxiliar a direção da empresa na tomada de decisão de forma eficaz, pois estará baseada em dados.

As vantagens e limitações desse tipo de entrevista segundo Gil (1999) são:

- Possibilita obtenção de dados referentes aos mais diversos aspectos da vida social;
- Permite a obtenção de dados em profundidade acerca do comportamento humano;
- Os dados obtidos podem ser classificados e quantificados.

- No exige que o respondente saiba ler ou escrever
- Maior flexibilidade no trabalho de investigação (pode-se explicar o significado das perguntas, captar expressões corporais, tonalidade de voz e ênfase das respostas)

Quanto às limitações foi observada falta de motivação do entrevistado para responder o questionário e pouco compromisso em relatar as respostas corretas. O vocabulário insuficiente e problemas psicológicos foram outro fator observado. O custo do treinamento dos entrevistadores e a aplicação das entrevistas também foram levados em conta.

A entrevista não estruturada foi a escolhida por dar maior liberdade ao entrevistado de expor o que conhece do sistema atual e proporcionar o entendimento do que ele espera do novo projeto.

3.3 Stakeholders/Usuários

Nome	Cargo	Interesse
Stakeholder 1	Diretora Comercial	Gerenciar as informações de vendas através de relatórios e fazer os lançamentos de produtos.
Stakeholder 2	Colaboradores	Operar o sistema alimentando o mesmo com os dados provenientes das operações diárias de vendas.

3.4 Pesquisa bibliográfica e definição de ferramentas

A seguir foi realizado o levantamento do material bibliográfico e a análise das ferramentas que serão utilizadas no projeto.

A linguagem de programação escolhida foi C#, totalmente orientada a objetos, fácil de aprender e utilizar, sendo robusta e com boa performance. O SGBD utilizado foi o MS SQL

para armazenar e fazer a manutenção dos dados devido à sua robustez, por ser enxuto e ainda gratuito.

4 RESULTADOS

Através desta análise e observando o objetivo em questão – a criação de um software para cálculo de comissões – notamos a existência de vários softwares neste segmento, mas nenhum capaz de suprir a necessidade específica desta empresa. O motivo é que as alíquotas de comissão são diferentes, os produtos não são os mesmos e os relatórios emitidos não estão de acordo com a necessidade.

O sistema será utilizado em um computador servidor e compartilhado com as máquinas clientes, utilizado por pessoas autorizadas e devidamente treinadas para seu uso. Ele fornecerá a essas pessoas as funções previamente definidas e permitidas para o usuário em questão.

Para o desenvolvimento utilizamos a linguagem C# e o banco de dados SQL. O banco de dados irá conter os nomes dos cursos, alíquotas, segmentos dos cursos, etc.

O software irá captar os dados das vendas e gerar relatórios de comissões de cada vendedor.

Ao escolher a forma de pagamento e o tipo (que é o segmento), o software irá usar uma alíquota pré-estabelecida para calcular a comissão que será lançada para o colaborador informado.

Poderá emitir relatórios por colaborador ou por período (diário, mensal ou anual) e também por cursos e segmentos.

Desta forma poderá auxiliar de forma específica e rápida no cálculo de comissões que antes eram feitos de forma manual.

4.1 Levantamento de requisitos

O levantamento de requisitos é uma área da Engenharia de Software que permite entender o que se espera do software. Através deles podemos compreender a descrição das funções e restrições que o software deve possuir e a partir daí analisar, documentar e verificar essas funções. Podem ser classificados como requisitos funcionais e não funcionais, vejamos abaixo.

4.2 Requisitos Funcionais do Sistema

São declarações de serviços que o sistema deve prover, descrevendo o que o sistema deve fazer.

Tabela 1 – Requisitos Funcionais

Código	Nome	Descrição
RF01	Autenticação	O sistema deverá dispor de tela de login, sendo permitido o acesso apenas a usuários previamente cadastrados.
RF02	Cancelamento Login	No frame de login o acesso poderá ser cancelado pelo usuário, suspendendo a execução do sistema.
RF03	Privilégios da diretora comercial	A diretora comercial terá acesso total ao sistema, bem como será a única autorizada a cadastrar novos usuários e produtos e excluir os já existentes.
RF04	Área de administração do sistema	A administração do sistema será disposta em um menu apenas à diretora comercial, onde haverá as funções de inclusão, exclusão e alteração de novos usuários e produtos.
RF05	Cadastro de cursos e livros	O sistema deverá permitir a Diretora comercial o cadastro de todos os cursos e livros
RF06	Cadastro de colaboradores	Cadastro de cada colaborador para que esteja disponível na tela de venda.
RF07	Cadastro de Vendas	O sistema deve permitir aos usuários a opção de efetuar o cadastro da venda.
RF8	Impressão	O sistema deverá permitir a impressão de relatórios de vendas, com nome do vendedor e período.

Fonte: o autor

Os requisitos citados foram baseados no levantamento feito com os *stakeholders* envolvidos.

4.3 Requisitos Não Funcionais do Sistema

Descrevem restrições sobre os serviços ou funções oferecidas pelo sistema.

Tabela 2 – Requisitos Não Funcionais

RNF01	Linguagem de programação	O sistema será desenvolvido em C# e o banco de dados padrão SQL Server.
RNF02	Sistema Operacional	O sistema será executado no sistema operacional Windows .
RNF03	Segurança de acesso	O sistema deverá ser acessado somente por usuários autorizados e cadastrados com senha de segurança média/alta.
RNF04	Tempo de resposta	As consultas e operações não deverão ultrapassar 10 segundos.
RNF05	Conclusão do projeto	O sistema deverá ser desenvolvido no máximo em dois meses, salvo imprevisto devidamente justificado.
RNF06	Banco de dados	O banco de dados será relacional e deverá atender as propriedades ACID e padrão SQL.
RNF07	Facilidade de uso	Os usuários deverão estar aptos a operar o sistema (suas principais funcionalidades) em no máximo 30 dias.

Fonte: o autor

Os requisitos não funcionais abordam aspectos de qualidade importantes em sistemas de software. Se tais requisitos não são levados em consideração, então o sistema de software poderá ser inconsistente e de baixa qualidade.

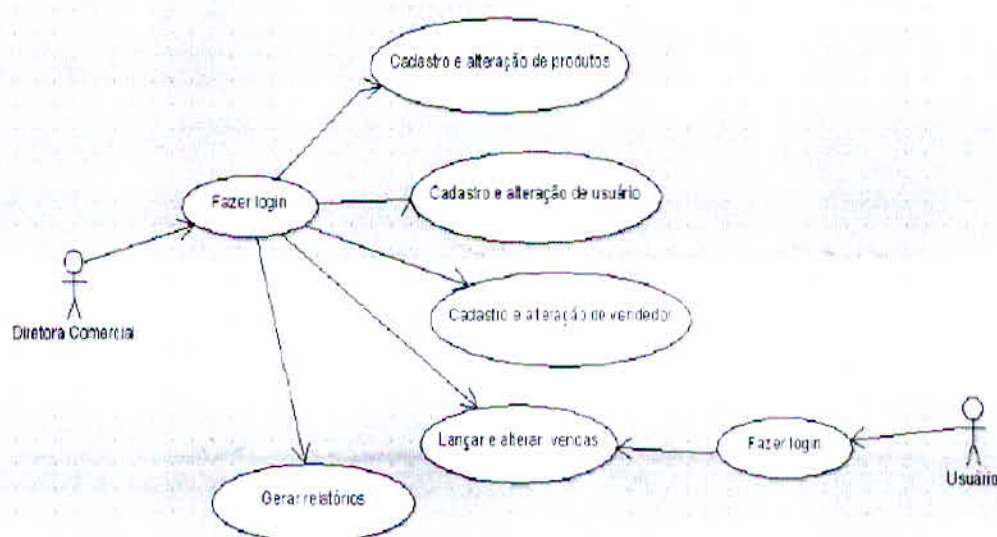
4.4 Modelagem e diagramas

Para Booch (2012), um diagrama de caso de uso captura um contrato entre os stakeholders (é um termo sem uma tradução consensual na comunidade brasileira e por isto foi mantido no original. Basicamente significa parte interessada – com interesses – envolvida no desenvolvimento do sistema) de um sistema sobre seu comportamento. O caso de uso descreve o comportamento do sistema sob diversas condições conforme o sistema responde a uma requisição de um dos stakeholders, chamado ator primário.

Para representar os casos de uso do software em questão utilizaremos o diagrama de Caso de Uso que é um artefato da UML. A UML ajuda a visualizar o produto que será construído e apoia a comunicação. É formada por elementos que são usados para criar diagramas, que podem representar uma determinada parte do sistema ou um ponto de vista do mesmo. (SILVA, 2014)

Abaixo segue o Caso de Uso do software em questão:

Figura 1 – Diagrama de Caso de Uso



Fonte: o autor

Nesse aspecto o diagrama de Casos de Uso modela a funcionalidade do sistema através de atores e casos de uso. Casos de uso são serviços ou funções fornecidas pelo sistema aos seus usuários. (SILVA, 2014)

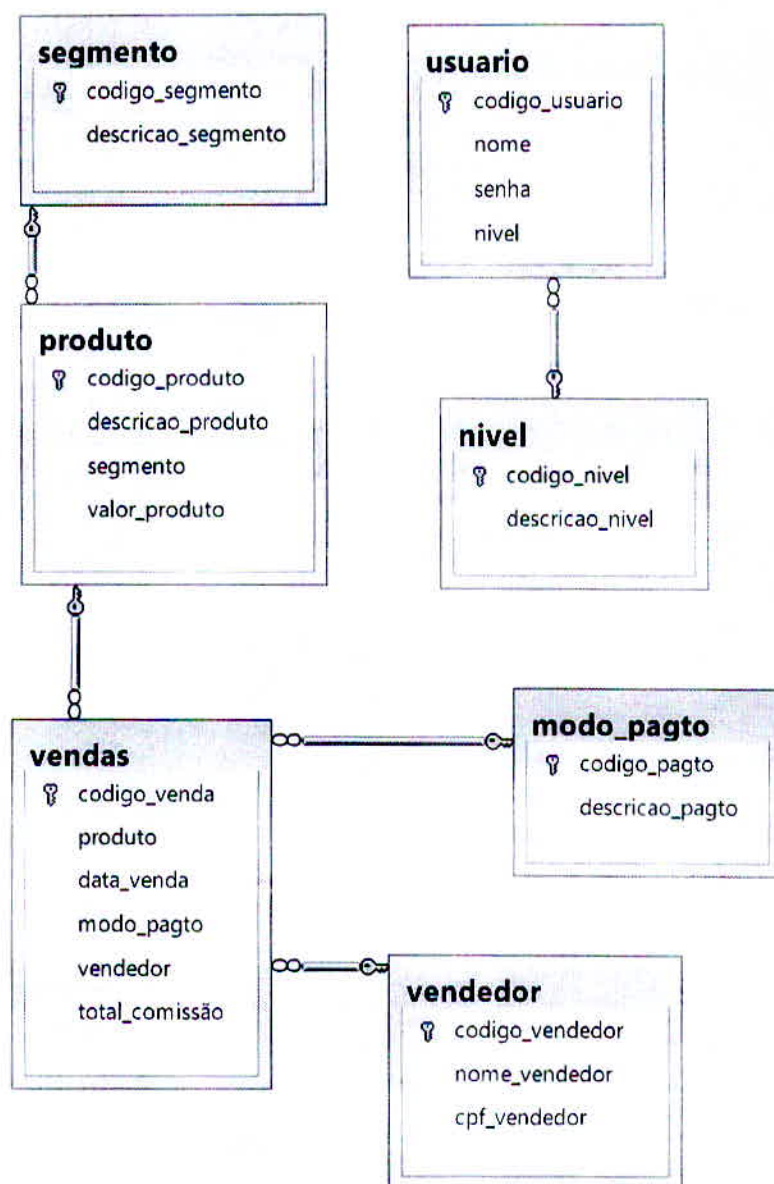
Ele no auxiliar na compreensão geral do sistema, entendendo suas ramificações e caminhos a serem seguidos.

4.5 Diagrama de entidade relacionamento

O Diagrama Entidade Relacionamento utiliza elementos gráficos para descrever o modelo de dados de um sistema com alto nível de abstração. O DER parte do princípio que o mundo real é formado por um conjunto de objetos chamados entidades e por relacionamentos entre esses objetos [Silberschatz et. al,1999].

Um DER é composto de entidades, relacionamentos, seus atributos e cardinalidades. As entidades representam um objeto ou conceito da realidade e os relacionamentos exibem qual a ligação entre elas, utilizando das cardinalidades para definir a sua interação.

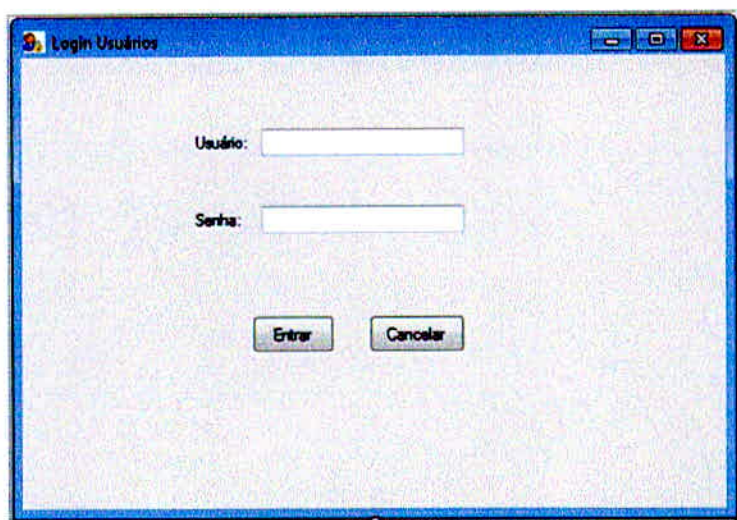
Figura 2 – Diagrama Entidade Relacionamento



Seu objetivo é facilitar o projeto de banco de dados por meio de um modelo independente de implementação e de fácil compreensão por parte do usuário

4.6 Resumo das Características do Sistema

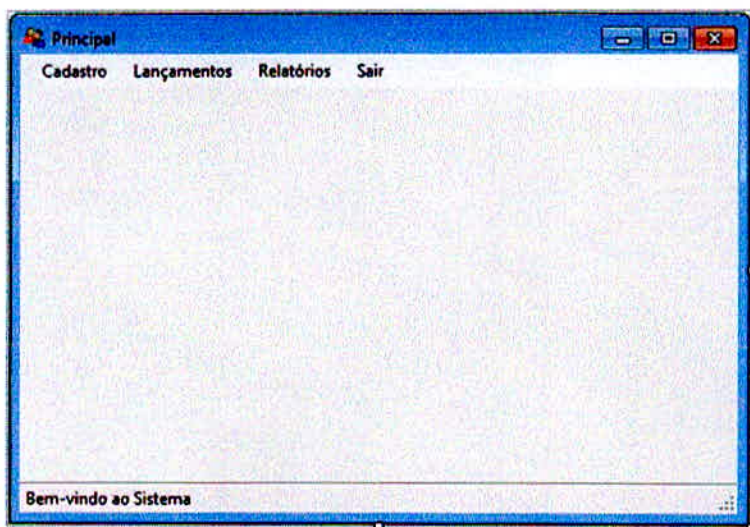
Para acessar o software o usuário terá que efetuar seu *login*, através de usuário e senha.

Figura 3 – Tela de Login

Fonte: o autor

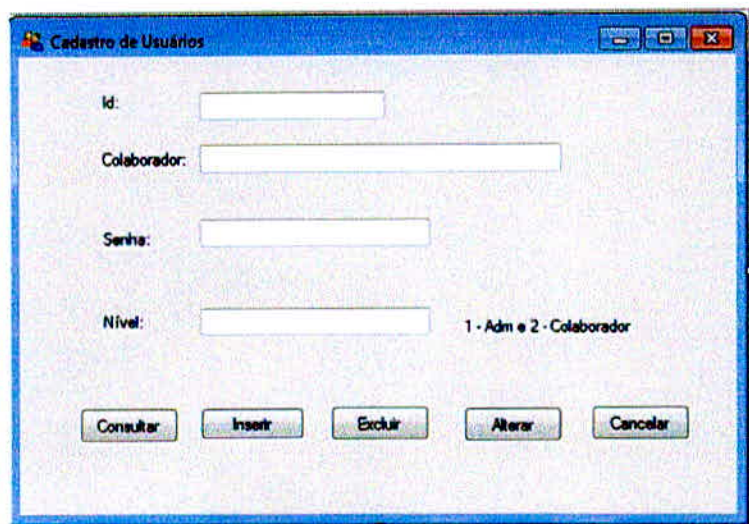
De acordo com o usuário cadastrado o mesmo terá ou não permissão para determinados menus.

A próxima tela é a principal onde ficam o menu de acesso para várias opções.

Figura 4 – Tela Principal

Fonte: o autor

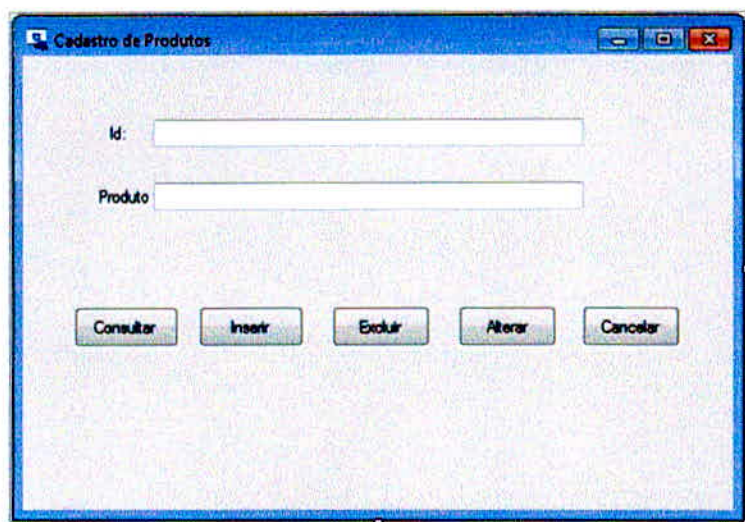
Os menus estão de acordo com o nível cadastrado para cada usuário.

Figura 5 – Tela de Cadastro de Usuários

Fonte: o autor

Possibilita a inserção dos usuários bem como seu nível de acesso ao sistema. Pode-se também consultar, alterar e excluir o cadastro do usuário.

Na tela de cadastro de produtos podem ser inseridos os cursos e livros.

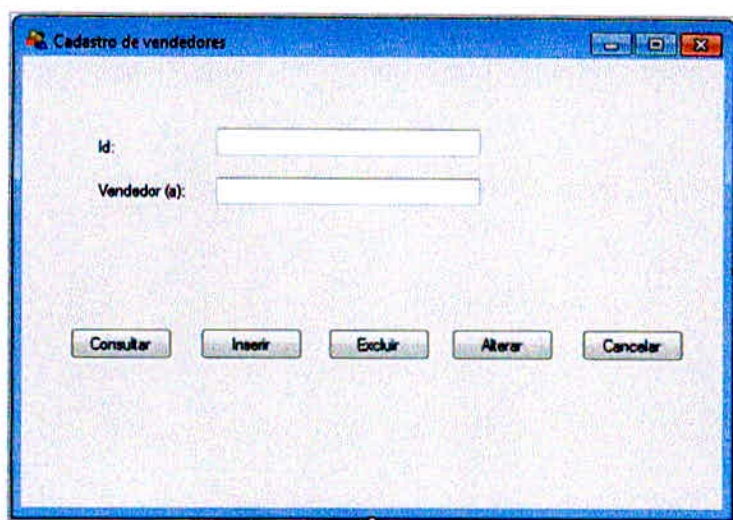
Figura 6 – Tela de Cadastro de Produtos

Fonte: o autor

Podem ser feitas consultas, alterações e exclusões.

Foi desenvolvida uma tela para cadastro de vendedores, que permite também a consulta e exclusão.

Figura 7 – Tela de Cadastro de Vendedores

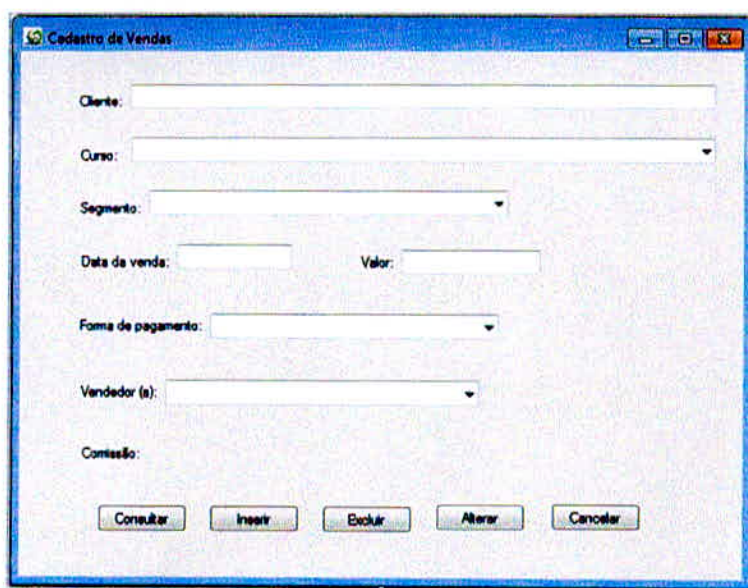


The screenshot shows a window titled "Cadastro de vendedores". It contains two text input fields: "Id:" and "Vendedor (a):". Below these fields are five buttons: "Consultar", "Inserir", "Excluir", "Alterar", and "Cancelar".

Fonte: o autor

Esta é a tela de lançamento das vendas onde foi feita uma programação utilizando o segmento e forma de pagamento para gerar o valor da comissão paga pelo curso vendido.

Figura 8 – Tela de Lançamento das Vendas

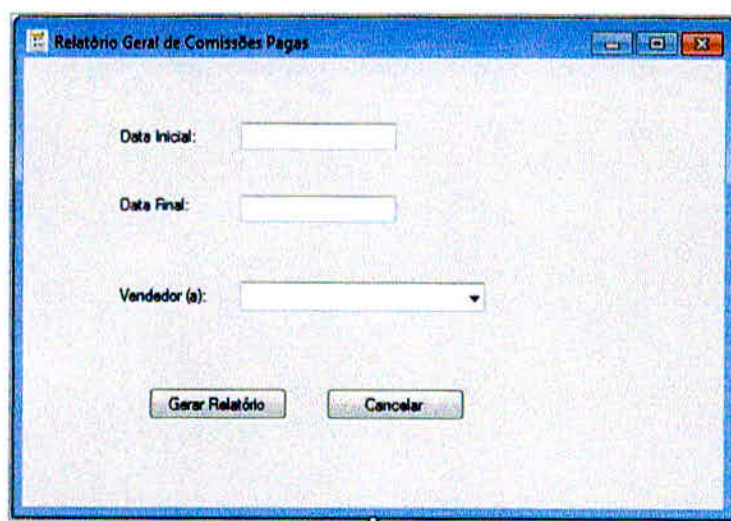


The screenshot shows a window titled "Cadastro de Vendas". It contains several input fields: "Cliente:" (text), "Curso:" (dropdown), "Segmento:" (dropdown), "Data da venda:" (text), "Valor:" (text), "Forma de pagamento:" (dropdown), "Vendedor (a):" (dropdown), and "Comissão:" (text). Below these fields are five buttons: "Consultar", "Inserir", "Excluir", "Alterar", and "Cancelar".

Fonte: o autor.

Nesta tela pode-se gerar o relatório de vendas, totalizando o valor por vendedor.

Figura 9 – Tela de Gerar Relatório

A imagem mostra uma janela de aplicativo com o título "Relatório Geral de Comissões Pagas". O layout da janela é simples, com um fundo branco e uma borda azul. No topo, há uma barra de título com ícones de minimizar, maximizar e fechar. O conteúdo principal da janela contém três linhas de entrada de dados. A primeira linha é rotulada "Data Inicial:" e possui um campo de texto branco. A segunda linha é rotulada "Data Final:" e também possui um campo de texto branco. A terceira linha é rotulada "Vendedor (a):" e possui um menu suspenso com uma seta para baixo no final. Abaixo desses campos, há dois botões retangulares com o texto "Gerar Relatório" e "Cancelar".

Fonte: o autor.

Para que o sistema atenda as necessidades da empresa foi necessário um projeto organizado do mesmo. A primeira tarefa que foi a compreensão do problema, pois segundo Silva (2014), é preciso obter um melhor entendimento antes do início do desenvolvimento da solução, isto irá nos ajudar a encontrar a solução correta.

No desenvolvimento de software temos dois espaços: o espaço do problema e o espaço da solução. Ao focarmos nossos esforços no espaço do problema teremos condições de identificar as necessidades dos *stakeholders*/negócio, também chamadas de *Needs*.(SILVA, 2014)

Após identificarmos as *Needs*, ou seja, as necessidades do negócio e dos *stakeholders*, é o momento de focarmos nossos esforços no espaço da solução. Neste momento teremos condições de identificar as características do software, também chamadas de *Features*. As *Features* podem ser identificadas como “serviços” que a solução deverá oferecer para atender as *Needs*(necessidades) do negócio/*stakeholders*.(SILVA, 2014)

Esses “serviços” ou características segundo Wesley(2000), não são, normalmente, bem definidos e podem até, serem conflitantes entre si, mas são, no entanto, uma representação das reais necessidades.

Após a identificação das *Features* da solução, teremos condições de identificar os requisitos. Os requisitos são derivados das *Features*. Para cada *Feature* identificada, temos um ou mais requisitos de software, que irão tratar em maiores detalhes como as *Features* serão atendidas. Da mesma forma que cada *Feature* deverá atender um ou mais *Need*, cada requisito deverá atender uma ou mais *Feature*.(SILVA, 2014)

Os requisitos podem ser divididos em Requisitos Funcionais (RF) e Requisitos Não funcionais (RNF). Iremos nos ater aos Requisitos Funcionais, pois eles descrevem as funcionalidades que o software deverá oferecer, ou seja, o comportamento e a informação que a solução irá gerenciar. (SILVA, 2014)

Os Requisitos Funcionais podem ainda conter Regras de Negócio (RN) que são definidas pelo RUP¹ como declarações sobre políticas ou condições que devem ser satisfeitas. Uma regra de negócio é pertencente a um negócio, ou uma organização e não a um sistema, ou seja, ela provém da atividade exercida pela organização.

Abaixo segue a relação de *Needs*, *Features*, Requisitos e Regras de Negócio do sistema desenvolvido para a empresa do setor educacional:

Tabela 3 - Needs

NEEDS	
ID	Descrição
NE001	Controlar informações de vendas/comissões de forma eficiente e ágil.
NE002	Gerar relatórios que auxiliem a gestão.
NE003	Guardar as informações com segurança.

Fonte: o autor.

Tabela 4 - Features

FEATURES	
ID	Descrição
FE001	Manter Produtos
FE002	Manter Vendedores
FE003	Manter usuários do sistema.
FE004	Controlar acesso do sistema.
FE005	Manter Vendas

¹ RUP – Rational Unified Process ou Processo Unificado Rational é um processo proprietário de Engenharia de Software criado pela Rational Software Corporation.

Fonte: o autor.

Tabela 5 – Regras de Negócio

REGRAS DE NEGÓCIO	
ID	Descrição
RN001	Os novos usuários poderão ser cadastrados somente pelo administrador do sistema.
RN002	Os usuários somente poderão ser alterados pelos administradores do sistema ou pelo próprio usuário.
RN003	Os usuários somente poderão ser excluídos do sistema pelo administrador.
RN04	Os usuários somente poderão acessar o sistema se informarem um login e uma senha válidos e tenham permissão de acesso no sistema.

Fonte: o autor.

5 CONCLUSÃO

Este projeto teve como objetivo a criação de um software para cálculo de comissão de vendas de cursos e livros de uma instituição de educação, com o objetivo de informatizar os trabalhos manuais.

As instituições de educação têm crescido no Brasil e a concorrência aumentado, fazendo que busquem mecanismos que melhorem a gestão. Este software visa aprimorar e melhorar os processos, agilizando e dando um retorno mais rápido para a tomada de decisão dos gestores. Através do cadastro da venda e geração de relatórios isso será possível. Podemos encará-lo como um diferencial competitivo, pois auxiliará de maneira mais rápida nas decisões da gestão.

Para o desenvolvimento do sistema foram utilizados conceitos de Sistemas de Informação e Engenharia de Software, contando também com as facilidades do ambiente de desenvolvimento Visual Basic, da robustez do banco de dados MySQL e sempre tentando aliar boas práticas de programação. Procurou-se dar ênfase no processo de criação do software e no cumprimento dos requisitos apontados quando do início do projeto.

Antes da implementação do software todo processo de lançamento das vendas e cálculo de comissões era feito manualmente, o que demandava tempo e não era totalmente confiável.

Pode-se observar que o software está apenas em uma fase inicial, podendo ainda ser melhorado, implementando a opção de cancelamento de vendas e a de controle de log.

REFERÊNCIAS

- AULER, M. **Lições de um professor rebelde**. *Administração & Marketing*, v3 n.24, p.23-26, set. 1986. In SILVEIRA, Amélia, COLOSSI, Nelson, SOUZA, Cláudia Gonçalves (org). *Administração Universitária*, Florianópolis, 1988, p.28.
- BOOCH, Grady et al. **UML – Guia do Usuário**, 2ª Edição. Elsevier Editora Ltda, Rio de Janeiro, 2012.
- BURROWS, Chris. **New C# Features in the .NET Framework 4**, Junho 2010.
- DATE, C.J.; *Int. a Sistemas de Bancos de Dados*, tradução da 4a.edição norte-americana, Editora Campus, 1991.
- DATE, C.J. **Introdução a Sistema de Banco de Dados**. Traduzida da 7ª Ed. americana. Rio de Janeiro: Campus, 2000.
- DON Box e Chris Sells. **Essential .NET: Volume 1 - The Common Language Runtime**. Addison-Wesley Pearson Education, Boston, 2003.
- FARRER H. et ali. **Algoritmos Estruturados**. Editora Guanabara S. A., Rio de Janeiro, 1985.
- GIL, A. C. **Métodos e técnicas de pesquisa social**. 4. ed. São Paulo: Atlas, 1994. 207p.
- HEUSER C. A. **Projeto de Banco de Dados**. 3ª Edição. Porto. Alegre: Sagra-Luzzato, 2001.
- KORTH, H.F. e Silberschatz, A.; **Sistemas de Bancos de Dados**, Makron Books, 2a. edição revisada, 1994.
- LAUDON, K. C. e LAUDON, J. P. **Sistemas de informação**, Rio de Janeiro: Livros Técnicos e Científicos, 3ª. edição, 1999.
- LIMA, Edwin; REIS, Eugênio. **C# e .NET para desenvolvedores**. Rio de Janeiro: Campus, 2002.
- MACHADO, Felipe Nery Rodrigues e ABREU, Maurício Pereira de. **Projeto de Banco de Dados: Uma Visão Prática**. São Paulo: Érica, 1996.
- MAYA, Alcides. **Linguagem de programação I (linguagem C#)**. Disponível em: < www.alcidesmaya.com.br/apostilas/linguagem_c.pdf > Acesso em: 29 set 2015.
- MASCARENHAS, Sidnei Augusto, **Metodologia Científica**, 1ª Ed. São Paulo: Pearson Education do Brasil, 2012.
- MEDEIROS, M. **Banco de dados para sistemas de informação**. Florianópolis: Visual Books, 2006.

MSDN. **O que há de novo na linguagem e no compilador C# 2.0**, 2005.

MSDN. **Overview of C# 3.0**. Anders Hejlsberg, Mads Torgersen, 2007.

MSDN. Introdução à linguagem C# e o .NET Framework, 2012.

ROWLEY, J. **Informática para bibliotecas**. Tradução de Antônio Agenor Briquet de Lemos. 3.ed. Brasília : Briquet de Lemos/Livros, 1994.

SILBERSCHATZ, A.; Korth, H.; Sudarshan, S. **Sistema de Banco de Dados**, 3. ed. Makron Books, 1999.

SILVA, Rodrigo Gomes da. **Guia de Estudos: Introdução à Engenharia de Requisitos**. [S.l.: s.n], Varginha: UNIS, 2014.

VELLOSO, Fernando de Castro. **Informática: Conceitos Básicos**. 6. ed. Rio de Janeiro: Campus LTDA, 2003.

WESLEY, Addison. Livro: **Gerenciamento de Requisitos de Software**. [S.l.: s.n], 2000.