

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

**EDUARDO PACHECO
JULIO CESAR BOFF**

**DESENVOLVIMENTO DE UM SISTEMA GERENCIAL PARA
EMPRESAS VAREJISTAS**

TRABALHO DE CONCLUSÃO DE CURSO

**PATO BRANCO
2013**

**EDUARDO PACHECO
JULIO CESAR BOFF**

**DESENVOLVIMENTO DE UM SISTEMA GERENCIAL PARA
EMPRESAS VAREJISTAS**

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

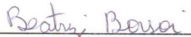
Orientadora: Beatriz Terezinha Borsoi

**PATO BRANCO
2013**

ATA Nº: 218

DEFESA PÚBLICA DO TRABALHO DE DIPLOMAÇÃO DO ALUNO **EDUARDO PACHECO**.

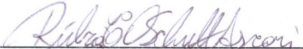
Às 17:03 hrs do dia 19 de setembro de 2013, Bloco V da UTFPR, Câmpus Pato Branco, reuniu-se a banca avaliadora composta pelos professores Beatriz Terezinha Borsoi (Orientadora), Andréia Scariot Beulke (Convidada) e Rúbia E. O. Schultz Ascari (Convidada), para avaliar o Trabalho de Diplomação do aluno Eduardo Pacheco, matrícula 1211030, sob o título **Desenvolvimento de um sSstema Gerencial para Empresas Varejistas**; como requisito final para a conclusão da disciplina Trabalho de Diplomação do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, COADS. Após a apresentação o candidato foi entrevistado pela banca examinadora, e a palavra foi aberta ao público. Em seguida, a banca reuniu-se para deliberar considerando o trabalho **APROVADO**. Às 17:30 hrs foi encerrada a sessão.



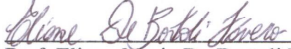
Profa. Beatriz Terezinha Borsoi, Dr.
Orientadora



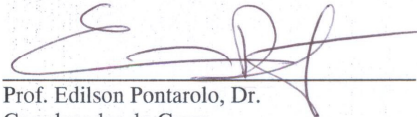
Profa. Andréia Scariot Beulke, Esp.
Convidada



Profa. Rúbia E. O. Schultz Ascari, M.Sc.
Convidada



Prof. Eliane Maria De Bortoli Fávero, M.Sc.
Coordenador do Trabalho de Diplomação

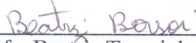


Prof. Edilson Pontarolo, Dr.
Coordenador do Curso


ATA Nº: 219

DEFESA PÚBLICA DO TRABALHO DE DIPLOMAÇÃO DO ALUNO **JULIO CESAR BOFF**.

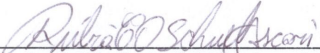
Às 17:03 hrs do dia 19 de setembro de 2013, Bloco V da UTFPR, Câmpus Pato Branco, reuniu-se a banca avaliadora composta pelos professores Beatriz Terezinha Borsoi (Orientadora), Andréia Scariot Beulke (Convidada) e Rúbia E. O. Schultz Ascari (Convidada), para avaliar o Trabalho de Diplomação do aluno Julio Cesar Boff, matrícula 1209396, sob o título **Desenvolvimento de um Sistema Gerencial para Empresas Varejistas**; como requisito final para a conclusão da disciplina Trabalho de Diplomação do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, COADS. Após a apresentação o candidato foi entrevistado pela banca examinadora, e a palavra foi aberta ao público. Em seguida, a banca reuniu-se para deliberar considerando o trabalho **APROVADO**. Às 17:30 hrs foi encerrada a sessão.



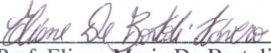
Profa. Beatriz Terezinha Borsoi, Dr.
Orientadora



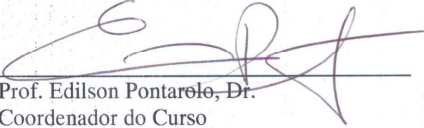
Profa. Andréia Scariot Beulke, Esp.
Convidada



Profa. Rúbia E. O. Schultz Ascari, M.Sc.
Convidada



Prof. Eliane Maria De Bortoli Fávero, M.Sc.
Coordenador do Trabalho de Diplomação



Prof. Edilson Pontarelo, Dr.
Coordenador do Curso

RESUMO

PACHECO, Eduardo; BOFF, Julio Cesar. Desenvolvimento de um sistema gerencial para empresas varejistas. 2013. 68 f. Monografia de Trabalho de Conclusão de Curso - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2013.

Lojas varejistas se caracterizam principalmente pela realização de vendas de produtos diretamente ao consumidor final. Esse processo é chamado de vendas a varejo. Um grande número de vendas acarreta a necessidade de um sistema gerencial para controle financeiro, logístico e comercial. Empresas informatizadas têm condições melhores para atender seus clientes devido a agilidade, segurança no momento da venda, histórico das operações, flexibilidade de preços no momento da venda, impressão de condicionais entre outros. As empresas varejistas na maioria se enquadram como micro e pequenas empresas, mas elas têm necessidade de segurança das operações e informações como as grandes empresas. Neste trabalho é reportada a análise e desenvolvimento de um sistema gerencial para empresas varejistas com uso da linguagem de programação Delphi. Também foram utilizadas a ferramenta de modelagem DeSign for Database, o sistema gerenciador de banco de dados Firebird e a ferramenta administrativa de banco de dados IBExpert.

Palavras-chave: Sistema gerencial. Firebird. IBExpert. Linguagem de programação Delphi. Sistema *desktop*.

ABSTRACT

PACHECO, Eduardo; BOFF, Julio Cesar. Development of a management system for retail companies. 2013. 68 f. Monografia de Trabalho de Conclusão de Curso - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2013.

Retail stores are characterized by the sales of products directly to the end consumer, where the wide number of sales provides the need for a management system to manage them financially, logistically and commercially. Computerized companies have better conditions to meet its customers due to agility, security at the time of sale and pricing flexibility insurance, history of operations, conditional printing among others. The retailers mostly qualify as micro and small businesses, or small, that needs security operations and information as large companies requiring solutions easy to use and with an attractive cost. This paper reported the analysis and development of a management system for retailers, where the programming language chosen was Delphi, which can provide development resources relevant to the proposed objectives could be achieved. We also used the modeling tool Dezin for Database, the system manager Firebird database administration tool and database IBEExpert.

Key-words: Management system. Firebird database. IBEExpert. Delphi Programming Language. Desktop system.

LISTA DE FIGURAS

Figura 1 - Tela principal da Ferramenta Visual Paradigm	19
Figura 2 - Tela principal da ferramenta DeZign for Databases.....	20
Figura 3 - Ambiente de Trabalho da Ferramenta Administrativa IBExpert	23
Figura 4 - Ambiente Integrado de Desenvolvimento Delphi XE3.....	24
Figura 5 - Tela Principal Cooperative System	27
Figura 6 - Tela de Classificação Cooperative System.....	27
Figura 7 - Tela Principal Point System	28
Figura 8 - Tela Geração Cupom Point System.....	28
Figura 9 - Formulário padrão de cadastro	29
Figura 10 - Formulário padrão de pesquisa	30
Figura 11 - Diagrama de Casos de Uso	36
Figura 12 - Diagrama entidades e relacionamentos.....	39
Figura 13 - Cadastro de Clientes.....	51
Figura 14 - Pesquisa de Clientes	53
Figura 15 - Pesquisa de Pessoas.....	54
Figura 16 - Cadastro de Pessoas.....	55
Figura 17 - Lançamentos de Pré-Venda	57
Figura 18 - Pesquisa de Pessoas na Pré-Venda.....	59
Figura 19 - Pesquisa de Produtos na Pré-Venda.....	59
Figura 20 - Relatório de Pré-Venda.....	60
Figura 21 - Formulário de Backup	61
Figura 22 - Formulário de Restauração de Dados	62

LISTA DE QUADROS

Quadro 1 - Requisitos do sistema	35
Quadro 2 - Requisitos não funcionais	37
Quadro 3 - Caso de uso registrar pré-venda	38
Quadro 4 - Cadastro de cadeias (tbCadeias)	40
Quadro 5 - Cadastro de cargos (tbCargos)	40
Quadro 6 - Cidades (tbCidades).....	40
Quadro 7 - Cadastro de clientes (tbClientes)	41
Quadro 8 - Controle de terminais (tbControleTerminais).....	41
Quadro 9 - Cadastro de convênios (tbConvenios)	41
Quadro 10 - Cadastro dos Dados Contábeis (tbDadosContabeis).....	42
Quadro 11 - Dados da Empresa (tbEmpresas)	42
Quadro 12 - Data do Fechamento do Movimento Diário (tbFechamento).....	42
Quadro 13 - Cadastro de Fornecedores (tbFornecedores)	43
Quadro 14 - Tabela de vinculo fornecedor/produto (tbFornProd).....	43
Quadro 15 - Cadastro de Funcionários (tbFuncionarios)	43
Quadro 16 - Cadastro das grades tributárias dos produtos (tbGrades).....	44
Quadro 17 - Cadastro de Grupos (tbGrupos)	44
Quadro 18 - Cadastro do ICMS (tbICMS)	44
Quadro 19 - Cadastro do IPI (tbIPI).....	44
Quadro 20 - Itens da pré-vendas (tbItemPreVenda)	45
Quadro 21 - Cadastro de Marcas (tbMarcas)	45
Quadro 22 - Tabela dos Modelos Fiscais (tbModelosFiscais).....	45
Quadro 23 - Cadastro do NCM (tbNCMSH)	45
Quadro 24 - Cadastro de Pessoas (tbPessoas)	46
Quadro 25 - Cadastro de PIS/COFINS (tbPISCOFINS).....	46
Quadro 26 - Dado das Pré-Vendas (tbPreVenda).....	46
Quadro 27 - Cadastro de Produtos (tbProdutos).....	47
Quadro 28 - Cadastro de Representantes (tbRepresentantes).....	47
Quadro 29 - Tabela de Séries fiscais (tbSeriesFiscais).....	47
Quadro 30 - Cadastro de Subgrupos (tbSubGrupos)	48
Quadro 31 - Tabela dos Tipos de Produtos (tbTipoProduto).....	48
Quadro 32 - Tabela de UF (tbUF)	48
Quadro 33 - Tabela de Unidades de Medidas (tbUN)	48
Quadro 34 - Cadastro de Usuário (tbUsuarios)	49
Quadro 35 - Tabela de Versões do Sistema e Banco de Dados (tbVersao)	49
Quadro 36 - Relação das funcionalidades implementadas	50

LISTA DE ABREVIATURAS E SIGLAS

CNAE	Classificação Nacional de Atividades Econômicas
CNPJ	Cadastro Nacional de Pessoas Jurídicas
COFINS	Contribuição para o Financiamento da Seguridade Social
CORBA	<i>Common Object Request Broker Architecture</i>
CST	Código de Situação Tributária
DBMS	<i>Data Base Management System</i>
DDL	<i>Data Definition Language</i>
IBGE	Instituto Brasileiro de Geografia e Estatística
HTML	<i>HiperText Markup Language</i>
ICMS	Imposto sobre Circulação de Mercadorias e Serviços
IDE	<i>Integrated Development Environment</i>
IDL	<i>Interactive Data Language</i>
IPI	Imposto sobre Produtos Industrializados
NCM	Nomenclatura Comum do Mercosul
PDF	<i>Portable Document Format</i>
PIS	Programa de Integração Social
RAD	<i>Rapid Application Development</i>
SGBD	Sistema Gerenciador de Banco de Dados
SQL	<i>Structured Query Language</i>
SysML	<i>Systems Modeling Language</i>
UML	<i>Unified Modeling Language</i>
VCL	<i>Visual Component Library</i>
XML	<i>Extensible Markup Language</i>

SUMÁRIO

1 INTRODUÇÃO	10
1.1 CONSIDERAÇÕES INICIAIS	10
1.2 OBJETIVOS	11
1.2.1 Objetivo Geral	11
1.2.2 Objetivos Específicos	11
1.3 JUSTIFICATIVA	12
1.4 ESTRUTURA DO TRABALHO	13
2 REFERENCIAL TEÓRICO	14
2.1 BANCO DE DADOS	14
2.2 DESENVOLVIMENTO DE SOFTWARE <i>DESKTOP</i>	16
2.3 SISTEMAS DE INFORMAÇÃO	16
3 MATERIAIS E MÉTODO	18
3.1 MATERIAIS	18
3.1.1 Visual Paradigm	18
3.1.2 DeZign For Database	20
3.1.3 Banco de Dados Firebird	22
3.1.4 Ferramenta Administrativa IExpert	22
3.1.5 Linguagem de Programação Delphi	23
3.2 MÉTODO	26
3.2.1 Estudos de Casos em Delphi	26
3.2.2 Formulários Padrão	29
4 RESULTADO	31
4.1 APRESENTAÇÃO DO SISTEMA	31
4.2 MODELAGEM DO SISTEMA	32
4.3 APRESENTAÇÃO DO SISTEMA	49
4.4 IMPLEMENTAÇÃO DO SISTEMA	62
5 CONCLUSÃO	67
REFERÊNCIAS	68

1 INTRODUÇÃO

Este capítulo apresenta as considerações iniciais do trabalho, abrangendo uma visão geral do sistema implementado como resultado da realização deste trabalho, os objetivos, a justificativa de realização do mesmo e a organização do texto.

1.1 CONSIDERAÇÕES INICIAIS

O número de micro e pequenas empresas existentes no Brasil é bastante elevado. Dados de 2010 indicam que 99% das empresas do país são caracterizadas como micro e pequenas e que essas empresas são responsáveis por 60% dos empregos (BRASIL, 2013).

Esses números justificam a necessidade que essas empresas sejam gerenciadas com o auxílio de sistemas informatizados, possibilitando que elas tenham um controle mais efetivo de suas receitas e despesas. Contudo, é importante que esses sistemas tenham interface que facilite o seu uso, que os dados sejam apresentados de forma clara e que não alterem significativamente a forma de realização das atividades e dos processos de negócio realizados manualmente. Dessa forma, será mais fácil as pessoas aceitarem o uso desses sistemas e o aprendizado de uso será mais rápido, agilizando a transição entre o processo manual e informatizado.

Verificou-se, assim, a possibilidade de desenvolver um aplicativo computacional para esse segmento de mercado que são micro e pequenas empresas do comércio. O sistema desenvolvido como resultado deste trabalho implementa as funcionalidades essenciais para esse tipo de empresa. Em versões posteriores outras funcionalidades serão implementadas, como, por exemplo, a emissão de nota fiscal eletrônica.

Como ambiente de programação será utilizado o Delphi XE3. A escolha é decorrente da grande quantidade de componentes e recursos disponibilizados pela ferramenta e também de componentes de terceiros que são gratuitos. Essa

linguagem também facilita a interligação com os vários periféricos utilizados nesse nicho de mercado, como, por exemplo, impressoras de cupom fiscal e leitores de código de barras.

1.2 OBJETIVOS

Este trabalho de conclusão de curso se refere ao desenvolvimento de software e a seguir serão descritos seu objetivo geral e seus objetos específicos. O objetivo geral está relacionado ao resultado principal obtido com o desenvolvimento deste trabalho e os objetivos específicos o complementam.

1.2.1 Objetivo Geral

Desenvolver um software gerencial para micro e pequenas empresas do setor varejista que permita realizar todo o controle de movimentação de produtos, focando o processo de pré-vendas.

1.2.2 Objetivos Específicos

Como forma de complementar o objetivo geral foram definidos os seguintes objetivos específicos:

- Desenvolver um software gerencial para o varejo com as seguintes funcionalidades:
 - Cadastros base do sistema: cadeias, cargos, pessoas, clientes, fornecedores, funcionários, usuários, produtos, grupos e marcas.
 - Grade Tributária para fins fiscais estaduais e federais.

- Cópia de segurança: formulário para realizar *backup* e recuperação do mesmo.
- Movimentações: geração de pré-venda, processamento e impressão.
- Relatórios: apresentar as informações dos produtos, fornecedores, clientes e pré-vendas.

1.3 JUSTIFICATIVA

O software para gerenciar empresas varejistas de pequeno e médio porte possibilitará ao usuário realizar o controle fiscal e comercial, dentre outras funcionalidades do negócio. Por meio do aplicativo desenvolvido todo o processo de venda de produtos passará pela pré-venda, gerando venda condicional ou avulsa. A qualquer momento um relatório referente às movimentações poderá ser gerado, tanto como de outros dados do sistema que podem auxiliar na tomada de decisões e para negociações.

Os dados dos cadastros são armazenados à medida que são inseridos. Esse procedimento facilita a recuperação nos casos de falta de energia, por exemplo, quando uma venda está sendo realizada ou outros cadastros do sistema estão em processo de preenchimento. No momento que o sistema é reinicializado o cadastro que estava em preenchimento é rerepresentado com os dados que já haviam sido informados.

Para banco de dados foi escolhido o Firebird devido a essa tecnologia possuir recursos considerados suficientes para o aplicativo desenvolvido e por ser *open source* (código aberto). A ferramenta de administração desse banco de dados é o IBExpert e facilita a tarefa de administração do banco de dados. O Firebird é um banco de dados de fácil compatibilidade com diversas linguagens. Além disso, apresenta recursos de segurança importantes e necessários para qualquer sistema computacional.

1.4 ESTRUTURA DO TRABALHO

O Capítulo 1 apresenta as considerações iniciais, os objetivos e a justificativa para o desenvolvimento deste trabalho. No Capítulo 2 está o referencial teórico apresentando conceitos relacionados a banco de dados, desenvolvimento de software *desktop* e sistemas de informação. O Capítulo 3 apresenta as ferramentas e o método utilizados no desenvolvimento do trabalho juntamente com suas características. No capítulo 4 estão os resultados. Esse capítulo contém a apresentação do sistema, a modelagem do mesmo por meio da descrição das tabelas de dados, diagrama de entidade e relacionamento, a descrição do sistema e a descrição das funcionalidades implementadas com exemplos de códigos. Por fim, no Capítulo 5 está a conclusão com as considerações finais.

2 REFERENCIAL TEÓRICO

Este capítulo contém o referencial teórico do trabalho apresentando conceitos sobre banco de dados, desenvolvimento de aplicações para ambiente *desktop* e sistemas de informação.

2.1 BANCO DE DADOS

Banco de dados é um conjunto de dados organizados e relacionados entre si de maneira a ter sentido para o contexto ao qual os dados se referem. Por exemplo, um banco de dados para um consultório médico contém dados sobre pacientes, consultas e agendamentos. Um banco de dados de um sistema de uma empresa de comércio contém dados sobre clientes, fornecedores, estoque, contas a pagar e a receber, dentre outros. Esses dados podem ser organizados em um conjunto de arquivos que armazenam algum tipo de informação de forma organizada para serem usadas por meios de leituras, inclusões, alterações e exclusões. Um banco de dados é uma estrutura lógica que tem como objetivo o armazenamento de dados visando reduzir a inconsistência e a redundância de informações.

“Dado é qualquer elemento identificado em sua forma bruta que, por si só, não conduz a compreensão de terminado fato ou situação.” “Informação é o dado trabalhado que permite ao executivo tomar uma decisão.” (OLIVEIRA, 2010, p.24).

Um Sistema Gerenciador de Banco de Dados (SGBD) tem como princípio auxiliar e facilitar o processo de desenvolvimento de software. Isso porque oferece uma interface que permite criar o banco, as tabelas e os seus campos com mais facilidade, geralmente de forma gráfica. Esses sistemas também facilitam a implementação porque permitem acesso aos dados armazenados agilizando testes e verificação das operações sendo realizadas.

Um SGBD é constituído por um conjunto de dados associados a programas para acessos a esses dados (SILBERSCHATZ, KORTH, SUDARSHAN, 1999). É um dos sistemas de software mais importantes de todo o sistema de banco de dados,

acompanhado de outras ferramentas como utilitários, geradores de relatórios, gerador de transações, entre outros, afirma Date (2000).

Silberschatz, Korth e Sudarshan (1999) indicam três níveis de abstração para a modelagem de um banco de dados:

a) Nível físico – descreve como esses dados estão armazenados. As estruturas de dados em nível mais elementar são descritas em detalhes.

b) Nível lógico – descreve quais dados estão armazenados no banco e os relacionamentos entre os mesmos. Esse nível é utilizado por administradores de banco de dados que precisam decidir quais dados devem pertencer ao banco.

c) Nível visão – descreve apenas parte do banco de dados. Assim, um mesmo banco de dados pode ser representado por diversas visões que são definidas de acordo com interesses das equipes envolvidas no desenvolvimento e mesmo do cliente.

A representação do nível lógico e as visões são realizadas por meio de modelos de dados e esses modelos podem ser (SILBERSCHATZ, KORTH, SUDARSHAN, 1999; TAKAI, ITALIANO, FERREIRA, 2005): hierárquico, em redes, relacional, orientado a objetos, semântico e funcional. O modelo relacional também denominado de entidades e relacionamentos é o modelo utilizado para a definição das tabelas, campos e relacionamentos do banco de dados.

O modelo de dados relacional tem como base a teoria dos conjuntos e a álgebra relacional e a estrutura fundamental desse modelo é a relação (tabela) (TAKAI, ITALIANO, FERREIRA, 2005). Uma relação é constituída por um ou mais atributos (campos) que traduzem o tipo de dado que será armazenado. Cada instância do esquema (linha) é chamada de tupla (registro). O modelo relacional implementa estruturas de dados organizadas em relações. Porém, para trabalhar com essas tabelas, algumas restrições precisaram ser impostas para evitar aspectos como: repetição de informação, incapacidade de representar parte da informação e perda de informação. Essas restrições são: integridade referencial, chaves e integridade de junções de relações (TAKAI, ITALIANO, FERREIRA, 2005).

Em relação ao aspecto de repetição de informação, atualmente com a capacidade dos meios de armazenamento de dados e que são relativamente baratos, essa não é mais uma preocupação essencial. E em técnicas mais recentes de auxílio à tomada de decisão como cubos de dados e mineração de dados informações são armazenadas de forma repetida. Nessas técnicas campos de dados

calculados também podem ser armazenados. Isso agiliza a recuperação dos dados. Porém, ressalta-se que com dados armazenados de forma repetida pode haver inconsistência na atualização dos mesmos. Isso pode ser resolvido por uma implementação adequada do sistema que manipula os dados.

2.2 DESENVOLVIMENTO DE SOFTWARE *DESKTOP*

Um software para ambiente *desktop* tem como uma das suas características ser rápido (se comparado à necessidade de acesso remoto, que depende da largura de banda da rede), seguro (inclusive pela impossibilidade de acesso externo se a máquina não estiver em rede) e capaz de utilizar com facilidade todos os recursos da máquina.

Atualmente existem diversas linguagens de programação para desenvolvimento de software para ambiente *desktop*, muitas delas possuem componentes importantes para acelerar o processo de desenvolvimento de software. Um mapa com o histórico de algumas linguagens de programação desde o seu surgimento até o ano atual, de 2013, pode ser obtido em Levenez (2013).

2.3 SISTEMAS DE INFORMAÇÃO

Um sistema de informação é uma estrutura lógica capaz de armazenar e processar dados, transformando-os em informações de utilidade para o usuário.

O uso de sistemas de informação nos diversos tipos de organização é cada vez mais evidente. A necessidade de transformar informação em conhecimento para melhorar a forma de a organização interagir com o ambiente externo e interno tem sido um fator relevante para a manutenção no mercado.

Segundo Laudon e Laudon (1999) um sistema de informação pode ser visto como um conjunto de ferramentas relacionadas com o objetivo de receber informações, processá-las, armazená-las e recuperá-las, de forma a fornecer

subsídios que facilitem o planejamento e o controle no processo de tomada de decisão de organizações.

Sistemas de informação aplicados em processos em empresas comerciais são conhecidos como automação comercial. De acordo com Andrade (2002) automação se refere à automatização de um processo repetitivo, resultando em agilidade e confiabilidade dos resultados.

3 MATERIAIS E MÉTODO

Neste capítulo são apresentadas as ferramentas e tecnologias utilizadas para a modelagem do sistema, a criação e a manutenção do banco de dados e também a linguagem de programação utilizada para implementar o software.

3.1 MATERIAIS

Foram utilizadas as seguintes ferramentas e tecnologias:

a) Visual Paradigm – para a documentação da modelagem que é baseada na UML (*Unified Modeling Language*).

b) DeZign for Database v6 – para o desenvolvimento das entidades relacionais. Essa ferramenta possibilita criar as entidades do banco de dados e os seus relacionamentos e gerar *scripts* para o banco de dados Firebird.

c) Firebird – como banco de dados.

d) IBExpert – como ferramenta administrativa do banco de dados.

e) Linguagem de Programação Delphi – como linguagem de programação que é integrada a um ambiente de desenvolvimento.

3.1.1 Visual Paradigm

Visual Paradigm for UML é uma ferramenta de modelagem que permite elaborar todos os tipos de diagramas da UML (VISUAL PARADIGM, 2013). Essa ferramenta fornece suporte para gerenciamento de casos de uso, diagrama de requisitos SysML (*Systems Modeling Language*) e projeto de banco de dados por meio de diagrama de entidades e relacionamentos. A Figura 1 apresenta a interface principal dessa ferramenta.

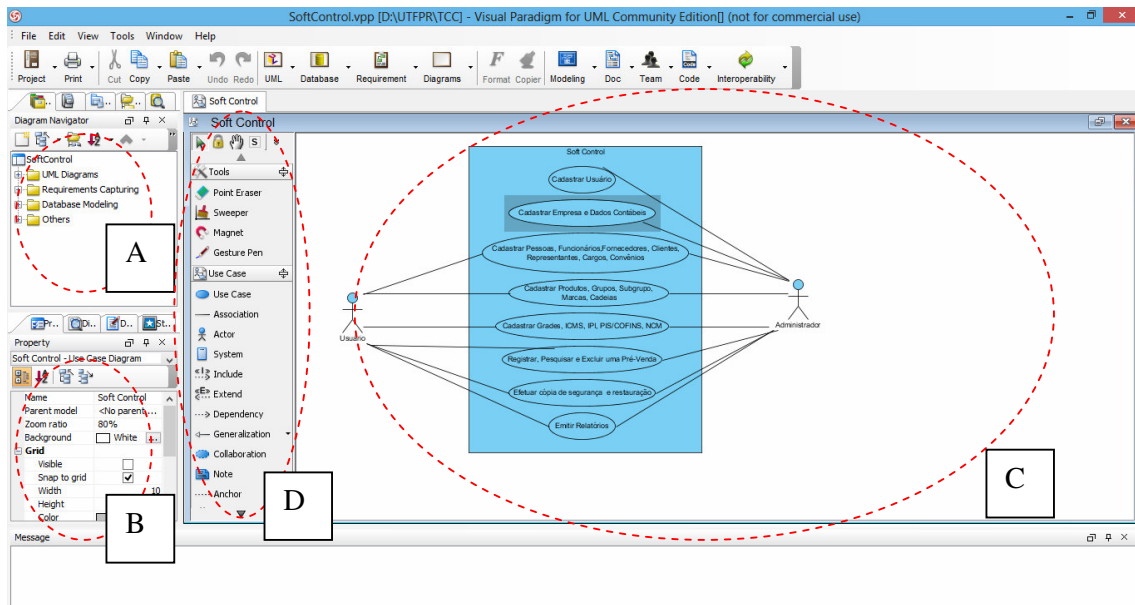


Figura 1 - Tela principal da Ferramenta Visual Paradigm

As partes destacadas com circulos pontilhados na Figura 1 indicam:

A) Navegador de diagramas – apresenta a organização dos projetos existentes. Eles são separados em pastas conforme o tipo de diagrama.

B) Propriedades – apresenta as propriedades do componente selecionado que está em edição.

C) Área de edição – é a área de criação dos diagramas por meio dos componentes dispostos na lateral esquerda.

D) Ferramentas e componentes – são os componentes utilizados para compor os diagramas e as ferramentas relacionadas para a edição dos diagramas.

Visual Paradigm é compatível com a UML 2.1, permite a visualização, criação e edição de processos de negócio. E também possibilita a geração de código na linguagem Java a partir dos diagramas. Além de engenharia reversa em Java, C++, *Extensible Markup Language* (XML) Schema, .NET e *Common Object Request Broker Architecture Interactive Data Language* (CORBA IDL) e permite a geração de códigos compatíveis com XML.

3.1.2 DeZign For Database

DeZign for Databases (DEZIGN, 2013) é uma ferramenta para projeto, criação e manutenção de banco de dados por meio de diagramas de entidades e relacionamentos. O projeto e a geração de bancos de dados são realizados graficamente e por meio de instruções SQL (*Structured Query Language*).

A tela principal da ferramenta DeZign for Databases é apresentada na Figura 2. Essa ferramenta permite criar e alterar bancos de dados e fazer engenharia reversa de bases de dados existentes.

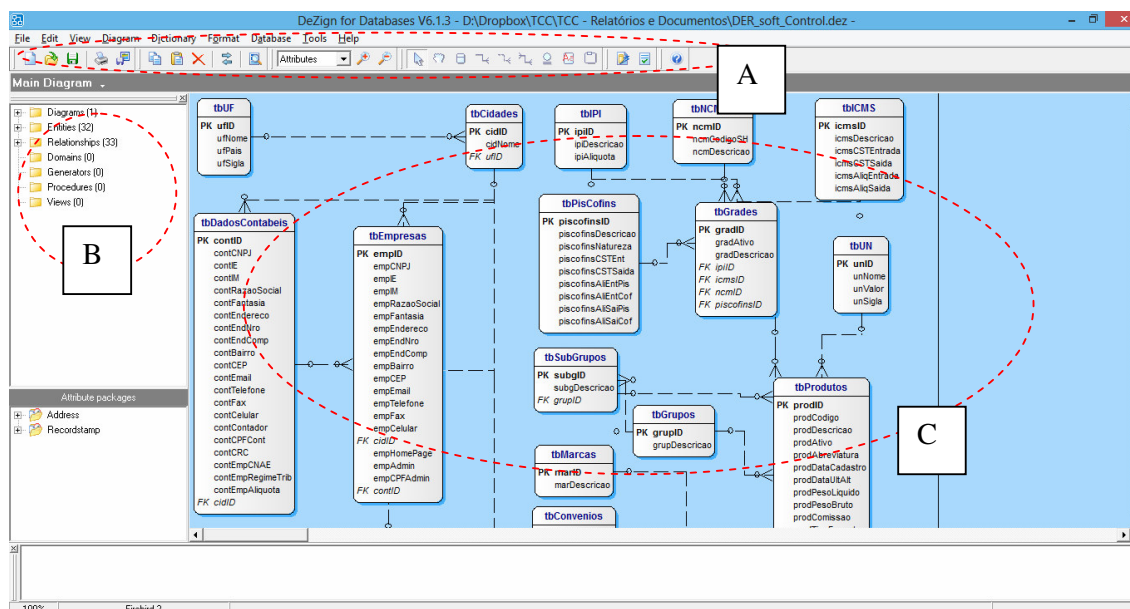


Figura 2 - Tela principal da ferramenta DeZign for Databases

A ferramenta DeZign for Databases é composta basicamente de (áreas circuladas na Figura 2):

A) Barra de ferramentas – localizada na parte superior da tela. Nessa barra estão os recursos comuns de aplicativos como novo, abrir, salvar e excluir. Na metade direita desta barra estão os elementos para a modelagem do banco de dados.

B) Arquivos – pastas com os diagramas, as entidades, os relacionamentos, os domínios, os procedimentos e as visões do diagrama em edição. Na parte inferior dessa barra lateral são apresentadas as propriedades do elemento selecionado.

C) Área de edição – utilizada para a composição dos diagramas.

Os principais recursos da ferramenta DeZign for Databases são (DEZIGN, 2013):

a) Modelagem visual do banco de dados – a modelagem é feita graficamente por meio de diagramas de entidades e relacionamentos. O projeto pode ser visualizado em diferentes níveis de detalhe.

b) Geração da base de dados – a ferramenta gera *scripts* DDL (*Data Definition Language*) completos para criar a base de dados ou gera a base de dados diretamente.

c) Importar base de dados – para derivar um modelo gráfico a partir de bases de dados existentes. A engenharia reversa é feita diretamente a partir do banco de dados ou importada de *scripts* SQL.

d) Sincronização de modelo ou base de dados – funcionalidades de sincronização e comparação para os casos de uso, podendo ser realizado: a) do modelo para a base de dados; b) do modelo para *script*; c) da base de dados para o modelo; d) do *script* para o modelo; e) do modelo para o modelo.

e) Relatórios – geração de relatórios com diferentes níveis de detalhe. Descrições e outras informações relevantes podem ser exportadas para HTML (*HiperText Markup Language* ou Linguagem de Marcação de Hipertexto), Microsoft Word ou PDF (*Portable Document Format*).

f) Verificação de erros – os modelos podem ser verificados quanto à existência de erros durante a fase de projeto. Os elementos do modelo são verificados se estão corretos e completos.

g) Modelos físicos ou modelos independentes de banco de dados – suporte à modelagem para um DBMS (*Data Base Management System*) físico ou ser independente de banco de dados, gerando um modelo portátil de tipos de dados lógicos.

A versão 6 da ferramenta DeZign for Databases oferece, também, um recurso para comparação e junção de projetos, permitindo comparar e juntar versões distintas de um mesmo modelo de dados.

3.1.3 Banco de Dados Firebird

O banco de dados Firebird¹ é descendente do código aberto do Interbase 6.0² desenvolvido pela Borland®. De acordo com Cantu (2005), por volta de 1990 a Borland abriu o código fonte do Interbase 6.0 e um grupo de desenvolvedores criou um novo banco de dados com a linguagem C++ baseado nesse código fonte.

Ainda segundo Cantu (2005), o Firebird conquistou os desenvolvedores que utilizam diversas tecnologias de programação como Java, Delphi, PHP, .NET, entre outras. A aceitação inicial fez com esses desenvolvedores migrassem do banco de dados Interbase para o Firebird, devido à tecnologia estar em pleno desenvolvimento e ser de código aberto.

Cantu (2005) também destaca a facilidade e a rapidez de instalação do Firebird, que não necessita de espaço definido, ou seja, o banco de dados vai crescendo conforme os dados são inseridos e não diminui de tamanho quando os dados são excluídos. Entretanto, os espaços dos arquivos excluídos são reaproveitados para novos dados.

Cantu (2005) ressalta que o Firebird não traz nenhuma ferramenta gráfica para administrar o banco de dados, porém existem diversas ferramentas disponíveis no mercado para *download*. Uma lista dessas ferramentas administrativas está em (FIREBIRD, 2013).

3.1.4 Ferramenta Administrativa IBExpert

Uma das principais necessidades dos administradores ou desenvolvedores de aplicações com o Firebird é uma ferramenta que possibilite administrar seu banco de dados com facilidade.

¹ <http://www.firebirdsql.org/en/firebase-brazil/>, visitado em 23/03/2012.

² <http://www.embarcadero.com/products/interbase>, visitado em 23/03/2012.

O IBExpert³, apresentado na Figura 3, é uma ferramenta que possibilita a administração de bancos de dados Firebird. Dentre os muitos recursos que ela oferece destacam-se o suporte aos bancos de dados Interbase e Firebird, a existência de editores para todos os objetos do banco de dados (tabelas, *triggers* e *stored procedures*) e de um construtor visual de consultas SQL e a documentação do banco de dados em formato HTML.

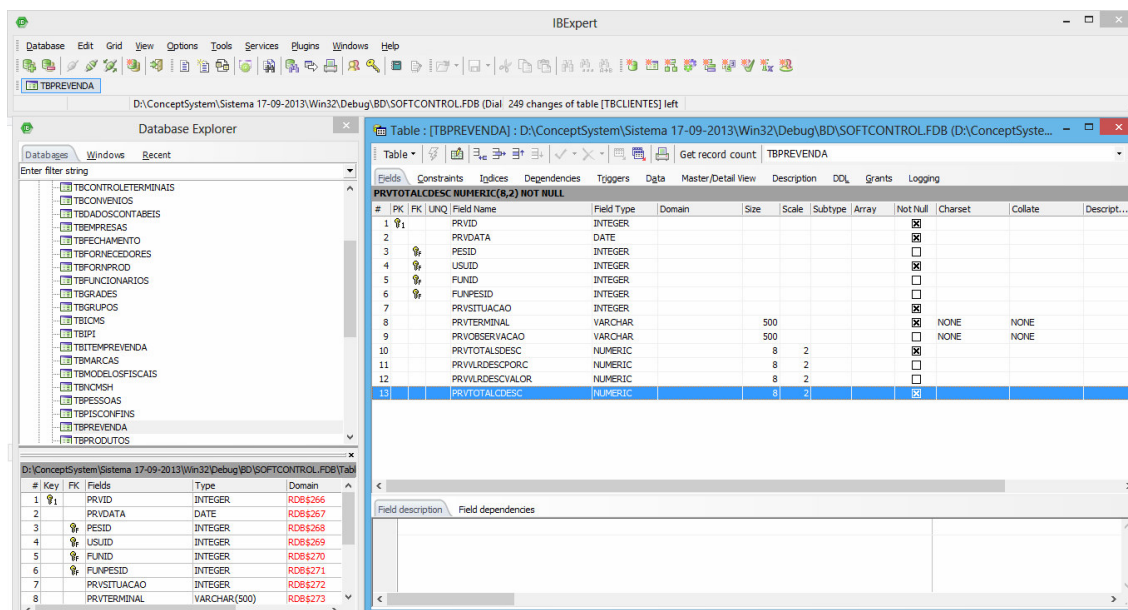


Figura 3 - Ambiente de Trabalho da Ferramenta Administrativa IBExpert

3.1.5 Linguagem de Programação Delphi

O Delphi⁴ é uma ferramenta utilizada para desenvolver aplicações rapidamente (RAD - *Rapid Application Development*), utilizando um ambiente integrado para desenvolvimento (IDE - *Integrated Development Environment*) visual e orientado a objetos.

Segundo Manzano e Mendes (1999), o Delphi surgiu da linguagem de programação *Object Pascal*, que teve como origem a linguagem Pascal, que foi desenvolvida por Niklaus Wirth, com objetivo de ensinar acadêmicos universitários a programar.

³ <http://ibexpert.net/ibe/>, visitado em 21/03/2012.

⁴ <http://www.embarcadero.com/br/products/delphi>, visitado em 23/03/2012.

A ferramenta de desenvolvimento permite a construção de aplicações para objetivos distintos de forma rápida e com bom desempenho, devido à facilidade de trabalhar com componentes visuais e com a possibilidade de explorar diversos recursos do Windows.

“À medida que se desenvolve a inserção de componentes no formulário, o código de programa é implementado automaticamente. Apesar desse recurso, será necessário, em várias ocasiões, recorrer à intervenção do programador no código apresentado, para que ele se adapte corretamente à necessidade da aplicação em desenvolvimento” (MANZANO, MENDES, 1999, p.7).

A Figura 4 apresenta a interface da IDE Delphi.

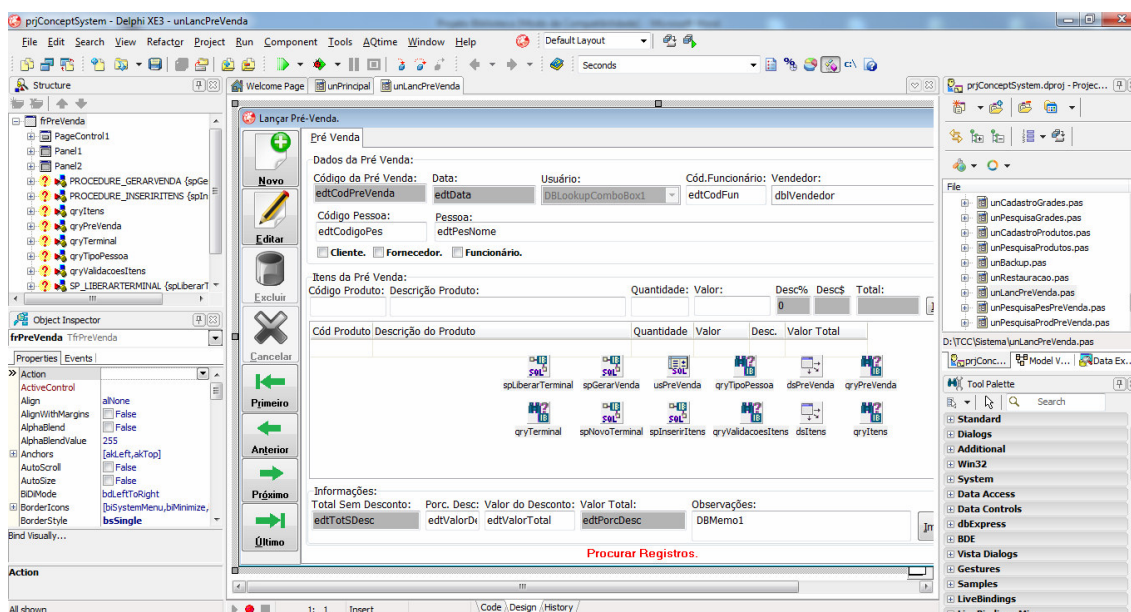


Figura 4 - Ambiente Integrado de Desenvolvimento Delphi XE3

Dentre as principais características da ferramenta de desenvolvimento de software Delphi, apresentada na Figura 4, destacam-se:

- IDE possui uma paleta composta por diversos botões *drop-down* que podem ser arrastados para a janela do formulário.
- É possível trabalhar com o código fonte integrado com os componentes visuais utilizados nos formulários.
- É possível criar novos componentes e adicionar componentes e ferramentas de terceiros.
- Pode-se visualizar o código fonte de diversos componentes localizados dentro da biblioteca VCL (*Visual Component Library*, ou biblioteca de

componentes visuais), sendo possível também fazer alterações nos componentes da ferramenta de desenvolvimento.

- Utilizando o *Project Manager* é possível visualizar e organizar a estrutura do projeto, ou seja, visualizar e estruturar as *units* e formulários.

No desenvolvimento do sistema foram utilizados os pacotes padrão de desenvolvimento Delphi, principalmente as seguintes páginas de componentes:

- **Interbase:** Os componentes do interbase permitem estabelecer uma conexão com o banco de dados Firebird. Citando os principais componentes desta página, tem-se: a) *IBDatabase* realiza a conexão com o banco de dados; b) *IBQuery* por meio de instrução SQL seleciona e realiza filtro das informações que serão exibidas; e, c) *IBTable* para realizar o acesso diretamente a uma tabela de dados.
- **Data Access:** Nessa aba estão localizados os componentes para acesso aos dados, sendo que os mesmos servem como intermediadores para acessar o banco de dados. Por exemplo, os conjunto de dados (*DataSets*) podem relacionar-se com todos os componentes da aba *Data Access*, com o uso do componente *DataSource*.
- **Data Controls:** É constituído por diversos componentes que facilitam operações com o banco de dados, acesso a informações, compor um *grid* com os dados de uma tabela e buscar um campo por meio de código. Por exemplo, um *DBEdit* disponibiliza o acesso a um campo de determinada tabela, um *DBGrid* permite exibir os dados de um tabela, um *DBLookupComboBox* localiza um campo de uma segunda tabela e um *DBNavigator* faz a navegação das informações exibidas.
- **Standard e Additional:** Estas abas são constituídas por componentes mais comuns do Delphi, que servem para apresentar informações ou fazer uso de outros componentes como, por exemplo, o *MainMenu* que faz a criação de um menu para obter acesso a formulários ou a outras recursos.
- **FastReport:** É uma ferramenta utilizada para o desenvolvimento de relatórios, possui compatibilidade com os componentes de acesso ao banco de dados do Delphi XE3, sendo que para isso é utilizado um *frxDBDataSet*.

3.2 MÉTODO

Inicialmente foi necessário estudar as tecnologias envolvidas nas etapas do ciclo de vida do sistema. Esse estudo envolveu as ferramentas de modelagem e as utilizadas para implementação do sistema.

O estudo da ferramenta de modelagem DeSign for Database teve como objetivo o entendimento do seu funcionamento e das formas de elaboração de um diagrama de entidade e relacionamento (projeto conceitual). Na próxima etapa aspectos sobre o funcionamento do banco de dados Firebird, sua origem, seus atributos e formas de utilização foram levados em consideração. A implementação do projeto conceitual do banco de dados no Firebird foi realizada por meio da ferramenta administrativa IBExpert. Finalizando com a definição de um ambiente visual e orientado a objetos baseado na linguagem de programação Delphi como ferramenta de desenvolvimento.

A seguir são apresentados estudos de caso realizados na fase de aprendizado das tecnologias.

3.2.1 Estudos de Casos em Delphi

Para conhecer e familiarizar-se com o ambiente de desenvolvimento de software Delphi na versão XE3 foram desenvolvidos diversos estudos de caso, os quais se referiam a problemas similares aos que seriam encontrados no desenvolvimento do sistema proposto.

Os casos estudados sempre estavam relacionados com algum componente ou ferramenta específica, visando utilizar o máximo possível de recursos, tanto como compreender melhor a linguagem e o funcionamento do ambiente de programação Delphi. Com o uso de duas aplicações buscou-se observar diferentes formas de desenvolvimento, levando em consideração agilidade no desenvolver e desempenho e usabilidade para o usuário.

A seguir são apresentadas as aplicações desenvolvidas na fase de aprendizado das tecnologias.

- 1) Desenvolvimento do projeto e de uma aplicação para controle de laudos de maçãs para uso em cooperativas, sendo descritos a seguir a tela principal, formulário de lançamento de laudos, formulário de pesquisa e cadastro.

A Figura 5 apresenta a interface inicial dessa aplicação.

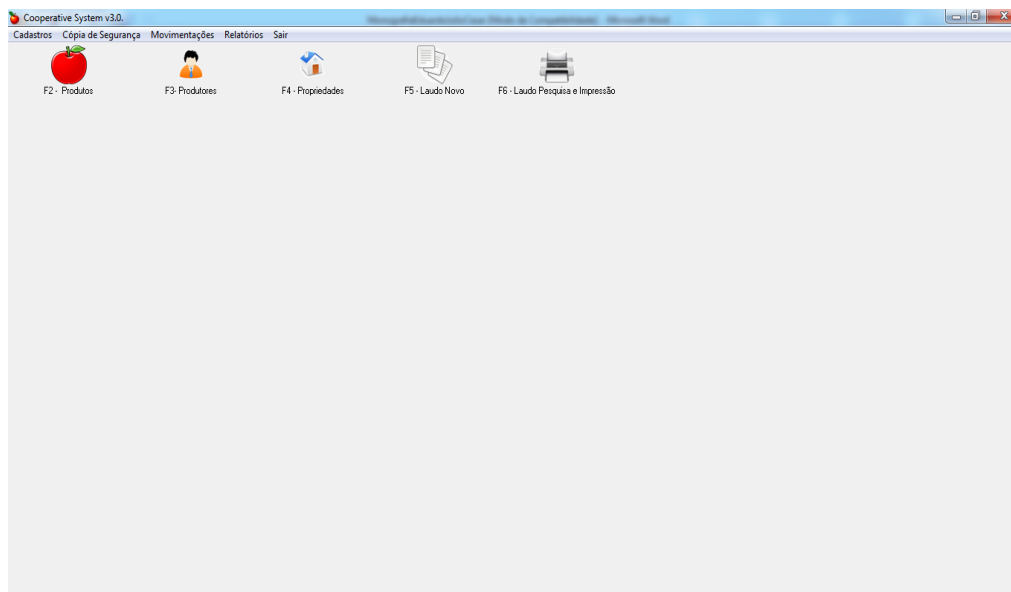


Figura 5 - Tela Principal Cooperative System

Na Figura 6 é apresentada a tela para inclusão de dados para classificação do produto sendo cadastrado.

Código Laudo	Data	Produtor
<input type="text"/>	<input type="text"/>	<input type="text"/>
Usuário	Propriedade	
<input type="text"/>	<input type="text"/>	
Código de Barras	<input type="text"/>	
Quantidade	<input type="text"/>	
<input type="text" value="1"/>		
F5 - Cancelar		
F6 - Lançamento Manual		
F7 - Remover Registro		
F8 - Encerrar		

Figura 6 - Tela de Classificação Cooperative System

- 2) Desenvolvimento de um Sistema para Pontos de Fidelidade, utilizado para geração e manutenção de cupons de ponto de fidelidade. Pontos de fidelidade são bônus fornecidos a clientes. A tela principal deste sistema é apresentada na Figura 7.

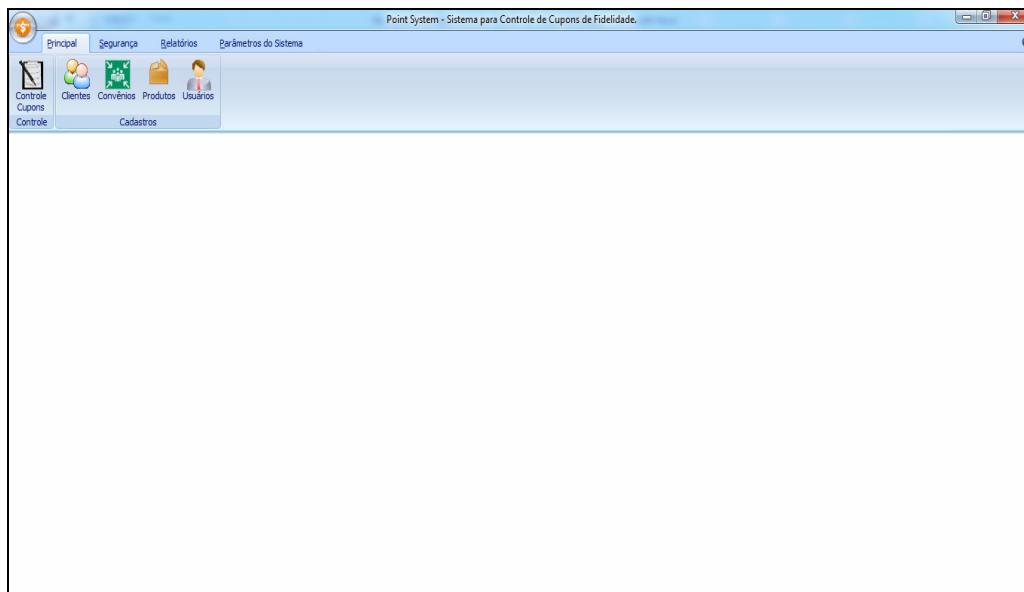


Figura 7 - Tela Principal Point System

A tela para geração de cupom é semelhante à tela de entrada de dados para o sistema para classificação de produtos, apresentado na Figura 8.

Figura 8 - Tela Geração Cupom Point System

3.2.2 Formulários Padrão

Para o desenvolvimento do projeto proposto, foram desenvolvidos alguns formulários chamados de 'padrão'. Esses formulários pré-fabricados ajudam a desenvolver uma aplicação com maior rapidez, pois podem ser utilizados para aplicar o conceito de herança. Sua principal vantagem é trazer elementos comuns encontrados na maioria dos formulários de cadastro, tornando esses elementos padronizados e funcionando como modelos para o usuário desenvolvedor. A Figura 9 apresenta um formulário padrão para cadastro.

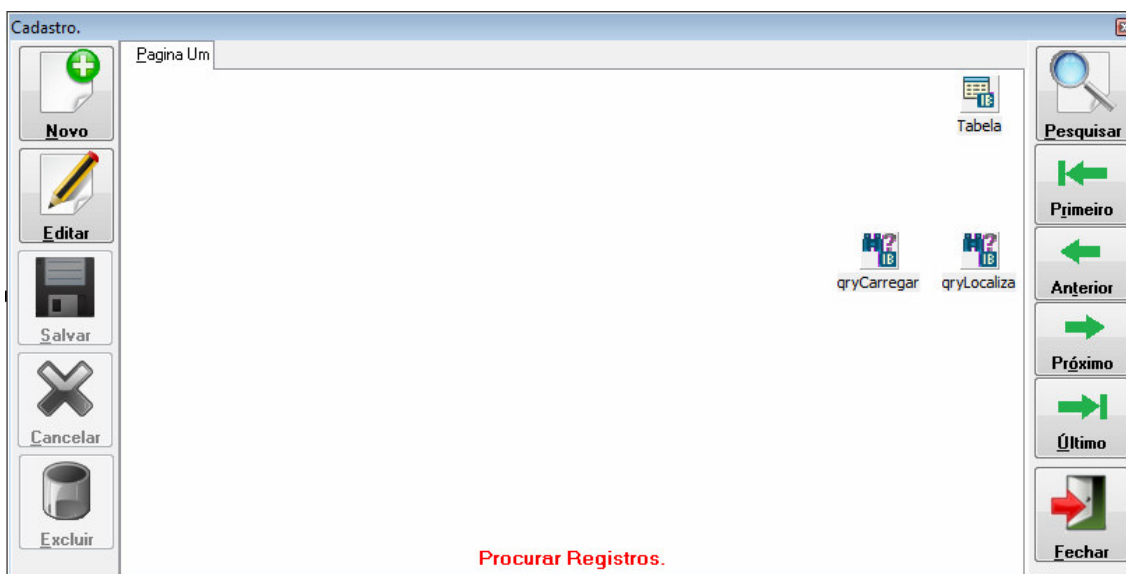


Figura 9 - Formulário padrão de cadastro

A Figura 10 apresenta o formulário padrão para pesquisa.

Pesquisa.

Pesquisa por: ???

tab IBQuery1 DataSource1

Alterar

Atualizar

Fechar

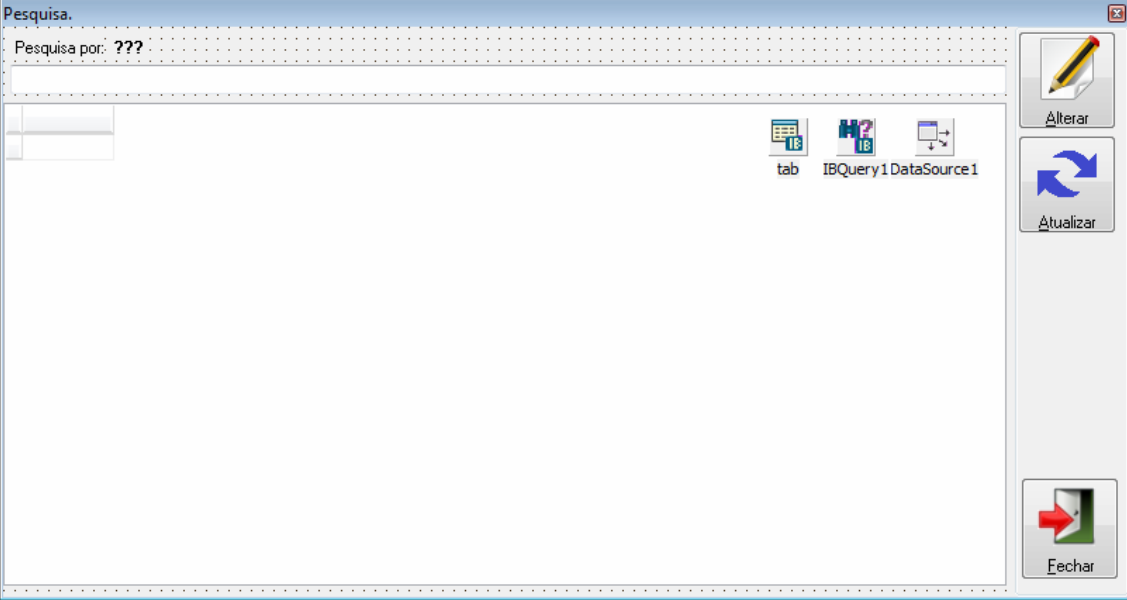


Figura 10 - Formulário padrão de pesquisa

4 RESULTADO

Este capítulo apresenta o que foi obtido como resultado da realização deste trabalho que é a modelagem e o desenvolvimento de um sistema gerencial para empresas varejistas.

4.1 APRESENTAÇÃO DO SISTEMA

O sistema fará o controle do processo de pré-vendas, provendo as informações aos usuários e armazenando-as no banco de dados em tempo real. Assim, se ocorrer algum problema como queda de energia, por exemplo, as informações referentes às pré-vendas não serão perdidas e posteriormente poderá ser dada continuidade na operação ou impressão da mesma.

O acesso ao sistema será restrito ao administrador e aos usuários previamente cadastrados por meio de *login* e senha. Os dados da empresa, são compostos pelos dados cadastrais (cadastro completo com todos os dados, Cadastro Nacional de Pessoas Jurídicas (CNPJ), endereço, razão social, entre outros), juntamente com os dados contábeis, que serão preenchidos pelo administrador do sistema.

O sistema trabalhará com uma entidade pessoa, uma generalização de todas as pessoas relacionadas com o sistema por meio do cadastro de pessoa, herdando os dados e completando com dados específicos e complementares, gerando cadastros de clientes, funcionários e fornecedores. Também relacionando com o cadastro de usuários e representantes.

O cadastro de produtos será relacionado com os cadastros de grupos e subgrupos, marcas, cadeias, unidades de medida e as grades tributárias. As grades tributárias são compostas por todos os dados fiscais dos produtos como IPI (Imposto sobre Produtos Industrializados), ICMS (Imposto sobre Circulação de Mercadorias e Serviços), PIS/COFINS (Programa de Integração Social/ Contribuição para o Financiamento da Seguridade Social), CST (Código de Situação Tributária) de entrada e saída e NCM (Nomenclatura Comum do Mercosul). Essa funcionalidade

agiliza o processo de manutenção quando acontece qualquer alteração fisco-contábil, permitindo que o usuário altere somente a tributação ou outro dado contábil direto na grade, alterando, assim, todos os produtos com as mesmas características e relacionados com a mesma grade cadastrada.

Um dos requisitos mais importantes do sistema se refere à criação das pré-vendas. Para iniciar uma pré-venda deve ser informado: código ou nome do funcionário (vendedor), código ou nome da pessoa, a data que é obtida do sistema operacional, o usuário que está logado no sistema e o campo observações. Na parte de baixo da janela haverá um campo para selecionar os produtos. Para o produto deverá ser informada quantidade maior que zero. O preço virá do cadastro de produtos, que pode ser alterado, e poderá ser informado o valor de desconto em valor ou percentual, caso exista. Uma pré-venda pode ter um ou mais produtos. Um produto depois de inserido pode ser removido ou editado. Após salvar a pré-venda ela pode ser navegada, pesquisada e impressa em formato de documento não fiscal como uma condicional ou como um orçamento.

Uma condicional é situação na qual o produto é levado pela pessoa em modo de experimentação e decisão da compra a ser realizada pela pessoa. O orçamento é um rascunho e/ou levantamento de preços e condições que precedem à compra.

Cada inserção no cadastro de pré-venda é inserida diretamente no banco de dados juntamente com uma transação. Caso ocorra algum problema ou queda de energia os dados já lançados não serão perdidos.

O sistema disponibilizará diversos relatórios para os usuários, como relatório de clientes, fornecedores, produtos, grades, cadeia de preços e das pré-vendas realizadas. Os relatórios poderão ser salvos em formato PDF.

O sistema terá formulário para efetuar *backup* do banco de dados e restauração dos mesmos, promovendo a prevenção da segurança das informações armazenadas no sistema.

4.2 MODELAGEM DO SISTEMA

O levantamento de dados foi realizado por meio de observações em empresas varejistas que utilizam algum software gerencial e, principalmente, em empresas que não possuem nenhum sistema e que utilizam meio de controle das atividades e informação que às vezes se limitam a anotações em cadernos ou em planilhas eletrônicas de cálculo.

A análise levou em consideração a ideia principal de automatizar os processos de maneira que a empresa não sofra com a mudança, ou seja, reverter os processos manuais em automatizados de maneira fácil de entender e aprender, visando às regras de usabilidade para atender usuários avançados e até mesmo aqueles sem habilidade com equipamentos de tecnologia.

Segundo Wazlawick (2011), a etapa de análise de requisitos serve para estruturar e detalhar os requisitos de forma que eles possam ser abordados na fase de elaboração para o desenvolvimento de outros elementos como casos de uso, classes e interfaces. O levantamento de requisitos é o processo de descobrir quais são as funções que o sistema deve realizar e quais são as restrições que existem sobre essas funções. Todas as operações efetuadas pelo sistema virão a constituir a funcionalidade do sistema e por isso são chamadas também de requisitos funcionais.

Pelas etapas do processo de análise foi possível identificar os requisitos funcionais constantes no Quadro 1.

Identificação	Requisito	Descrição
F1	Cadastrar Usuários	Cadastro dos usuários que poderão ter acesso ao sistema, sendo usuário simples ou administrador (acesso a configurações).
F2	Cadastrar Empresa	Dados da empresa usuária do sistema.
F3	Cadastrar Cadeias	Cadeias de preços de produtos para facilitar o processo de alteração de preços por produtos da mesma família, volume/peso.
F4	Cadastrar Cargos	Cargos dos funcionários. Cada cargo pode ter um tipo de acesso diferente no sistema.
F5	Cadastrar Pessoas	Dados de uma entidade, tanto pessoa física como jurídica. Uma pessoa pode ser funcionário da empresa, cliente e mesmo fornecedor.
F6	Cadastrar Clientes	Dados exclusivos de clientes, como limite, data de vencimentos, e completados dos dados herdados de Pessoa.
F7	Cadastrar Fornecedores	Dados exclusivos dos fornecedores, como ramo de atividade e televendas e

		completados dos dados herdados de pessoa.
F8	Cadastrar Funcionários	Dados exclusivos de funcionários, como carga horária, data admissão, cargo, salário, número carteira de trabalho e número do PIS e completados dos dados herdados de pessoa.
F9	Cadastrar Convênios	Convênio de clientes, separando um grupo de clientes para gerenciamento e negociações.
F10	Cadastrar Dados Contábeis	Dados do contador juntamente com os dados fiscais da empresa usuária do sistema, como código CNAE (Classificação Nacional de Atividades Econômicas), regime tributário entre outros.
F11	Cadastrar Grades Tributárias	Grade tributária dos produtos para agilizar o processo de cadastro, determinando um tipo de produto e vinculando a ela alíquotas de ICMS, PIS/COFINS, IPI e NCM.
F12	Cadastrar Grupos	Os grupos permitem subdividir os produtos.
F13	Cadastrar Subgrupo	Uma subdivisão do grupo de produtos.
F14	Cadastrar ICMS	Cadastro de alíquotas de ICMS e seus códigos CST de entrada e saída.
F15	Cadastrar PIS/COFINS	Cadastro de Alíquotas de PIS/COFINS e seus códigos CST de entrada e saída e natureza.
F16	Cadastrar IPI	Cadastro das alíquotas de IPI.
F17	Cadastrar Marcas	Cadastro de marcas dos produtos para permitir melhor filtragem das informações.
F18	Cadastrar NCM	Cadastro dos NCM dos produtos que serão vinculados às grades tributárias.
F19	Cadastrar Produto	Cadastro dos produtos que serão comercializados pela empresa no processo de pré-venda.
F20	Cadastrar Representantes	Os representantes vinculados aos fornecedores. Um representante pode estar vinculado a diversos fornecedores.
F21	Cadastrar Unidade de Medida	As unidades de medida com sua sigla e valor para uso nas comercializações.
F22	Registrar Pré-Venda	A pré-venda é gerada armazenando o código do usuário, o nome do vendedor, a pessoa, os produtos e se necessário observações. A pré-venda poderá se tornar uma NF-e, uma condicional ou até mesmo um cupom fiscal.
F23	Pesquisar Pré-Venda	Permite localizar as pré-vendas já geradas e que estão armazenados no sistema.
F24	Excluir Pré-Venda	Excluir as pré-vendas cadastradas.

F25	Imprimir Pré-Venda	Realizar a impressão das pré-vendas.
F26	Emitir Relatório de Pré-Venda	Relatório das pré-vendas armazenadas por data, por usuário ou por pessoa.
F27	Emitir Relatório de Produtos	Relatório de produtos cadastrados.
F28	Emitir Relatório de Pessoas	Relatório de pessoas cadastradas.
F29	Efetuar controle do Fechamento de Movimento	Controlar a data do último fechamento de movimento para não permitir adulteração de informações registradas em dias anteriores e persistência dos dados.
F30	Armazenar Versão do Sistema	Armazenar qual a versão do sistema e qual versão do banco de dados, para manter consistência em futuras manutenções e modificações.
F31	Efetuar Controle de Terminais	O sistema ao iniciar uma pré-venda deve armazenar o terminal. Caso aconteça uma falha, o sistema poderá recuperar a pré-venda em preenchimento.
F32	Efetuar cópia de segurança (<i>backup</i>)	A realização de cópia de segurança dos dados armazenados no sistema.
F33	Efetuar restauração da cópia de segurança	Restaurar cópias de segurança realizadas dos dados do sistema.

Quadro 1 - Requisitos do sistema

Esses requisitos foram organizados sob a forma de casos de uso apresentados na Figura 11. Nessa figura estão os casos de uso considerados essenciais para o processo principal de geração de pré-venda.

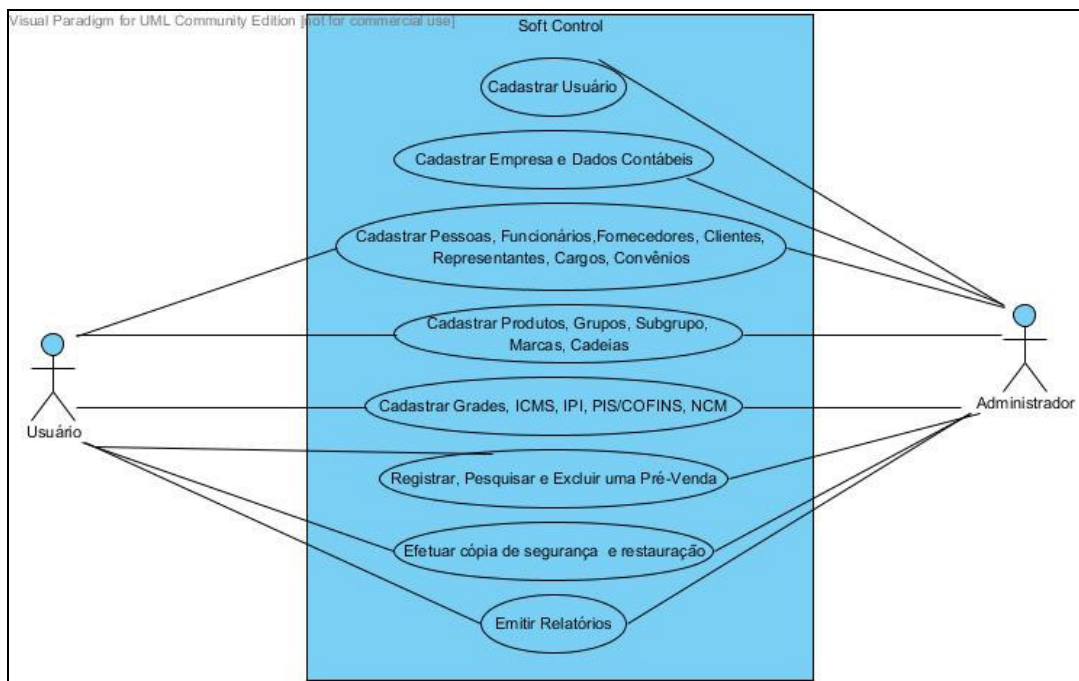


Figura 11 - Diagrama de Casos de Uso

O Quadro 2 lista os requisitos não funcionais relacionados às principais entidades do sistema.

Conceito	Observação
Cadeias	Não podem ser excluídas cadeias que já tenham relacionamento com produtos.
Cargos	Não podem ser excluídos cargos que já tenham relacionamento com funcionários.
Clientes	Não podem ser excluídos clientes que já tenham relacionamento com pré-vendas.
Convênios	Não podem ser excluídos convênios que já tenham relacionamento com clientes.
Dados Contábeis	É permitido apenas um formulário com os dados contábeis da empresa.
Empresa	É permitido apenas um formulário com os dados da empresa no sistema não deixando salvar campos obrigatórios em branco. Os dados devem se referir à empresa usuária do sistema.
Fornecedores	Não podem ser excluídos fornecedores que já tenham relacionamento com produtos.
Funcionários	Não podem ser excluídos funcionários que já tenham relacionamento com pré-vendas e usuários.
Grades	Não podem ser excluídas grades que já tenham relacionamento com produtos.
Grupos	Não podem ser excluídos grupos que já tenham relacionamento com produtos.

ICMS	Não podem ser excluídos ICMS que já tenham relacionamento com grades.
IPI	Não podem ser excluídos IPI que já tenham relacionamento com grades.
Marcas	Não podem ser excluídas marcas que já tenham relacionamento com produtos.
Pessoas	Não podem ser excluídas pessoas que já tenham relacionamento com clientes, funcionários ou fornecedores.
PIS/COFINS	Não podem ser excluídos PIS/COFINS que já tenham relacionamento com grades.
Pré-Venda	As pré-vendas não podem ser excluídos caso haja algum produto lançado.
Produtos na Pré-Venda	A inclusão só será efetuada se o produto estiver devidamente cadastrado.
Produtos	Não podem ser excluídos fornecedores que já tenham relacionamento com fornecedores e pré-vendas.
Representantes	Não podem ser excluídos representantes que já tenham relacionamento com fornecedores.
Subgrupo	Não podem ser excluídos subgrupos que já tenham relacionamento com grupos.
UN	Não podem ser excluídas unidades de medida que já tenham relacionamento com produtos.
Usuários	Não podem ser excluídos usuários que já tenham relacionamento com alguma pré-venda.

Quadro 2 - Requisitos não funcionais

O Quadro 3 apresenta a descrição do requisito funcional Registrar Pré-Venda.

F22 Registrar Pré-Venda	Oculto ()
<p>Descrição: O usuário clica no botão novo, informa o código do vendedor ou nome, código da pessoa ou nome, seleciona o produto por código ou descrição, informa a quantidade, verifica o preço de venda que veio do cadastro que pode ser alterado se desejado. Se houver desconto em valor ou porcentagem deve ser informado. Se houver mais produtos repetir o processo. No rodapé da janela serão apresentadas as informações do total sem desconto, valores caso tenha descontos, valor total e um campo para observações se necessário.</p> <p>Usuário: Usuário.</p> <p>Informações de Entrada: vendedor, pessoa, produtos com quantidade e observações.</p> <p>Informações de Saída: Pré-venda.</p> <p>Restrições lógicas: Não permitir a inclusão de produtos, vendedores ou pessoas não cadastrados no sistema. A quantidade e o preço de produto não podem ser menor ou igual a zero. Os valores dos descontos não podem ser negativos ou que ultrapassem o valor total bruto. Não permitir salvar uma pré-venda sem produtos lançados.</p> <p>Restrições tecnológicas: Leitor de código de barras para leitura dos códigos dos produtos.</p>	
Requisitos Não-Funcionais	

Nome	Restrição	Categoria	Desejável	Permanente
NF 22.1 Informar vendedor	O usuário deve informar o vendedor pelo código ou pelo nome, ou seja, devidamente cadastrado como usuário.	Interface	()	(x)
NF 22.2 Informar Pessoa	O usuário deve informar a pessoa relacionada a pré-venda, que pode ser tanto um fornecedor, cliente ou funcionário.	Interface	()	(x)
NF 22.3 Informar Produtos	O usuário deve informar um ou mais produtos relacionando quantidade, valores de desconto em valor ou percentual e se preferir alterar o valor que busca do preço de venda do cadastro.	Interface	()	(X)
NF 22.4 Janela única	Todo o processo de lançamentos de produtos na pré-venda deve ser em uma única janela.	Interface	()	(X)
NF 22.5 <i>Grid</i> de Produtos	Durante todo o processo de lançamentos de produtos, os mesmos já lançados devem ser exibidos em um <i>grid</i> na parte inferior contendo o código do produto, descrição, quantidade, valor do desconto e valor total.	Interface	()	(X)

Quadro 3 - Caso de uso registrar pré-venda

A Figura 8 mostra o diagrama de entidade e relacionamento do sistema desenvolvido por meio da ferramenta DeZign For Database, descrita na subseção 3.1.2, com o propósito de apresentar as tabelas e o relacionamento existente entre elas.

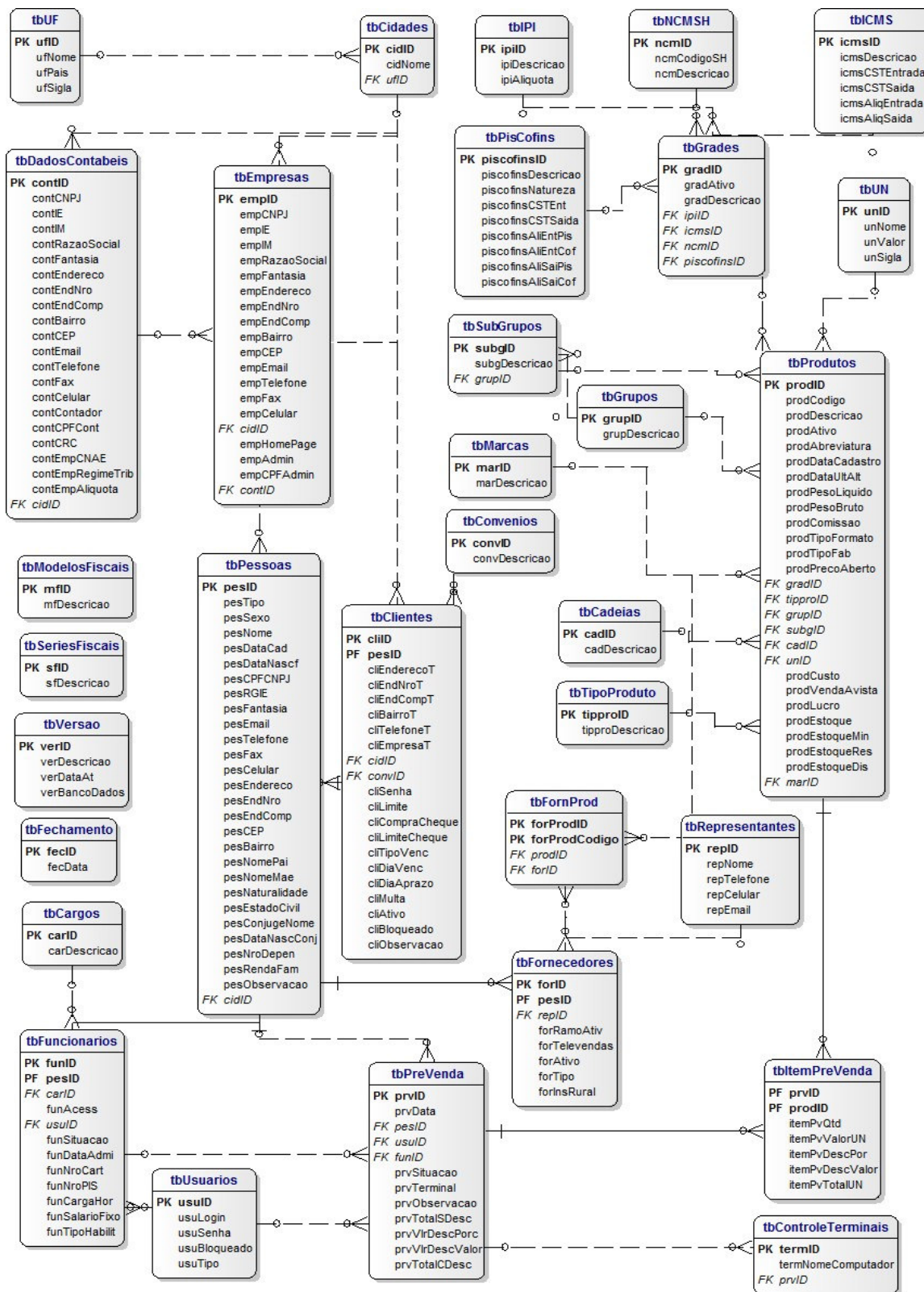


Figura 12 - Diagrama entidades e relacionamentos

Na relação das tabelas apresentada a seguir é possível verificar a finalidade de cada uma, bem como o seu nome, a relação de campos, a definição das restrições de chave primária e chave estrangeira, campos nulos não permitidos e a descrição dos campos.

O Quadro 3 descreve os campos do cadastro de cadeia de preços, grupo de um mesmo tipo de produto, da mesma marca e volume, para facilitar a troca de preços quando houver alteração.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
CADID	Integer	*		*	Código da cadeia de preço.
CADDESCRICAÇÃO	VarChar(100)			*	Descrição da cadeia de preço.

Quadro 4 - Cadastro de cadeias (tbCadeias)

O Quadro 5 descreve os campos do cadastro de cargos dos funcionários.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
CARID	Integer	*		*	Código do cargo.
CARDESCRICAÇÃO	VarChar(100)			*	Descrição do cargo.

Quadro 5 - Cadastro de cargos (tbCargos)

O Quadro 6 descreve os campos dos dados das cidades de todo o Brasil juntamente com o código do IBGE (Instituto Brasileiro de Geografia e Estatística), que é utilizado para indicar a cidade da empresa, da pessoa e do contador.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
CIDID	Integer	*		*	Código da cidade e do IBGE.
CIDNOME	VarChar(100)			*	Nome da cidade.
UFID	Integer		*	*	Código da Unidade Federativa.

Quadro 6 - Cidades (tbCidades)

O Quadro 7 descreve os campos do cadastro de clientes, que se completam com os dados herdados da tabela pessoa.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
CLIID	Integer	*		*	Código do cliente.
PESID	Integer	*	*	*	Código da pessoa.
CLIENDERCOT	VarChar(100)				Endereço da pessoa.
CLIENDNROT	VarChar(10)				Número do endereço do trabalho do cliente.
CLIENDCOMPT	VarChar(50)				Complemento do endereço do trabalho do cliente.
CLIBAIRROT	VarChar(50)				Bairro do trabalho da cliente.
CLITELEFONET	VarChar(11)				Telefone do trabalho da cliente.
CLIEMPRESAT	VarChar(100)				Nome da empresa trabalho da cliente.

CIDID	Integer		*		Cidade do trabalho da cliente.
CONVID	Integer		*		Convênio do cliente.
CLISENHA	VarChar(14)				Senha do cliente.
CLILIMITE	Numeric(8,2)			*	Limite de crédito do cliente.
CLICOMPACHEQUE	VarChar(1)			*	Cliente de cheque.
CLILIMITECHEQUE	Numeric(8,2)				Valor do limite do cheque.
CLITIPOVENC	VarChar(1)			*	Tipo de vencimento.
CLIDIAVENC	Integer			*	Dia do vencimento.
CLIDIAAPRAZO	Integer				Dias de prazo após o vencimento.
CLIMULTA	Numeric(8,2)				Valor multa juros mês.
CLIATIVO	VarChar(1)			*	Cliente ativo ou não.
CLIBLOQUEADO	VarChar(1)			*	Cliente bloqueado ou não.
CLIOBSERVACAO	VarChar(500)				Observação do cliente.

Quadro 7 - Cadastro de clientes (tbClientes)

O Quadro 8 descreve os campos da tabela do controle de terminais, que serve para controlar todos os terminais abertos relacionados com suas respectivas pré-vendas, garantido que em caso de falhas o usuário possa recuperar os dados do formulário que estava sendo manipulado.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
TERMID	Integer			*	Código do terminal.
TERMNOMECOMPUTADOR	VarChar(100)			*	Identificação do Computador.
PRVID	Integer		*	*	Código de identificação da pré-venda.

Quadro 8 - Controle de terminais (tbControleTerminais)

O Quadro 9 descreve os campos de cadastro dos convênios.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
CONVID	Integer	*		*	Código do convênio.
CONVDESCRICAOC	VarChar(100)			*	Descrição do convênio.

Quadro 9 - Cadastro de convênios (tbConvenios)

O Quadro 10 descreve os campos do cadastro dos dados contábeis da empresa.

Nome do Campo	Tipo (tamanho)	P K	FK	Não Nulo	Descrição
CONTID	Integer	*		*	Código dos dados contábeis.
CONTICNPJ	VarChar(14)			*	CNPJ do contador.
CONTIE	VarChar(15)			*	Inscrição estadual do contador.
CONTIM	VarChar(15)				Inscrição municipal do contador.
CONTRAZAOSOCIAL	VarChar(100)			*	Razão social do contador.
CONFANTASIA	VarChar(100)			*	Nome da fantasia do contador.
CONTENDereco	VarChar(100)			*	Endereço do contador.
CONTENDNRO	VarChar(10)			*	Número do endereço do contador.
CONTENDCOMP	VarChar(50)				Complemento do endereço do contador

CONTBAIRRO	VarChar(50)			*	Bairro do contador.
CONTCEP	VarChar(8)			*	CEP do contador
CONTEMAIL	VarChar(100)				E-mail contador.
CONTTELEFONE	VarChar(11)				Telefone do contador.
CONFAX	VarChar(11)				Fax do contador.
CONTCELULAR	VarChar(11)				Celular do contador.
CIDID	Integer		*	*	Código da cidade.
CONTCONTADOR	VarChar(100)				Nome do contador.
CONTCPFCONT	VarChar(11)				CPF do contador da empresa.
CONTCRC	VarChar(15)				CRC do contador.
CONTEMPCNAE	VarChar(7)				Código CNAE da empresa.
CONTEMPREGIMETTRIB	VarChar(1)				Código do regime tributário da empresa.
CONTEMPALQUOTA	Numeric(8,2)				Alíquota da Empresa se simples nacional.

Quadro 10 - Cadastro dos Dados Contábeis (tbDadosContabeis)

O Quadro 11 descreve os campos dos dados da empresa usuária do software.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
EMPID	Integer	*		*	Código da empresa.
EMPCNPJ	VarChar(14)			*	CNPJ da empresa.
EMPIE	VarChar(15)			*	Inscrição Estadual da empresa.
EMPIM	VarChar(15)				Inscrição Municipal da empresa.
EMPRAZASOCIAL	VarChar(100)			*	Razão Social da empresa.
EMPFANTASIA	VarChar(100)			*	Nome Fantasia da empresa.
EMPENDRECO	VarChar(100)			*	Endereço da empresa.
EMPENDNRO	VarChar(10)			*	Número do Endereço da empresa.
EMPENDCOMP	VarChar(50)				Complemento do Endereço da empresa.
EMPBAIRRO	VarChar(50)			*	Bairro da empresa.
EMPCEP	VarChar(8)			*	CEP da empresa.
EMPEMAIL	VarChar(100)				E-mail da empresa.
EMPTTELEFONE	VarChar(11)				Telefone da empresa.
EMPFAX	VarChar(11)				Fax da empresa.
EMPCELULAR	VarChar(11)				Celular da empresa.
CIDID	Integer		*	*	Código da cidade.
CONTID	Integer		*	*	Código dos dados contábeis da empresa.
EMPHOMEPAGE	VarChar(100)				Página web da empresa.
EMPADMIN	VarChar(100)				Nome o administrador da empresa.
EMPCPFADMIN	VarChar(11)				CPF do administrador da empresa.

Quadro 11 - Dados da Empresa (tbEmpresas)

O Quadro 12 descreve os campos da data de fechamento do movimento diário.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
FECID	Integer	*		*	Código do fechamento.
FECDATA	Date			*	Data do fechamento.

Quadro 12 - Data do Fechamento do Movimento Diário (tbFechamento)

O Quadro 13 descreve os campos de cadastro de fornecedores, que se completam com os dados herdados da tabela pessoa.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
FORID	Integer	*		*	Código do fornecedor.
PESID	Integer	*	*	*	Código da pessoa.
REPID	Integer		*		Código do representante.
FORRAMOATIV	VarChar(100)				Ramo de atividade do fornecedor.
FORTELEVENDAS	VarChar(11)				Tele vendas do fornecedor.
FORATIVO	VarChar(1)			*	Fornecedor ativo ou não.
FORTIPO	VarChar(1)			*	Tipo fornecedor, Lucro Real, Lucro Presumido, Simples Nacional, Produtor Rural, Pessoa Física, MEI.
FORINSRURAL	VarChar(15)				Inscrição rural.

Quadro 13 - Cadastro de Fornecedores (tbFornecedores)

O Quadro 14 descreve os campos da tabela que é criada a partir do recebimento dos produtos dos fornecedores, armazenando o código interno do produto no fornecedor.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
FORPRODID	Integer	*		*	Código do vínculo fornecedor/produto
FORPRODCODIGO	Integer	*		*	Código do produto do fornecedor.
PRODID	Integer		*	*	Código do produto.
FORID	Integer		*	*	Código do fornecedor.

Quadro 14 - Tabela de vínculo fornecedor/produto (tbFornProd)

O Quadro 15 descreve os campos do cadastro de funcionários da empresa, que se completam com os dados herdados da tabela pessoa.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
FUNID	Integer	*		*	Código do funcionário.
PESID	Integer	*	*	*	Código da pessoa.
CARID	Integer		*		Código do cargo.
FUNACESS	VarChar(1)			*	Acesso do funcionário no sistema.
USUID	Integer		*	*	Código do usuário.
FUNSITUACAO	VarChar(1)			*	Situação do funcionário.
FUNDATAADMI	DATE			*	Data de admissão do funcionário.
FUNNROCART	VarChar(40)				Número da carteira de trabalho.
FUNNROPIS	VarChar(40)				Número do PIS do funcionário.
FUNCARGAHOR	VarChar(40)				Carga horária do funcionário.
FUNSALARIOFIXO	Numeric(8,2)			*	Salário do funcionário.
FUNTIPOHABILIT	VarChar(10)				Tipo de habilitação do funcionário.

Quadro 15 - Cadastro de Funcionários (tbFuncionarios)

O Quadro 16 descreve os campos do cadastro das grades tributárias que irá facilitar o cadastro dos produtos, generalizando informações fiscais por essa divisão e NCM.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
GRADID	Integer	*		*	Código da grade tributária.
GRADATIVO	VarChar(1)			*	Indica se a grade esta ativa ou não.
GRADESCRICAÇÃO	VarChar(100)			*	Descrição da grade tributária
IPIID	Integer		*	*	Código do IPI.
ICMSID	Integer		*	*	Código do ICMS.
NCMID	Integer		*	*	Código do NCM.
PISCOFINSID	Integer		*	*	Código do PIS/COFINS.

Quadro 16 - Cadastro das grades tributárias dos produtos (tbGrades)

O Quadro 17 descreve os campos do cadastro dos grupos, utilizados para agrupamento dos produtos.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
GRUPID	Integer	*		*	Código do grupo de produtos.
GRUPDESCRICAÇÃO	VarChar(100)			*	Nome do grupo de produtos.

Quadro 17 - Cadastro de Grupos (tbGrupos)

O Quadro 18 descreve os campos do cadastro da tributação ICMS dos produtos.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
ICMSID	Integer	*		*	Código do ICMS.
ICMSDESCRICAÇÃO	VarChar(100)			*	Descrição do ICMS.
ICMSCSTENTRADA	Integer			*	Código CST de Entrada.
ICMSCSTSAIDA	Integer			*	Código CST de Saída.
ICMSALIQUENTRADA	Numeric(8,2)				Valor Percentual da Alíquota de Entrada.
ICMSALIQUAIDA	Numeric(8,2)				Valor Percentual da Alíquota de Saída.

Quadro 18 - Cadastro do ICMS (tbICMS)

O Quadro 19 descreve os campos do cadastro do valor do IPI dos produtos.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
IPIID	Integer	*		*	Código do IPI.
IPIDESCRICAÇÃO	VarChar(100)			*	Descrição do IPI.
IPIALIQOTA	Numeric(8,2)			*	Alíquota do IPI.

Quadro 19 - Cadastro do IPI (tbIPI)

O Quadro 20 descreve os campos dos itens da pré-venda.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
PRVID	Integer	*	*	*	Código da pré-venda
PRODID	Integer	*	*	*	Código do produto.

ITEMPVQTD	Numeric(8,3)			*	Quantidade do Produto.
ITEMPVVALORUN	Numeric(8,2)			*	Valor Unitário do Produto.
ITEMPVDESCPOR	Numeric(8,2)				Valor desconto unitário em porcentagem.
ITEMPVDESCVALOR	Numeric(8,2)				Valor desconto unitário em valor.
ITEMPVTOTALUN	Numeric(8,2)			*	Valor Total do Produto.

Quadro 20 - Itens da pré-vendas (tbItemPreVenda)

O Quadro 21 descreve os campos do cadastro de marcas dos produtos.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
MARID	Integer	*		*	Código da marca.
MARDESCRICAÇÃO	VarChar(100)			*	Descrição da marca.

Quadro 21 - Cadastro de Marcas (tbMarcas)

O Quadro 22 descreve os campos dos tipos de modelos fiscais existentes.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
MFID	Integer	*		*	Código dos modelos fiscais.
MFDESCRICAÇÃO	VarChar(100)			*	Descrições dos modelos fiscais.

Quadro 22 - Tabela dos Modelos Fiscais (tbModelosFiscais)

O Quadro 23 descreve os campos do cadastro do NCM.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
NCMID	Integer	*		*	Código do NCM.
NCMCODIGOSH	VarChar(8)			*	Código da Nomenclatura Comum do MERCOSUL (NCM).
NCMDESCRICAÇÃO	VarChar(100)			*	Descrição do NCM.

Quadro 23 - Cadastro do NCM (tbNCMSH)

O Quadro 24 descreve os campos de cadastro de Pessoas que faz a generalização de uma entidade, a qual pode se tornar ao mesmo tempo um cliente, fornecedor ou funcionário.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
PESID	Integer	*		*	Código da pessoa.
PESTIPO	VarChar(1)			*	Tipo da pessoa.
PESSEXO	VarChar(1)			*	Sexo da pessoa.
PESNOME	VarChar(100)			*	Nome da pessoa.
PESDATA CAD	DATE				Data do Cadastro no sistema.
PESDATANASC F	DATE				Data de Nascimento ou data de fundação.
PESCPF CNPJ	VarChar(14)			*	CPF ou CNPJ da pessoa.
PESRGIE	VarChar(15)			*	Inscrição estadual ou RG da pessoa.
PESFANTASIA	VarChar(15)			*	Fantasia.
PESEMAIL	VarChar(100)				E-mail da pessoa.
PESTELEFONE	VarChar(11)				Telefone da pessoa.
PESFAX	VarChar(11)				Fax da pessoa.

PESCELULAR	VarChar(11)				Celular da pessoa.
PESENDereco	VarChar(11)			*	Endereço da pessoa.
PESENDNRO	VarChar(10)			*	Número do endereço da pessoa.
PESENDCOMP	Varchar(50)			*	Complemento do endereço da pessoa.
PESCEP	VarChar(8)				CEP da pessoa.
PESBAIRRO	VarChar(50)			*	Bairro da pessoa.
PESNOMEPAI	VarChar(100)				Nome do pai.
PESNOMEMAE	VarChar(100)				Nome da mãe.
PESNATURALIDADE	VarChar(100)				Naturalidade da pessoa.
PEESTADOCIVIL	VarChar(1)				Estado civil.
PESCONJUGENOME	VarChar(100)				Nome cônjuge da pessoa.
PESDATANASCCONJ	DATE				Data de nascimento do conjugue.
PESNRODEPEN	Integer				Número de dependentes da pessoa.
PESRENDAFAM	Numeric(8,2)				Renda familiar da pessoa.
PESOBSEVACAO	VarChar(500)				Observação.
CIDID	Integer		*	*	Cidade da pessoa.

Quadro 24 - Cadastro de Pessoas (tbPessoas)

O Quadro 25 descreve os campos do cadastro da tributação PIS/COFINS dos produtos.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
PISCOFINSID	Integer	*		*	Código do PIS/COFINS.
PISCOFINSDESCRICA0	VarChar(100)			*	Descrição do PIS/COFINS.
PISCOFINSNATUREZA	Integer			*	Código da Natureza da Receita.
PISCOFINS CSTENT	Integer			*	Código CST de Entrada.
PISCOFINS CSTSAIDA	Integer			*	Código CST de Saída.
PISCOFINSALIENTPIS	Numeric(8,3)				Alíquota Entrada de PIS.
PISCOFINSALIENTCOF	Numeric(8,3)				Alíquota Entrada de COFINS.
PISCOFINSALISAIPIS	Numeric(8,3)				Alíquota Saída de PIS.
PISCOFINSALISAICOF	Numeric(8,3)				Alíquota Saída de COFINS.

Quadro 25 - Cadastro de PIS/COFINS (tbPISCOFINS)

O Quadro 26 descreve os campos armazenados no processo de pré-venda.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
PRVID	Integer	*		*	Código da pré-venda
PRVDATE	DATE			*	Data da pré-venda.
PESID	Integer		*		Código da Pessoa.
USUID	Integer		*	*	Código do Usuário.
FUNID	Integer		*		Código do funcionário.
PRVSITUACAO	Integer			*	Situação da pré-venda, finalizada ou não.
PRVTERMINAL	VarChar(500)			*	Identificação do terminal.
PRVOBSEVACAO	VarChar(500)				Observações da pré-venda.
PRVTOTALDESC	Numeric(8,2)			*	Valor Total sem desconto.
PRVLRDESCPORC	Numeric(8,2)				Valor Total do desconto em porcentagem.
PRVLRDESCVALOR	Numeric(8,2)				Valor Total do desconto em valor.
PRVTOTALCDESC	Numeric(8,2)			*	Valor Total com desconto.

Quadro 26 - Dado das Pré-Vendas (tbPreVenda)

O Quadro 27 descreve os campos do cadastro dos produtos.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
PRODID	Integer	*		*	Código do produto.
PRODATIVO	VarChar(1)			*	Indica se o produto esta ativo ou não.
PRODCODIGO	VarChar(30)			*	Código de barra do produto.
PRODESCRICA	VarChar(100)			*	Nome do produto.
PRODABREVIATURA	VarChar(100)			*	Abreviatura do nome do produto.
PRODDATACADASTRO	DATE			*	Data do cadastro do produto.
PRODDATAULTALT	DATE			*	Data da última alteração do produto.
PRODPELIQUIDO	Numeric(10,3)			*	Valor de peso líquido do Produto.
PRODPESOBRUTO	Numeric(10,3)			*	Valor de peso bruto do Produto.
PRODCOMISSAO	Numeric(2,2)				Valor percentual da comissão do Produto.
PRODTIPOFORMATO	VarChar(1)			*	Informa qual tipo de formato do item (Truncamento / Arredondamento)
PRODTIPOFAB	VarChar(1)			*	Informa qual tipo de fabricação do item (Própria / Terceirizada)
PRODPRECOABERTO	VarChar(1)			*	Indica se o produto pode ser vendido com preço em aberto.
GRADID	Integer		*	*	Código da grade tributária.
TIPPROID	Integer		*	*	Código do tipo do produto.
GRUPID	Integer		*	*	Código do grupo.
SUBGID	Integer		*		Código do subgrupo.
CADID	Integer		*	*	Código da cadeia de preço.
UNID	Integer		*	*	Código da unidade.
PRODCUSTO	Numeric(10,3)				Valor de custo do Produto.
PRODVENDAAVISTA	Numeric(10,3)			*	Valor de venda avista do Produto.
PRODLUCRO	Numeric(10,3)				Valor do percentual do lucro do Produto.
PRODESTOQUE	Numeric(10,3)				Quantidade do estoque final do Produto.
PRODESTOQUEMIN	Numeric(10,3)				Quant. do estoque mínimo do Produto.
PRODESTOQUERES	Numeric(10,3)				Quant. do estoque reservada do Produto.
PRODESTOQUEDISP	Numeric(10,3)				Quant. do estoque disponível do Produto.
MARID	Integer		*	*	Código da marca.

Quadro 27 - Cadastro de Produtos (tbProdutos)

O Quadro 28 descreve os campos do cadastro de representantes que será relacionado aos fornecedores.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
REPID	Integer	*		*	Código do representante.
REPNO	VarChar(100)			*	Nome do representante.
REPTLEFONE	VarChar(11)			*	Telefone do representante.
REPCELULAR	VarChar(11)			*	Celular do representante.
REPEMAIL	VarChar(100)				Email do representante.

Quadro 28 - Cadastro de Representantes (tbRepresentantes)

O Quadro 29 descreve os campos da numeração das séries fiscais.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
SFID	Integer	*		*	Código das séries fiscais.
SFDESCRICA	VarChar(100)			*	Descrição das séries fiscais.

Quadro 29 - Tabela de Séries fiscais (tbSeriesFiscais)

O Quadro 30 descreve os campos do cadastro dos subgrupos, o qual faz uma subdivisão dos grupos dos produtos.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
SUBGID	Integer	*		*	Código do subgrupo.
SUBGDESCRICAÇÃO	VarChar(100)			*	Descrição do subgrupo.
GRUPID	Integer		*	*	Código do Grupo.

Quadro 30 - Cadastro de Subgrupos (tbSubGrupos)

O Quadro 31 descreve os campos da tabela que armazena os tipos de produtos que podem ser cadastrados (finalidade de cada um).

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
TIPPROID	Integer	*		*	Código do tipo do produto.
TIPPRODESCRICAÇÃO	VarChar(100)			*	Descrição do tipo do produto.

Quadro 31 - Tabela dos Tipos de Produtos (tbTipoProduto)

O Quadro 32 descreve os campos das Unidades Federativas que divide o cadastro das cidades.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
UFID	Integer	*		*	Código da Unidade Federativa.
UFNOME	VarChar(100)			*	Nome da Unidade Federativa.
UFPAIS	VarChar(2)			*	Sigla do País da Unidade Federativa.
UFSIGLA	VarChar(2)			*	Sigla da Unidade federativa.

Quadro 32 - Tabela de UF (tbUF)

O Quadro 33 descreve os campos das Unidades de Medidas de comercialização dos produtos.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
UNID	Integer	*		*	Código da Unidade de Medida
UNSIGLA	VarChar(2)			*	Sigla da Unidade de Medida
UNNOME	VarChar(50)			*	Nome da Unidade de Medida.
UNVALOR	Integer			*	Valor da Unidade de Medida.

Quadro 33 - Tabela de Unidades de Medidas (tbUN)

O Quadro 34 descreve os campos do cadastro de usuários para acesso ao sistema.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
USUID	Integer	*		*	Código do usuário.
USULOGIN	VarChar(30)			*	Login do usuário.
USUSENHA	VarChar(30)			*	Senha do usuário.

USUTIPO	VarChar(1)				Tipo do usuário.
USUBLOQUIADO	VarChar(1)			*	Opção de bloqueio do usuário.

Quadro 34 - Cadastro de Usuário (tbUsuarios)

O Quadro 35 descreve os campos que são armazenados as informações da versão do sistema e do banco de dados.

Nome do Campo	Tipo (tamanho)	PK	FK	Não Nulo	Descrição
VERID	Integer	*		*	Código da versão do sistema.
VERDESCRICAÇÃO	VarChar(20)			*	Descrição da versão do sistema.
VERDATAAT	DATE			*	Data e hora da atualização.
VERBANCODADOS	VarChar(20)			*	Versão do Banco de dados.

Quadro 35 - Tabela de Versões do Sistema e Banco de Dados (tbVersao)

4.3 APRESENTAÇÃO DO SISTEMA

A linguagem de programação Delphi, descrita na subseção 3.1.5, foi utilizada no desenvolvimento para ambiente Windows do sistema proposto.

Para uma melhor apresentação dos requisitos funcionais do sistema a tela principal apresenta um menu com as seguintes opções: “Cadastro”, “Movimentações”, “Relatórios”, “Cópia de Segurança” e “Configurações”.

O submenu “Cadastro” apresenta as opções para realizar manutenção, inclusão, alteração e exclusão de registros, nas tabelas de dados do sistema: cidades, produtos, pessoas, clientes, funcionários entre outras. Já no submenu “Movimentações” encontram-se a opção de geração de uma pré-venda com seus respectivos dados e itens dos produtos. Por último, no submenu “Relatórios” encontram-se as opções para extrair as principais informações geradas pelo sistema. A relação com todas as funcionalidades implementadas pode ser observada no Quadro 36.

Funcionalidade (opção do menu)	Descrição
Cadastro	
1. Cidades	Formulário para realizar o cadastro das cidades conforme campos descritos na Tabela 1 .
2. Clientes	Formulário para realizar o cadastro de cargos conforme campos descritos na Tabela 4 .
3. Fornecedores	Formulário para realizar o cadastro de entregadores conforme campos descritos na Tabela 2 .
4. Funcionários	Formulário para realizar o cadastro de usuários conforme campos

	descritos na Tabela 5 .
5. Pessoas	Formulário para realizar o cadastro de produtos conforme campos descritos na Tabela 6 .
6. Usuários	Formulário para realizar o cadastro de fornecedores conforme campos descritos na Tabela 9 .
7. Grupos	Formulário para realizar o cadastro de clientes conforme campos descritos na Tabela 3 .
8. Sub-Grupos	Formulário para realizar o cadastro da empresa conforme campos descritos na Tabela 13 .
9. Produtos	Formulário para realizar o cadastro de observações conforme campos descritos na Tabela 12 .
10. Unidades de Medidas	Formulário para realizar o cadastro de observações conforme campos descritos na Tabela 12 .
11. Grades	Formulário para realizar o cadastro de observações conforme campos descritos na Tabela 12 .
12. Marcas	Formulário para realizar o cadastro de observações conforme campos descritos na Tabela 12 .
13. Convênios	Formulário para realizar o cadastro de observações conforme campos descritos na Tabela 12 .
14. Cargos	Formulário para realizar o cadastro de observações conforme campos descritos na Tabela 12 .
15. Representantes	Formulário para realizar o cadastro de observações conforme campos descritos na Tabela 12 .
Movimentações	
1. Lançar Pré-Venda	Formulário para realizar o registro de uma compra conforme campos descritos na Tabela 10 e registros de itens compras conforme campos descritos na Tabela 11 .
Relatórios	
1. Produtos	Formulário para exibir dados dos produtos cadastrados conforme campos descritos na Tabela 6 .
2. Fornecedores	Formulário para exibir dados dos fornecedores cadastrados conforme campos descritos na Tabela 9 .
3. Clientes	Formulário para exibir dados dos clientes cadastrados conforme campos descritos na Tabela 3 .
4. Pedidos	Formulários para exibir dados de diversos pedidos registrados, a critério solicitado pelo usuário, conforme campos descritos na Tabela 7 e registros de itens pedidos conforme descritos na Tabela 8 .
Sair (verifica se a opção do usuário é finalizar a aplicação, opção "Sim")	

Quadro 36 - Relação das funcionalidades implementadas

O formulário (ou interface) para realizar a manutenção no cadastro de clientes e pessoas, mostrado na Figura 13, foi construído a partir do formulário padrão "Cadastro", descrito na subseção 3.2.2. Esse formulário permite o cadastro de pessoas que podem ser clientes ou fornecedores. É possível no momento da realização do cadastro fazer a verificação se a pessoa que está sendo cadastrada já possui registro na tabela "Pessoas". Isso é realizado por meio do botão com a descrição "... " (reticências) que na segunda tela de pesquisa pode ser visualizada.

Figura 13 - Cadastro de Clientes

Na Figura 13 estão marcadas e identificadas, com cinco retângulos, as principais partes encontradas em todos os formulários de cadastros: “Cidades”, “UF”, “Usuários”, “Produtos”, “Fornecedores”, “Clientes” entre outros. Essas partes estão descritas a seguir:

- A. Botões de manutenção: com componente BitBtn alinhados em um *panel* foram disponibilizadas as opções “Novo”, “Editar”, “Salvar”, “Excluir”, “Cancelar” para manutenção na tabela “tbPessoas” e “tbClientes”.
 - a. Novo: Cria um código novo para a inclusão por meio do auto incremento (acrescenta um) do código do último registro armazenado na tabela de dados, tanto na tabela “Pessoas” como na tabela “Clientes”, habilitando a região dos dados.
 - b. Salvar: Faz a gravação, em operações de inclusão ou alteração, dos dados informados pelo usuário, armazenando todos os campos na tabela “Clientes” e “Pessoas”.
 - c. Excluir: Exclui (apaga) o registro selecionado e deixa a tela no estado inicial, excluindo somente o registro do cliente, deixando os dados da pessoa disponíveis para outras operações.

- d. Cancelar: Cancela a operação de inclusão ou de alteração que estava sendo realizada e desabilita a tela para edição.
 - e. Fechar: Fecha a janela e retorna ao menu principal da aplicação.
- B. Região de dados: com componentes DBEdit foram selecionados os campos da tabela “Clientes”, configurando as propriedades: DataSource e DataField, para inclusão ou alteração dos dados. O componente DBLookupComboBox foi utilizado para apresentar todos os registros de convênios. Os componentes *Label* foram utilizados para rotular e informar ao usuário final quais informações ou campos cada componente de entrada está associado. Os componentes ComboBox foram utilizados para algumas operações que no banco de dados serão armazenados apenas alguns caracteres.
- C. Localizar um registro durante inserção ou alteração: o botão identificado por reticências (“...”) é utilizado para chamar o formulário de pesquisa de pessoas, podendo assim durante a realização de um cadastro localizar o registro de determinada pessoa no próprio formulário. Caso a pessoa localizada já esteja cadastrada como cliente, o componente Panel utilizado como barra de *status* mudará a propriedade para “Editando registro”.O registro poderá ser editado normalmente.
- D. Botões de pesquisa: por meio do componente BitBtn alinhados em um componente *Panel* foram disponibilizadas as opções “Pesquisar”, “Primeiro”, “Anterior”, “Próximo”, “Último” e “Fechar” para facilitar a pesquisa no formulário de cadastro de clientes, sendo que a pessoa relacionada ao cliente será localizada por meio de qualquer uma dessas operações.
- a. Pesquisar: Traz na tela o formulário de pesquisa clientes.
 - b. Primeiro: Posiciona o registro no primeiro cliente da tabela.
 - c. Anterior: Posiciona o registro pegando o anterior ao sendo apresentado na tela.
 - d. Próximo: Posiciona o registro no próximo registro ao apresentado na tela.
 - e. Último: Posiciona o registro no último inserido.
 - f. Fechar: Fecha a janela e retorna ao menu principal da aplicação.
- E. Painel de Status: Painel que informa o usuário sobre qual atividade está sendo realizada com os seguintes textos: “Procurar Registros”, “Editando um Registro!” e “Inserindo novo Registro!”

O formulário (ou interface) para realizar a pesquisa na tabela clientes, mostrado na Figura 14, foi construído a partir do formulário padrão “Pesquisa”, descrito na subseção 3.2.2. Esse formulário apresenta ao usuário a possibilidade de efetuar a pesquisa de um cliente, trazendo as informações relativas à tabela “Clientes”. Isso é realizado pelo clique na coluna da tabela clientes que ele deseja pesquisar. Devido o sistema ser desenvolvido para funcionamento em rede torna-se assim possível atualizar a região dos dados contidos no item C.

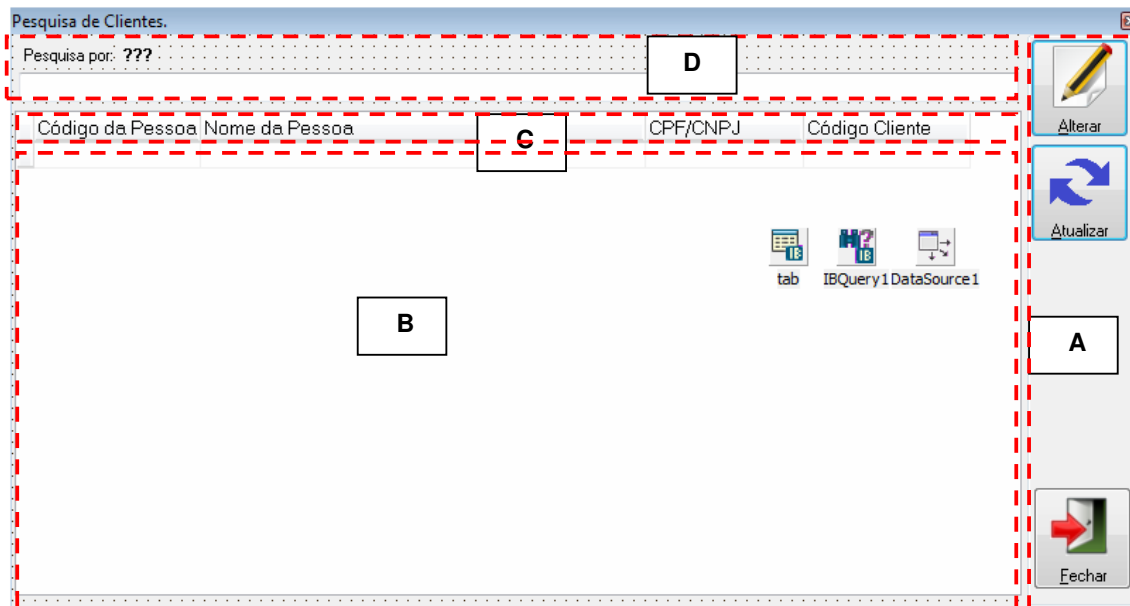


Figura 14 - Pesquisa de Clientes

As partes destacadas na imagem da Figura 14 são descritas a seguir:

- A. Botões de manutenção: com o componente BitBtn alinhados em um Panel foram disponibilizadas as opções “Novo”, “Editar”, “Salvar”, “Excluir”, “Cancelar” para manutenção na tabela “tbPessoas” e “tbClientes”.
 - a. Alterar: Envia o registro selecionado para o formulário de cadastro de clientes.
 - b. Atualizar: Atualiza os dados referentes à tabela clientes, podendo ou não estar filtrados.
 - c. Fechar: Fecha a janela e retorna ao menu principal da aplicação.
- B. Região de dados: utilizando componentes Dbgrid são listados os registros referentes à tabela tbClientes.

- C. Localizar um registro de determinada coluna: Com um clique no título de cada coluna é possível alterar o tipo de pesquisa. A coluna que será pesquisada ficará em com o título em negrito.
- D. Pesquisa: Ao digitar o texto no componente Edit é realizado o filtro nos registros da tabela. A propriedade caption do componente Label será alterada para o nome da coluna que estiver sendo realizada a pesquisa..

O formulário (ou interface) para realizar a pesquisa na tabela pessoas, mostrado na Figura 15, foi construído a partir do formulário padrão “Pesquisa”, descrito na subseção 3.2.2.

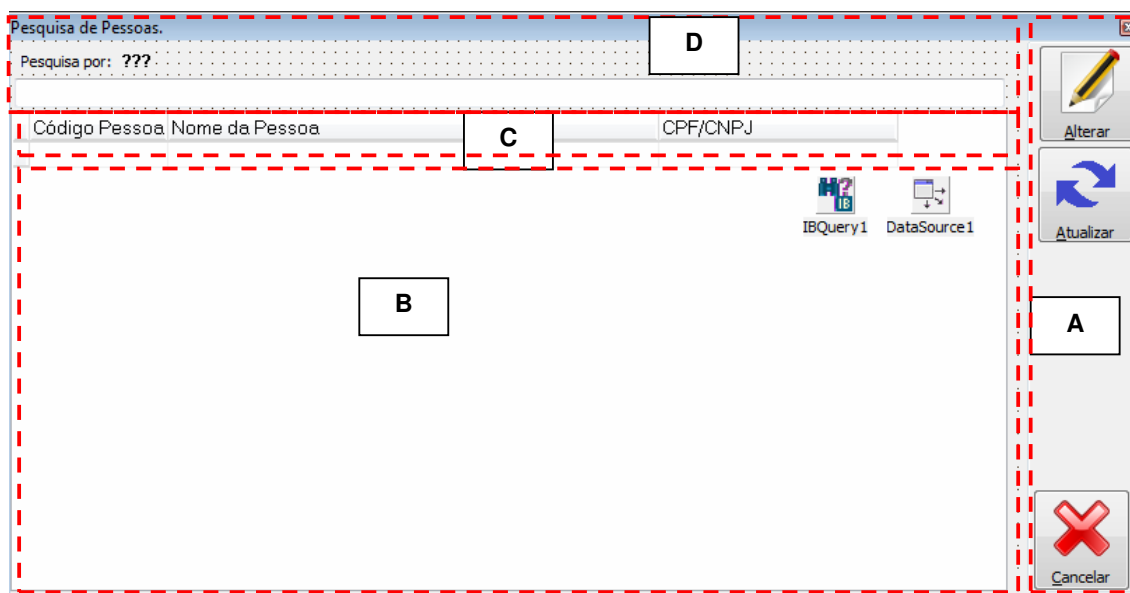


Figura 15 - Pesquisa de Pessoas

O formulário (ou interface) para realizar a manutenção no cadastro de pessoas, mostrado na Figura 16, foi construído a partir do formulário padrão “Cadastro”, descrito na subseção 3.2.2. Esse formulário possibilita efetuar o cadastro de uma pessoa fazer alterações e pesquisar as pessoas já cadastradas.

The image shows a screenshot of a software application window titled "Cadastro de Pessoas". The window contains a form for entering personal data. The form is divided into several sections: "Dados Pessoais" (Name, CPF, RG, Data Nascimento, Cidade, Sexo, Endereço, Complemento), "Contato" (Telefone, Fax, Celular), and "Informações Pessoais" (Email). The form is surrounded by a red dashed border. On the left side, there is a vertical toolbar with buttons for "Novo", "Editar", "Salvar", "Cancelar", and "Excluir". On the right side, there is a vertical toolbar with buttons for "Pesquisar", "Primeiro", "Anterior", "Próximo", "Último", and "Fechar". At the bottom of the form, there is a "Procurar Registros" button. The form is labeled with letters A, B, and C, and the search button is labeled with D.

Figura 16 - Cadastro de Pessoas

Na Figura 16 estão marcadas e identificadas, com retângulos pontilhados, as principais partes encontradas em todos os formulários de cadastros: “Cidades”, “UF”, “Usuários”, “Produtos”, “Fornecedores”, “Clientes” entre outros. Essas partes estão descritas a seguir:

- A. Botões de manutenção: com o uso do componente BitBtn alinhados em um Panel foram disponibilizadas as opções “Novo”, “Editar”, “Salvar”, “Excluir”, “Cancelar” para manutenção na tabela “tbPessoas” e “tbClientes”.
 - a. Novo: Cria um código novo para a inclusão por meio do auto-incremento (acrescenta um) do código do último registro armazenado na tabela de dados, na tabela “Pessoas”.
 - b. Salvar: Faz a gravação, em operações de inclusão ou alteração, dos dados informados pelo usuário, armazenando todos os campos na tabela “Pessoas”.
 - c. Excluir: Apaga (ou deleta) o registro selecionado e deixa a tela no estado inicial, deletando o registro de “Pessoas”.
 - d. Cancelar: Cancela a operação de inclusão ou de alteração que estava sendo realizada e desabilita a tela para edição.
 - e. Fechar: Fecha a janela e retorna ao menu principal da aplicação.
- B. Região de dados: com componentes DBEdit foram selecionados os campos da tabela “Clientes”, configurando as propriedades: DataSource e DataField, para

inclusão ou alteração dos dados. O componente DBLookupComboBox foi utilizado para apresentar todos os registros de convênios. Os componentes Label foram utilizados para rotular e informar ao usuário final quais informações ou campos cada componente de entrada está associado. O componentes ComboBox foram utilizados para algumas operações que serão armazenadas no banco de dados com apenas alguns caracteres.

- C. Botões de pesquisa: com o uso do componente BitBtn alinhados em um Panel foram disponibilizadas as opções “Pesquisar”, “Primeiro”, “Anterior”, “Próximo”, “Último” e “Fechar” para facilitar a pesquisa no formulário de cadastro de clientes, sendo que a pessoa relacionada ao cliente será localizada por meio de qualquer uma dessas operações.
- a. Pesquisar: Traz na tela o formulário de pesquisa de clientes.
 - b. Primeiro: Posiciona o registro no primeiro cliente da tabela.
 - c. Anterior: Posiciona o registro pegando o anterior ao da tela.
 - d. Próximo: Posiciona o registro pegando o próximo ao da tela.
 - e. Ultimo: Posiciona o registro no último inserido.
 - f. Fechar: Fecha a janela e retorna ao menu principal da aplicação.
- D. Painel de Status: Painel que informa o usuário sobre qual atividade está sendo realizada com os seguintes textos: “Procurar Registros”, “Editando um Registro!” e “Inserindo novo Registro!”.

O formulário (ou interface) para realizar a manutenção no cadastro de pessoas, mostrado na Figura 17.

Figura 17 - Lançamentos de Pré-Venda

Na Figura 17 estão marcadas e identificadas, com seis retângulos, as principais partes encontradas no formulário de pré-venda, que serão descritas a seguir:

A. Botões de manutenção e Pesquisa: com o uso do componente BitBtn alinhados em um componente *Panel* foram disponibilizadas as opções “Novo”, “Editar”, “Salvar”, “Excluir”, “Cancelar” para manutenção na tabela “tbPreVenda”, foram disponibilizadas as opções “Primeiro”, “Anterior”, “Próximo”, “Último” e “Fechar” para facilitar a pesquisa no formulário de lançamento e pré-venda, sendo que a navegação entre as pré-vendas poderão serem efetuadas.

- a. Novo: Cria um código novo para a inclusão por meio do auto-incremento (acrescenta um) do código do último registro armazenado na tabela de dados, na tabela “tbPreVenda”.
- b. Salvar: Faz a gravação, em operações de inclusão ou alteração, dos dados informados pelo usuário, armazenando todos os campos na tabela “Pessoas”.
- c. Excluir: Apaga (ou deleta) o registro selecionado e deixa a tela no estado inicial, excluindo o registro de “Pessoas”.

- d. Cancelar: Cancela a operação de inclusão ou de alteração que estava sendo realizada e desabilita a tela para edição.
 - e. Primeiro: Posiciona o registro no primeiro cliente da tabela.
 - f. Anterior: Posiciona o registro pegando o anterior ao da tela.
 - g. Próximo: Posiciona o registro pegando o próximo ao da tela.
 - h. Último: Posiciona o registro no último inserido.
- B. Dados da Pré-Venda: por meio de componentes DBLookupComboBox foram exibidos o usuário relacionado com a Pré-venda e também o vendedor. Um componente Edit permite realizar pesquisa por código de registro de “Pessoas” e “Funcionários”, descritos na pré-venda na função de vendedor. Por meio do evento onKeyPress do DBLookupComboBox da “Pessoa” é possível chamar o formulário de pesquisa de pessoas.
- C. Exibir tipo de pessoa: com o componente IBQuery é possível executar comandos na linguagem SQL verificando quais os tipos de cadastros cada pessoa possui, exibindo com a opção checked do componente CheckBox, sendo que um pessoa pode ser um Cliente, Funcionário ou Fornecedor. Nada impede que cada pessoa seja mais que uma opção ao mesmo tempo.
- D. Itens da Pré-Venda: Utilizando um componente Edit é possível localizar um produto pelo código ou pelo evento onKeyPress abrindo o formulário de pesquisa de produtos. Após a escolha o usuário é redirecionado ao Edit da quantidade, posteriormente o de valor, até que o item seja inserido no grid.
- E. Informações Pré-Venda: Exibir os totais da pré-venda, também permitindo (após a pré-venda ser salva) a geração em forma de relatório da pré-venda, possibilitando a impressão, ou salvar em outras formas de mídia.
- F. Botões de pesquisa e salvar: Os botões de pesquisa permitem localizar pré-vendas já salvas. Nessa área da tela é possível fazer outras operações relacionadas à pré-venda.
- a. Pesquisar: Traz na tela o formulário de pesquisa de pré-venda.
 - b. Salvar: Finaliza a pré-venda salvando as informações.
 - c. Fechar: Fecha o formulário de pré-venda.

O formulário (ou interface) para realizar a pesquisa tabela “Pessoas”, mostrado na Figura 18, foi construído a partir do formulário padrão “Pesquisa”, descrito na subseção 3.2.2.

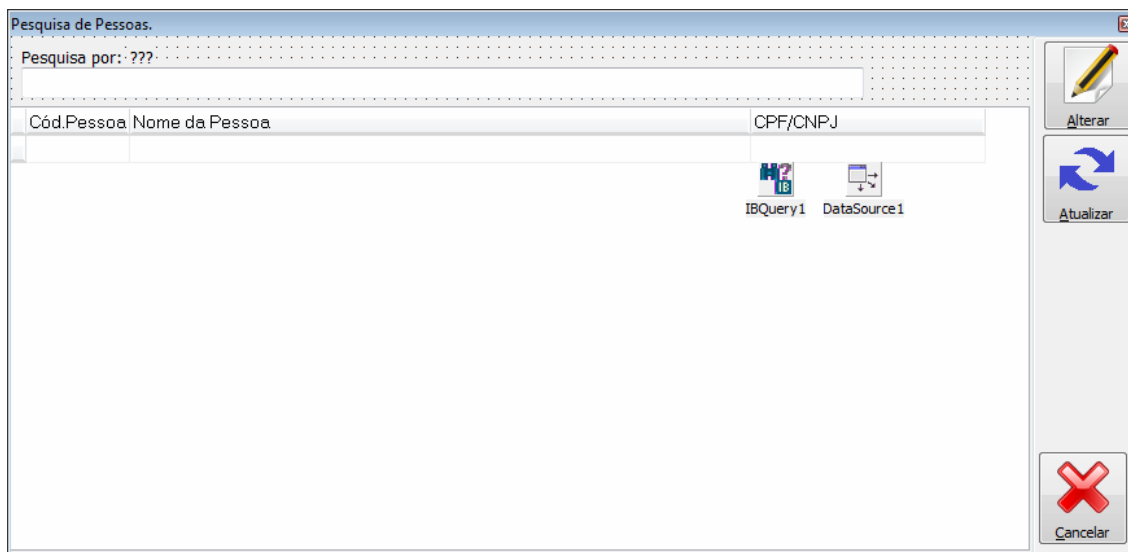


Figura 18 - Pesquisa de Pessoas na Pré-Venda

O formulário (ou interface) para realizar a pesquisa na tabela “Produtos”, mostrado na Figura 19, foi construído a partir do formulário padrão “Pesquisa”, descrito na subseção 3.2.2. Esse formulário possibilita ao usuário localizar um registro de um produto e deixar selecionado no formulário de Lançamento de pré-vendas.

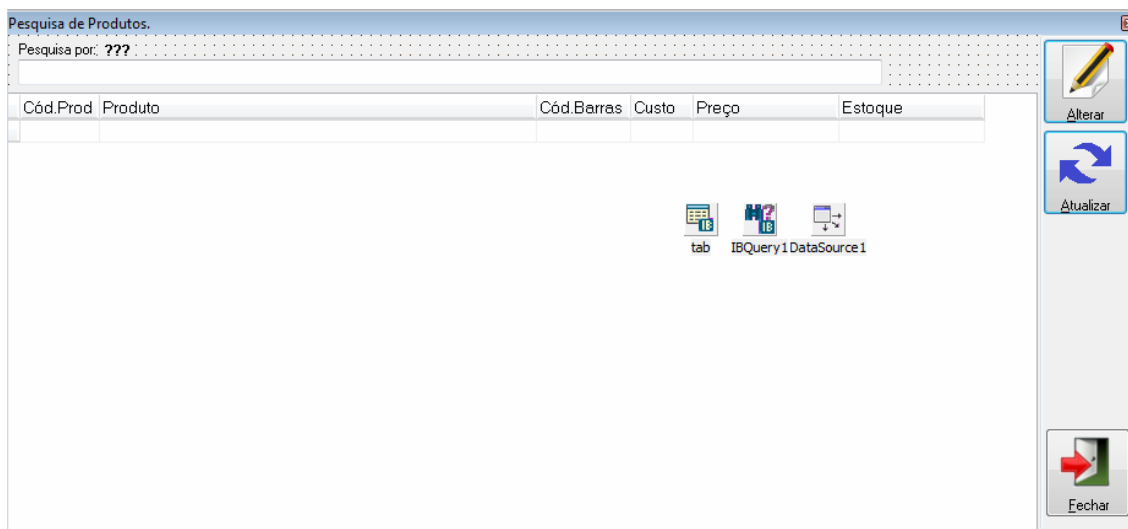


Figura 19 - Pesquisa de Produtos na Pré-Venda

O formulário (ou interface) para exibir informações sobre as pré-vendas já cadastradas é apresentado na Figura 20.

ReportTitle: ReportTitle1					
Relatório de Pré-Vendas	Data: [Date]				
GroupHeader: GroupHeader2					
[frxDBData] [frxDBDataset1."PE SNAME"]					
GroupHeader: GroupHeader1					
[frxDBDataset1."PRVID"]					
Código Pré-Venda: [frxDBDataset1."PR"] Data Pré-Venda: [frxDBDataset1.""]					
Cód.Produto	Descrição Produto	Quant.	Valor Unitário	Desc %	Valor Total
MasterData: MasterData1		frxDBDataset1			
[frxDBData]	[frxDBDataset1."PRODESCRICA0"]	[frxDB]	[frxDBDataset]	[frxD]	[frxDBData]
GroupFooter: GroupFooter1		Valor Total:		[SUM(<frxDBDat	
GroupFooter: GroupFooter2		Valor Total Cliente:		[SUM(<frxDBDat	
PageFooter: PageFooter1				[Page#]	

Figura 20 - Relatório de Pré-Venda

Os principais componentes utilizados para a composição do relatório da Figura 20 são:

- Componente frxReport: por meio deste componente é possível criar um novo relatório e definir as bandas do mesmo.
- Componente frxDBDataSet: componente utilizado para fazer a conexão entre a qryPreVenda com o frxReport.
- Grupo Pessoa (banda GroupHeader, propriedade Expression campo "PESID"): define que as Pré-Vendas serão agrupadas por Pessoa.
- Grupo Pré-Venda (banda GroupHeader, propriedade Expression campo "PRVID"): define que os produtos serão agrupados por pré-venda. Nesta banda também estão identificados os títulos das colunas do relatório: "Cód.Produto", "Descrição Produto", "Quantidade", "Valor Unitário", "Desc %" e "Valor Total".
- Detalhe ou corpo (MasterData): apresenta as informações dos produtos de uma pré-venda com o uso do componente Memo.
- Rodapé da Pré-Venda (banda GroupFooter): apresenta o total do pedido (soma de todos os produtos do pedidos).

G. Rodapé da Pessoa (banda GroupFooter): apresenta o total da pessoa (soma de todas as pré-vendas das pessoas).

H. Sumário ou resumo (PageFooter): apresenta o número de páginas do relatório.

O formulário (ou interface) desenvolvido para realizar a cópia de segurança (Figura 21) do banco de dados, foi feito visando que o usuário possa armazenar o *backup* no caminho ou dispositivo desejado, informado por meio de um Edit ou no botão 'Localizar Caminho'.

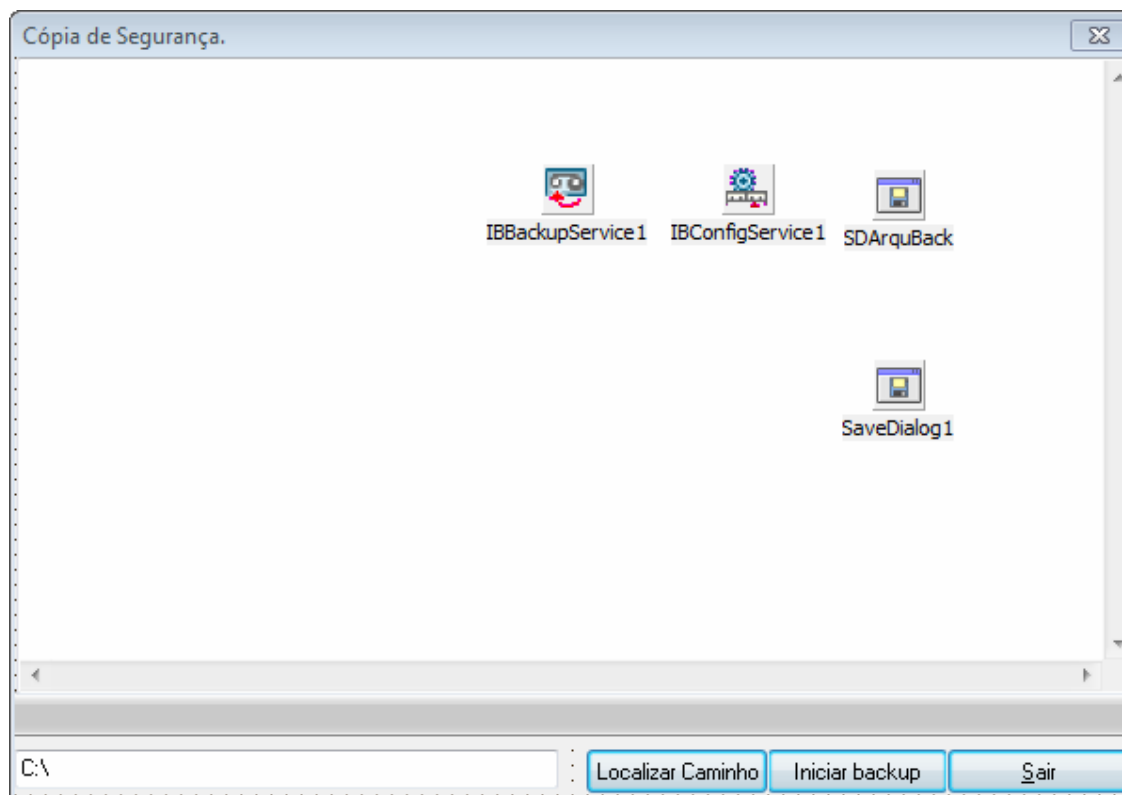


Figura 21 - Formulário de Backup

O formulário (ou interface) desenvolvido para realizar a restauração do banco de dados, apresentado na Figura 22, tem como princípio a localização do banco de dados, a partir do caminho onde está armazenada a cópia de segurança. Ao clicar no botão "Restaurar Banco" o arquivo de *backup* será convertido em um banco de dados.

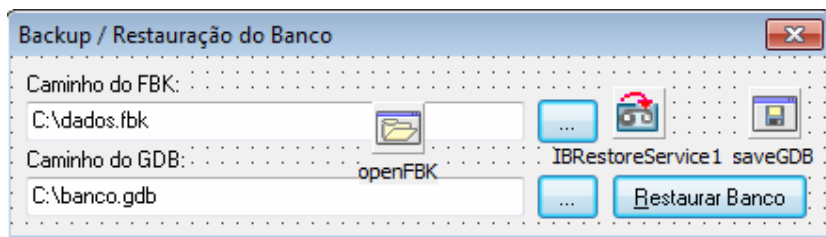


Figura 22 - Formulário de Restauração de Dados

4.4 IMPLEMENTAÇÃO DO SISTEMA

No desenvolvimento do projeto proposto foi possível destacar alguns aspectos importantes empregados na programação, como por exemplo: utilização de atributos públicos para indicar a tabela a ser usada no formulário padrão de “Cadastro”, descrito na subseção 3.2.2.

A Listagem 1 apresenta os trechos de códigos da implementação dos formulários.

```
//interface da classe do formulário para cadastro de Clientes
type
  TfrClientes = class(TForm)
    ...
  public
    Tabela: TIBTable;
    CampoChave: String;
    CampoLocaliza: String;
  end;

//invocando o formulário para cadastro de Clientes
procedure TfrPrincipal.Pessoas1Click(Sender: TObject);
begin
  frClientes := TfrClientes.Create(nil);
  frClientes.Tabela := md.tbClientes;
  frClientes.CampoChave := 'ClienteID';
  frClientes.CampoLocaliza := 'NomeCliente';
  frClientes.ShowModal;
  frClientes.Free;
end;
```

Listagem 1 - Código utilizado na chamada do cadastro de clientes

Ainda na Listagem 1 é possível observar dois aspectos de programação necessários para indicar a tabela de dados a ser usada no código do formulário, são eles:

- a. interface da classe do formulário para cadastro de cliente sendo possível observar os atributos públicos (public) que indicam a tabela de dados

(Tabela), o campo chave (CampoChave) e o campo utilizado para localizar um registro (CampoLocaliza);

- b. código utilizado para invocar (ou chamar) o formulário implementado para realizar o cadastro de clientes (frClientes).

A Listagem 2 apresenta como o código descrito no Quadro 1 foi adaptado para o contexto dos formulários padrão desenvolvidos no projeto.

```
//interface da classe do formulário para cadastro de Pesquisa Padrão
type
  TfrPesquisa = class(TForm)
    ...
  public
    CampoChave: String;
    CampoLocaliza: String;
    ColumnAux: TColumn;
    ...
  end;

//invocando o formulário de pesquisa com o formulário de Clientes
procedure TfrCadastroCliente.btnPesquisarClick(Sender: TObject);
begin
  inherited;
  frPesquisaClientes:= TfrPesquisaClientes.Create(nil);
  frPesquisaClientes.tab := Tabela;
  frPesquisaClientes.CampoChave := 'CLIID';
  frPesquisaClientes.CampoLocaliza := 'PESNOME';
  frPesquisaClientes.ShowModal;
  frPesquisaClientes.Free;
  funcaoAjustes();
end;
```

Listagem 2 - Código utilizado no formulário padrão para pesquisa e no formulário de cadastro de clientes

Na Listagem 2 é possível observar dois aspectos de programação necessários para indicar a tabela de dados a ser usada no código do formulário, são eles:

- a. interface da classe do formulário cadastro padrão sendo possível observar os atributos públicos (*public*) que indicam a tabela de dados (Tabela), o campo chave (CampoChave) e o campo utilizado para localizar um registro (CampoLocaliza);
- b. código utilizado para invocar (ou chamar) o formulário implementado para realizar a pesquisa de clientes(frPesquisaClientes).

A Listagem 3 apresenta o código utilizado para trabalhar com os dados apresentados em componentes do tipo ComboBox.


```

//interface utilizada para manipulação do combobox
interface

type
  Tfiltro = class
  private
  protected
  public
    procedure      QueryToComboBox(cb:TComboBox;      qry:TIBQuery;
campo:String; campoCodigo:String);
    function PegaCodigo(item:Integer):Integer;
    procedure      SelecionaCombo(codigo:      Integer;      campo:String;
cb:TComboBox; qryLocaliza: TIBQuery);
  published
  end;

implementation
var
  listCod:TStringList;
{ Tfiltro }
function Tfiltro.PegaCodigo(item: Integer): Integer;
begin
  Result := StrToInt(listCod.Strings[item]);
end;

procedure Tfiltro.QueryToComboBox(cb: TComboBox; qry: TIBQuery;
campo:String; campoCodigo:String);
begin
  qry.Open;
  cb.Items.Clear;
  listCod := TStringList.Create();
  qry.First;
  while not qry.Eof do
  begin
    cb.Items.Add(qry.FieldName(campo).AsString);
    listCod.Add(IntToStr(qry.FieldName(campoCodigo).AsInteger));
    qry.Next;
  end;
  qry.Close;
end;

procedure Tfiltro.SelecionaCombo(codigo: Integer; campo: String; cb:
TComboBox;
  qryLocaliza: TIBQuery);
begin
  qryLocaliza.ParamByName('codigo').AsInteger := codigo;
  qryLocaliza.Open;
  cb.Text := qryLocaliza.FieldName(campo).AsString;
  qryLocaliza.Close;
end;

end.

```

Listagem 3 - Código utilizado no formulário cadastro de subgrupos

Na Listagem 3 é possível destacar os aspectos de programação utilizados como demonstração no cadastro de SubGrupos, porém com uma classe genérica, podendo ser utilizado em todos os outros formulários.

- a. interface da classe do formulário cadastro padrão sendo possível observar os atributos públicos (*public*) que indicam a tabela de dados (Tabela), o campo chave (CampoChave) e o campo utilizado para localizar um registro (CampoLocaliza).

A Listagem 4 apresenta os trechos de códigos usados para implementação do formulário de *backup*.

```
//Procedimento criado para realizar o backup
procedure TfrBackup.BTGeneraBackClick(Sender: TObject);
var
  configurar:TIniFile;
  configurado:String;
begin
  configurar := TIniFile.Create('BD\configurar.ini');
  try
    Mem1.Clear;
    BTGeneraBack.Enabled := False;
    with IBackupService1 do
    begin
      md.db.Connected := False;
      DatabaseName      :=      configurar.ReadString('Configuracoes',
'caminho', '');
      ServerName := 'localhost';
      BackupFile.Clear;
      BackupFile.Add(caminho.Text+'dados.FBK');
      Protocol := TCP;
      Params.Clear;
      Params.Add('user_name=SYSDBA');
      Params.Add('password=masterkey');
      Active := True;
      Application.ProcessMessages;
      try
        ServiceStart;
        ProgressBar1.Min := 0;
        ProgressBar1.Max := 10;
        while not Eof do
        begin
          ProgressBar1.StepBy(1);
          Mem1.Lines.Add(GetNextLine);
        end;
      finally
        end;
        Active := False;
        ShowMessage('Backup Efetuado Com Sucesso!');
      end;
    except
      on E: Exception do
        begin
```

```

        ShowMessage('Backup Não Pode Ser Efetuado!');
        md.db.Connected := True;
    end;
end;
end;

```

Listagem 4 - Código utilizado na realização da cópia de segurança

Ainda na Listagem 4 é possível observar dois aspectos de programação necessários para fazer a localização do banco no arquivo de configuração e realizar o *backup* do banco de dados.

- a. No arquivo de configuração é localizado o caminho do banco de dados, sendo que ao iniciar o sistema o mesmo é configurado, logo após os outros parâmetros são passados e por fim o usuário recebe a informação se o *backup* foi efetuado ou não.

A Listagem 5 apresenta a forma utilizada para realizar a restauração do banco de dados.

```

//restauração do banco de dados
with IBRestoreService1 do
begin
    ServerName := 'localhost';
    LoginPrompt := False;
    Params.Add('user_name=SYSDBA');
    Params.Add('password=masterkey');
    Active := True;
    try
        Verbose := True;
        Options := [Replace, UseAllSpace];
        PageBuffers := 3000;
        PageSize := 4096;
        DatabaseName.Add(edtGDB.Text);
        BackupFile.Add(edtFbk.Text);
        ServiceStart;
    finally
        showmessage('Restauração realizada com sucesso!!');
        Active := False;
    end;
end;

```

Listagem 5 - Código utilizado para fazer a restauração do banco de dados.

Na Listagem 5 está a codificação realizada para fazer a restauração do banco de dados. O componente IBRestoreService é utilizado e mais as informações de localização a qual o banco de dados será gravado e a localização do arquivo de *backup*.

5 CONCLUSÃO

O desenvolvimento de um sistema para gerenciar empresas varejistas é importante devido percebida que empresas têm de organizar de forma rápida e segura os seus dados operacionais. Esses dados estão envolvidos no processo administrativo e comercial dessas empresas. O sistema desenvolvido como resultado deste trabalho pode ser utilizado por empresas varejistas de diversos ramos de atividades como confecções, calçados, brinquedos entre outros.

O processo que se destaca no sistema desenvolvido é o da pré-venda. Essa operação ocorre quando um cliente solicita um orçamento, uma condicional ou uma venda. Essa operação é importante para a comercialização de todos os produtos da empresa usuária do sistema, para garantir e aumentar a receita, ou projetar novas vendas.

A linguagem de programação Delphi se mostrou adequada para alcançar os objetivos propostos. Os componentes facilitaram o processo de desenvolvimento de software, exemplificando: a) utilização dos componentes das páginas “Data Access” e “Data Controls” para controle dos dados; b) projeto do banco de dados implementado usando o Firebird e o acesso as tabelas de dados por meio dos componentes “IBTable” e “IBQuery” da página “Interbase”; e, c) componentes “FastReport” disponíveis na página “FastReport 4.0” para a geração de relatórios.

Todas as funcionalidades levantadas para o sistema no processo de análise foram implementadas, com o uso dos recursos do ambiente de desenvolvimento Delphi. Permitindo, assim, alcançar o objetivo geral de desenvolver um software gerencial para empresas varejistas, destacando a operação de pré-venda.

REFERÊNCIAS

ANDRADE, Claudenir Campos. **Automação comercial com VB.NET e C#**. São Paulo: Editora Érica 2002.

BRASIL. **Mapa das micro e pequenas empresas**. Disponível em: <http://www.brasil.gov.br/empreendedor/empreendedorismo-hoje/o-mapa-das-micro-e-pequenas-empresas>. Acesso em: 18 ago. 2013.

CANTU, Carlos Henrique. **Firebird essencial**. Rio de Janeiro: Editora Ciência Moderna 2005.

DEZIGN. **DeZign for databases**. Disponível em <http://www.datanamic.com/dezign/index.html>>. Acesso em: 29 jul. 2013.

FIREBIRD. **Ferramentas administrativas**. Disponível em: <http://www.firebaseio.com.br/fb/downloads.php?categ=2>. Acesso em: 21 mar. 2013.

LAUDON, Kenneth C.; LAUDON, Jane Prince. **Sistemas de informação**. Rio de Janeiro: LTC - Livros Técnicos e Científicos S. A., 1999.

LEVENEZ, Eric. **Languages**, 2013. Disponível em: <http://www.levenez.com/lang/>>. Acesso em: 13 ago. 2013.

MANZANO, José Augusto N.G; MENDES, Sandro Santa. **Estudo dirigido de Delphi 5.0**. São Paulo: ÉRICA, 1999.

OLIVEIRA, Djalma de Pinho Rebouças de. **Sistemas, organização e métodos: uma abordagem gerencial**. 19ª ed. São Paulo: Editora Atlas, 2010.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistema de banco de dados**. 3. ed. São Paulo: Makron, 1999.

TAKAI, Osvaldo Kotaro; ITALIANO, Isabel Cristina; FERREIRA, João Eduardo. **Introdução a banco de dados**. 1995. Disponível em: <http://www.ime.usp.br/~jef/apostila.pdf>>. Acesso em: 4 abr. 2013.

VISUAL PARADIGM. **Visual Paradigm**. Disponível em <http://www.visual-paradigm.com/>>. Acesso em: 7 ago. 2013.

WAZLAWICK, Raul Sidnei. **Análise e projeto de sistemas de informação orientados a objetos**. 2ª ed. Rio de Janeiro: Editora Elsevier 2011.