

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

**BRUNA SCHEFFER DIAS
IURI MENIN**

**SISTEMA PARA GERENCIAMENTO DE
UMA LOJA DE CONFECÇÕES**

TRABALHO DE CONCLUSÃO DE CURSO

**PATO BRANCO
2013**

**BRUNA SCHEFFER DIAS
IURI MENIN**

**SISTEMA PARA GERENCIAMENTO DE
UMA LOJA DE CONFECÇÕES**

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: Profa. Beatriz T. Borsoi

**PATO BRANCO
2013**

ATA Nº: 208

DEFESA PÚBLICA DO TRABALHO DE DIPLOMAÇÃO DOS ALUNOS BRUNA SCHEFFER DIAS e IURI MENIN.

Às 10:25 hrs do dia 18 de abril 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 Vinicius Pegorini (Convidado), para avaliar o Trabalho de Diplomação da aluna Bruna Scheffer Dias, matrícula 1172310 e do aluno Iuri Menin, matrícula 1172336, sob o título **Sistema para Gerenciamento de uma Loja de Confeções**; 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 os candidatos foram entrevistados pela banca examinadora, e a palavra foi aberta ao público. Em seguida, a banca reuniu-se para deliberar considerando o trabalho **APROVADO**. Às 11:30 hrs foi encerrada a sessão.



Prof. Beatriz Terezinha Borsoi, Dr.
Orientadora



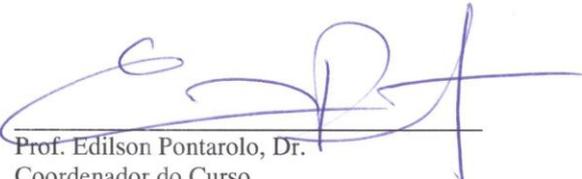
Prof. Andréia Scariot Beulke, Esp.
Convidada



Prof. Vinicius Pegorini, Esp.
Convidado



Prof. Omero Francisco Bertol, M.Sc.
Coordenador do Trabalho de Diplomação



Prof. Edilson Pontarolo, Dr.
Coordenador do Curso

RESUMO

DIAS, Bruna Scheffer; MENIN Iuri. **Sistema para gerenciamento de uma loja de confecções**. 2013. 61 f. Trabalho de conclusão de curso - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná. Pato Branco, 2013.

Os benefícios de sistemas de informação na realização de atividades de compra e venda de produtos e de prestação de serviços são evidentes. Esses sistemas automatizam uma série de atividades realizadas manualmente, reduzem o retrabalho pela não necessidade de inserção das mesmas informações, facilitam a busca de dados armazenados e agilizam a realização de atividades. O controle de estoque (entrada e saída de produtos) e de contas a pagar e a receber caracterizam esse tipo de sistema. Em lojas de confecções o cliente também pode levar peças condicionais. Nesse sistema as peças saem da loja sem que sejam baixadas do estoque, posteriormente, se efetivamente vendidas a transação de venda ocorre. O sistema desenvolvido como resultado deste trabalho se destina ao gerenciamento de uma loja de confecções. Como o mesmo foi desenvolvido para um estabelecimento comercial específico, atende suas particularidades. Essa loja possui um salão de beleza, ou seja, pertencente aos mesmos proprietários. Assim, o sistema também faz o gerenciamento dos serviços realizados no salão. A linguagem Delphi foi utilizada na implementação do sistema.

Palavras-chave: Linguagem Delphi. Gerenciamento de lojas de confecções. Sistema gerencial.

ABSTRACT

DIAS, Bruna Scheffer, MENIN Iuri. **System to manager a cloting store**. 2013. 61 f. Trabalho de conclusão de curso - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná. Pato Branco, 2013.

The benefits of information systems in activities of buying and selling products and services are quite obvious. Inventory control (input and output products) and accounts payable and receivable characterize this type of system. In clothing stores, the customers can also take clothes in conditional. In this system the clothes leave the store without being removed from the stock, after that if the sold occurs, the sale transaction occurs. The system developed as a result of this work is to manager a clothing store. How it was developed for a specific business establishment, meets some peculiarities. This store has a beauty salon that belongs to the same owners. The system also makes the management of services performed in the salon. Delphi Programming Language was used in the development of the system.

Keywords: Linguagem Delphi. Management of chotes store. Manager system.

LISTA DE FIGURAS

Figura 1 - Tela inicial da ferramenta Visual Paradigm	18
Figura 2 - Tela Principal da ferramenta Case Studio.....	19
Figura 3 - Tela inicial de IDE da linguagem Delphi	21
Figura 4 - Tela principal IBExpert	23
Figura 5 - Tela de registro IBExpert.....	23
Figura 6 - Tela principal do FastReport	24
Figura 7 - Diagrama de casos de uso do sistema para lojas de confecções.....	30
Figura 8 - Diagrama de classes.....	33
Figura 9 - Diagrama de entidade relacionamento do banco de dados	37
Figura 10 - Tela de <i>login</i> e senha	40
Figura 11 - Tela inicial do sistema	41
Figura 12 - Menu cadastros	41
Figura 13 - Menu cadastros administrativos	42
Figura 14 - Menu relatórios	42
Figura 15 - Menu gráficos	42
Figura 16 - Tela cadastro de clientes	43
Figura 17 - Tela cadastros de vendas.....	44
Figura 18 - Tela cadastros de empréstimos	44
Figura 19 - Tela cadastros de produtos.....	45
Figura 20 - Tela cadastros de <i>login</i> e senha.....	46
Figura 21 - Tela cadastros de empresas	46
Figura 22 - Tela cadastros de funções	47
Figura 23 - Tela cadastros de funcionários.....	47
Figura 24 - Tela cadastros de fornecedores	48
Figura 25 - Tela cadastros de compras	49
Figura 26 - Tela cadastros de serviços.....	49
Figura 27 - Tela cadastros de contas a pagar.....	50
Figura 28 - Tela cadastros de contas a receber	50
Figura 29 - Relatório de clientes.....	51
Figura 30 - Relatório de fornecedores	51
Figura 31 - Relatório de produtos.....	52
Figura 32 - Relatório de contas a pagar	52
Figura 33 - Relatório de contas a receber	53
Figura 34 - Gráfico de contas a pagar	53
Figura 35 - Relatório de contas a receber	54

LISTA DE QUADROS

Quadro 1 - Requisitos funcionais	28
Quadro 2 - Requisitos não-funcionais	29
Quadro 3 - Caso de uso Cadastrar	31
Quadro 4 - Caso de uso Alterar dados informados.....	31
Quadro 5 - Caso de uso Consultar dados.....	32
Quadro 6 - Caso de uso Excluir dados	32
Quadro 7 - Classe Empresas	33
Quadro 8 - Classe Funções	33
Quadro 9 - Classe Funcionários.....	34
Quadro 10 - Classe Clientes	34
Quadro 11 - Classe Fornecedores	35
Quadro 12 - Classe Condicional/Empréstimos.....	35
Quadro 13 - Classe Compras	35
Quadro 14 - Classe Contas a pagar.....	35
Quadro 15 - Classe Produtos	36
Quadro 16 - Classe Serviços.....	36
Quadro 17 - Classe Vendas	36
Quadro 18 - Classe Contas a receber.....	36
Quadro 19 - Tabela empresa.....	37
Quadro 20 - Tabela funções.....	37
Quadro 21 - Tabela funcionários	38
Quadro 22 - Tabela clientes.....	38
Quadro 23 - Tabela fornecedores	38
Quadro 24 - Tabela empréstimo/condicional	39
Quadro 25 - Tabela compras	39
Quadro 26 - Tabela contas a pagar	39
Quadro 27 - Tabela produtos	39
Quadro 28 - Tabela serviços.....	39
Quadro 29 - Tabela vendas.....	40
Quadro 30 - Tabela contas a receber	40
Quadro 31 - Tabela Usuários.....	40

LISTAGENS DE CÓDIGO

Listagem 1 - Procedure ConectaBanco	54
Listagem 2 - Procedure TfrmLogin.logar	55
Listagem 3 - TfrmMenu.trocardeUsurio1Click(Sender: TObject).....	56
Listagem 4 - TfrmCadClientes.DadosValidos.....	57
Listagem 5 - TfrmPadraoDB.btnNovoClick	58
Listagem 6 - TfrmPadraoDB.btnAlterarClick	58
Listagem 7 - TfrmPadraoDB.btnSalvarClick	58
Listagem 8 - TfrmPadraoDB.btnExcluirClick	59

LISTA DE SIGLAS

ADO	<i>ActiveX Data Objects</i>
ACID	<i>Atomicity, Consistency, Isolation, Durability</i>
API	<i>Application Programming Interface</i>
BDE	<i>Borland Database Engine</i>
CASE	<i>Computer Aided Software Engineering</i>
CLX	<i>Component Library for Cross-Platform</i>
CNPJ	Cadastro Nacional de Pessoas Jurídicas
COM	<i>Component Object Model</i>
CORBA	<i>Common Object Request Broker Architecture</i>
CPF	Cadastro de Pessoas Físicas
DDL	<i>Data Definition Language</i>
DER	Diagrama de Entidades e Relacionamentos
DFD	Diagrama de Fluxo de dados
DLL	<i>Dynamic-Link Library</i>
DM	<i>Data Module</i>
HTML	<i>Hypertext Markup Language</i>
ICMS	<i>Imposto sobre Circulação de Mercadorias e Prestação de Serviços</i>
IDE	<i>Integrated Development Environment</i>
IDL	<i>Interactive Data Language</i>
IP	<i>Internet Protocol</i>
OCX	<i>Abreviatura de OLE custom control</i>
OLE	<i>Object Linkng and Embedding</i>
PNG	<i>Portable Network Graphics</i>
RAD	<i>Rapid Application Devolopment</i>
RG	Registro Geral
SGBD	Sistema Gerenciador de Banco de Dados
SOAP	<i>Simle Object Access Protocol</i>
SQL	<i>Structured Query Language</i>
SQLC	<i>Structured Query Language Connection</i>
SysML	<i>Systems Modeling Language</i>
UDDI	<i>Universal Description, Discovery and Integration</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	11
1.4 ESTRUTURA DO TRABALHO	12
2 LINGUAGEM DELPHI	13
2.1 ALGUNS ASPECTOS DE EVOLUÇÃO DA LINGUAGEM DELPHI	13
2.2 DELPHI PARA AMBIENTE DESKTOP.....	15
2.2 DELPHI PARA AMBIENTE WEB.....	15
2.3 DELPHI PARA LINUX.....	16
3 MATERIAIS E MÉTODO	17
3.1 MATERIAIS	17
3.1.1 Visual Paradigm	17
3.1.2 Case Studio.....	19
3.1.3 IDE Delphi.....	20
3.1.4 Firebird	21
3.1.5 IBExpert	22
3.1.6 FastReport.....	24
3.1.7 Inno Setup.....	25
3.2 MÉTODO	25
4 SISTEMA DESENVOLVIDO.....	27
4.1 APRESENTAÇÃO DO SISTEMA.....	27
4.2 MODELAGEM DO SISTEMA	27
4.3 DESCRIÇÃO DO SISTEMA.....	40
4.4 IMPLEMENTAÇÃO DO SISTEMA.....	54
5 CONCLUSÃO.....	60
REFERÊNCIAS	61

1 INTRODUÇÃO

Neste capítulo são apresentadas as considerações iniciais contendo o contexto no qual se insere a proposta deste trabalho, que é um sistema para controle de uma loja de confecções que possui um salão de beleza vinculado à mesma. Também são apresentados os objetivos e a justificativa do trabalho. Por fim está a organização do texto.

1.1 CONSIDERAÇÕES INICIAIS

Uma loja de confecções que possui um salão de beleza vinculado, além de vender roupas, calçados e acessórios, realiza determinados serviços aos consumidores, como por exemplo, cortes de cabelo, maquiagens e outros relacionados à imagem da pessoa. Esse tipo de loja precisa ter controle de todas as suas movimentações, sejam elas financeiras ou de clientes. Os cadastros de clientes, fornecedores e funcionários ajudam a manter uma boa relação com o cliente focada na produtividade. O controle de contas a pagar e a receber proporciona ao administrador uma melhor organização de seu negócio, ajudando-o a melhorar a lucratividade do negócio.

Para uma empresa que realiza serviços e vendas para determinados clientes, assim como compra de vários fornecedores, o controle feito em papel pode se tornar ineficiente. Assim, verificou-se a possibilidade de desenvolver um sistema que permitisse o armazenamento e o gerenciamento dos dados das atividades realizadas por esse tipo de empresa para que desta maneira ela não deixe de atender aspectos legais e vise lucratividade, além da organização das atividades que realiza. O sistema desenvolvido como resultado deste trabalho é simples, mas visa auxiliar no controle de contas a pagar, contas a receber e empréstimos feitos a seus clientes. Os empréstimos se referem às roupas, acessórios e calçados levados pelo cliente sob a forma condicional, ou seja, ele poderá ou não adquirir os produtos que está levando para experimentar.

Para o desenvolvimento do sistema obtido como resultado deste trabalho foi utilizada a linguagem Delphi, com o banco de dados Firebird e o gerenciador de banco de dados IbExpert, o gerador de relatórios FastReports, o gerador de gráficos TDBChart e o gerador do executável Inno Setup.

Como forma de fundamentar conceitualmente o sistema implementado, no referencial teórico deste trabalho conceitos sobre a linguagem de programação Delphi são apresentados.

1.2 OBJETIVOS

O objetivo geral apresenta o resultado principal a ser obtido com o desenvolvimento deste trabalho e os objetivos específicos o complementam.

1.2.1 Objetivo Geral

Implementar um sistema computacional utilizando a linguagem Delphi para gerenciamento de uma loja de confecções e de um salão de beleza.

1.2.2 Objetivos Específicos

- Implementar uma forma de gerenciamento de contas a pagar, contas a receber, empréstimos, clientes, fornecedores e funcionários visando o gerenciamento facilitado e a organização de tais dados.
- Propor uma forma facilitada de efetuar o controle de serviços realizados que são relacionados às atividades de um salão de beleza.
- Possibilitar o controle de venda condicional a clientes.

1.3 JUSTIFICATIVA

Para Sebesta (2003) o estudo das linguagens de programação é valioso por uma série de razões dentre elas: o aumento da capacidade de usar diferentes construções para escrever programas, possibilita a escolha de linguagens para projetos de maneira mais inteligente e torna mais fácil a aprendizagem de novas linguagens.

Considerando esse contexto, neste trabalho conceitos relacionados à linguagem Delphi são apresentados por ser a linguagem utilizada para implementar o sistema. Conceitos,

classificação e histórico da linguagem de programação Delphi são apresentados porque a ênfase deste trabalho está no aprendizado e uso desta linguagem. O resultado desse aprendizado é o sistema implementado.

O sistema se destina ao controle gerencial de uma loja de confecções que possui um salão de beleza vinculado. Nesse sistema clientes, fornecedores e peças de roupas são cadastrados. E é realizado um controle de contas a pagar e a receber, bem como de peças entregues ao cliente na forma condicional. É um sistema simples, mas que apresenta as funcionalidades necessárias para auxílio na realização das atividades rotineiras de uma loja que trabalha com venda de roupas e da realização de serviços de salão de beleza.

1.4 ESTRUTURA DO TRABALHO

Este texto está organizado em capítulos, dos quais este é o primeiro e apresenta as considerações iniciais, o objetivo e a justificativa.

O Capítulo 2 apresenta o referencial teórico centrado na linguagem de programação Delphi.

O Capítulo 3 apresenta os materiais utilizados na modelagem e no desenvolvimento do sistema e o método. O método contém as principais atividades realizadas para desenvolver o trabalho.

No Capítulo 4 está o resultado obtido da realização deste trabalho que é a modelagem e a implementação do sistema. A modelagem é apresentada por meio de diagramas e explicações textuais e a implementação por telas e partes de código do sistema.

Por fim, está o Capítulo 5 com a conclusão.

2 LINGUAGEM DELPHI

Este capítulo apresenta conceitos, aspectos relacionados à evolução e sobre as versões da linguagem Delphi.

2.1 ALGUNS ASPECTOS DE EVOLUÇÃO DA LINGUAGEM DELPHI

A linguagem Delphi teve o seu início, primeira versão, em 1994, com o ambiente de programação Delphi 1.0. Esse ambiente era baseado na linguagem Object Pascal.

A linguagem Delphi na sua versão 2 possuía o objetivo de gerenciar múltiplos registros, o acesso a bancos de dados foi aperfeiçoado, o suporte à automação OLE (*Object Linking and Embedding*) e o tipo de dado variante, suporte e integração com o Windows 95, o tipo de dado *string* e a herança visual de formulário.

Na versão 3, denominada Delphi 3, incluiu a tecnologia de visão de código, o suporte à depuração DLL (*Dynamic-Link Library*), os modelos de componentes, o TeeChart, o Decision Cube, a tecnologia WebBroker, os pacotes de componentes, o ActiveForm e integração utilizando COM (*Component Object Model*) por meio de interface.

O Delphi 4 proporcionou o editor AppBrowser, novos recursos do Windows 98, melhor suporte a OLE e COM, componentes de banco de dados estendidos e muitas adições feitas nas classes básicas da VCL (*Visual Component Library*), incluindo suporte para acoplamento, restrição e controle de ancoragem.

O Delphi 5 incluiu mais aperfeiçoamentos a IDE (*Integrated Development Environment*), suporte estendido a banco de dados (com os conjuntos de dados específicos ADO (*ActiveX Data Objects*) e InterBase), uma versão aperfeiçoada do Midas com suporte para a Internet, a ferramenta de controle de versão TeamSource, capacidades de conversão, o conceito dos quadros e novos componentes.

O Delphi 6 adicionou suporte a recursos para o desenvolvimento independente de plataforma com a CLX (*Component Library for Cross-Platform*), a biblioteca de componentes independentes de plataforma, uma biblioteca de tempo de execução estendida, o mecanismo de banco de dados dbExpress, serviços da *web* e suporte para XML (*Extensible Markup Language*), uma estruturada de desenvolvimento para a *web*, mais aperfeiçoamento na IDE e muitos componentes e classes.

O Delphi 7 melhorou algumas dessas novas tecnologias com aperfeiçoamentos e correções (o suporte a SOAP (*Simple Object Access Protocol*) e DataSnap, por exemplo) e oferece suporte a tecnologias mais recentes (como os temas do Windows XP e UDDI (*Universal Description, Discovery and Integration*)). Uma das suas principais características é a facilidade de disponibilidade de um conjunto de ferramentas de terceiros: o mecanismo de produção de relatórios Rave, a tecnologia IntraWeb de desenvolvimento de aplicativos para a *web* e o ambiente de projeto ModelMaker.

Segundo Cantú (2003), o Delphi 7 é uma das ferramentas que possui mais recursos para criar aplicações Windows. Além de ser amplamente utilizado para aplicações Cliente/Servidor, ele também é muito utilizado para aplicações genéricas.

Uma das principais características da linguagem Delphi é o uso de componentes prontos. Os formulários são compostos a partir de elementos (componentes) predefinidos que possuem propriedades configuráveis e permitem a codificação de eventos (ações), realizando as funcionalidades pretendidas para o aplicativo. A linguagem Delphi faz uma boa combinação entre programação orientada a objetos e programação visual, seja para o ambiente Windows ou Linux.

A versão 2009 do Delphi é responsabilidade da empresa Embarcadero, que adquiriu da Borland a divisão CodeGear que é a fornecedora de ferramentas de desenvolvimento. Apesar de o novo recurso principal ser o suporte de Unicode na VCL, vários outros recursos destacam-se nessa versão, dentre eles estão (ARANTES, 2012):

- Novos controles na VCL, com destaque ao Ribbons estilo Office 2007.
- Melhorias na IDE, na ImageList, incluindo suporte a PNG (*Portable Network Graphics*).
- Novos recursos de linguagem, com destaque a Generics e Anonymous Methods.
- VCL para *web* atualizada com suporte a Ajax e Silverlight.
- DataSnap 2009, que simplifica o desenvolvimento de aplicações três camadas sem uso de COM.

O Embarcadero RAD Studio 2010, a versão 2010 do ambiente de desenvolvimento da linguagem Delphi, o *Rapid Application Development (RAD)* incorpora novas versões do Delphi, C++Builder e do Delphi Prism. O RAD Studio 2010 possibilita o desenvolvimento rápido de interfaces baseadas em *touch* e *gesture* para aplicações existentes ou novas, além de mais de 120 funções novas e aperfeiçoadas que prometem aumentar a produtividade do desenvolvedor de aplicações Windows e .Net (LIMA, 2012).

Em 2010 foi lançado o Delphi XE e em 2011 a versão XE2 que suporta o desenvolvimento de aplicativos para dispositivos móveis. O Embarcadero Delphi XE3 visa à construção de aplicações para Windows da versão XP a versão 8 e para Mac OS X Mountain Lion, incluindo Retina display, Slates e Surface Pro, com uma única base de código (EMBARCADERO, 2012).

2.2 DELPHI PARA AMBIENTE *DESKTOP*

A linguagem Delphi é muito utilizada no desenvolvimento de aplicações *desktop*, aplicações multicamadas e Cliente/Servidor, compatíveis com diversos bancos de dados. Delphi pode ser utilizado para vários tipos de desenvolvimento de projeto, abrangendo serviços a aplicações *web* (LAMIN, 2012).

Existem três versões disponíveis do Delphi (MARTINS, 2012): Desktop, Developer e Client/Server. A versão *desktop* inclui: o compilador Delphi; o Delphi IDE que é um ambiente integrado para desenvolvimento, teste e depuração de aplicações; o *Borland Database Engine* (BDE) que inclui DBase, Paradox e suporte a ODBC; a Visual Component Library com mais de 90 componentes; o Database Desktop que permite criar, visualizar, classificar, modificar e consultar tabelas de dados de diversos formatos, incluindo Paradox, DBase e SQL (*Structured Query Language*); e o Database Explorer que é um *browser* hierárquico de banco de dados, com capacidade de visualizar esquemas de banco de dados, criar, visualizar e editar dados em tabelas, criar e manter *alias*, executar comandos SQL, criar e manter dicionário de dados.

2.2 DELPHI PARA AMBIENTE WEB

Delphi possui diversos componentes de fonte aberta para a Internet chamada Internet Direct (Indy). Anteriormente os componentes Indy eram chamados de WinShoes.

Os componentes produtores de HTML (*Hypertext Markup Language*) do Delphi (disponíveis na página Internet da paleta de componentes) podem ser usados para gerar os arquivos HTML e, particularmente, para transformar uma tabela de banco de dados em uma tabela HTML (FACUNTE, 2003).

Em 2008, a CodeGear, unidade Borland para ferramentas de desenvolvimento, lançou o Delphi para PHP 2.0. Essa ferramenta é destinada à construção de aplicações *web*

interativas com o uso de componentes visuais de arrastar-e-soltar e um *framework* de componente PHP (LANUSSE, 2008).

A versão 2.0 de Delphi para PHP tem como foco tornar o desenvolvimento PHP para *web* ainda mais fácil em várias áreas (LANUSSE, 2008):

- a) Templates HTML com PHP dinâmico incorporado, permitindo trabalhar visualmente no desenvolvimento para *web* 2.0 com PHP;
- b) Biblioteca de Componentes Visuais (VCL) para PHP ampliada, com suporte aos pacotes e bibliotecas mais populares, incluindo o Zend Framework;
- c) Aperfeiçoamentos de produtividade e de desempenho, incluindo depuração de erros e formatação de código fonte;
- d) Novas funções de depuração para PHP.

2.3 DELPHI PARA LINUX

O Kylix é uma das primeiras ferramentas de desenvolvimento RAD para o sistema operacional Linux. Com ele, aplicações podem ser portadas do ambiente Windows para o ambiente Linux (SONNINO, 2001).

Da mesma maneira que o Delphi acessa a API (*Application Programming Interface*) do Windows, o Kylix pode acessar a API do Linux. As funções da biblioteca C padrão do Linux estão disponíveis quando incluída a *unitLibc* na cláusula *Uses*. Dessa maneira, é possível utilizar todos os recursos e funções do sistema operacional, permitindo executar outros programas e obter dados do usuário, dentre outros.

3 MATERIAIS E MÉTODO

Este capítulo apresenta os materiais e o método utilizados. Os materiais se referem às tecnologias e ferramentas utilizadas para modelar e implementar o sistema. O método reporta a sequência das principais atividades realizadas para desenvolver este trabalho.

3.1 MATERIAIS

Foram utilizadas as seguintes tecnologias e ferramentas para a modelagem e a implementação do sistema:

- a) Visual Paradigm - para a modelagem do sistema.
- b) Case Studio – para a modelagem do diagrama de entidades e relacionamentos do banco de dados.
- b) Delphi 7.0 - como linguagem de programação e IDE de desenvolvimento.
- c) Firebird 2.0 - para banco de dados.
- d) IbExpert 2012.02.21 - para gerenciador do banco de dados.
- e) FastReport 4.0 - para desenvolvimento dos relatórios.
- f) Inno Setup Compiles 5.5.1 – para desenvolvimento do executável.

3.1.1 Visual Paradigm

Visual Paradigm for UML é uma ferramenta de modelagem para todos os tipos de diagramas UML (*Unified Modeling Language*) (VISUAL PARADIGM, 2012). Essa ferramenta fornece suporte para gerenciamento de casos de uso, diagrama de requisitos SysML (*Systems Modeling Language*) e projeto de banco de dados com diagrama de entidades e relacionamentos. A Figura 1 apresenta um *print screen* da interface principal dessa ferramenta.

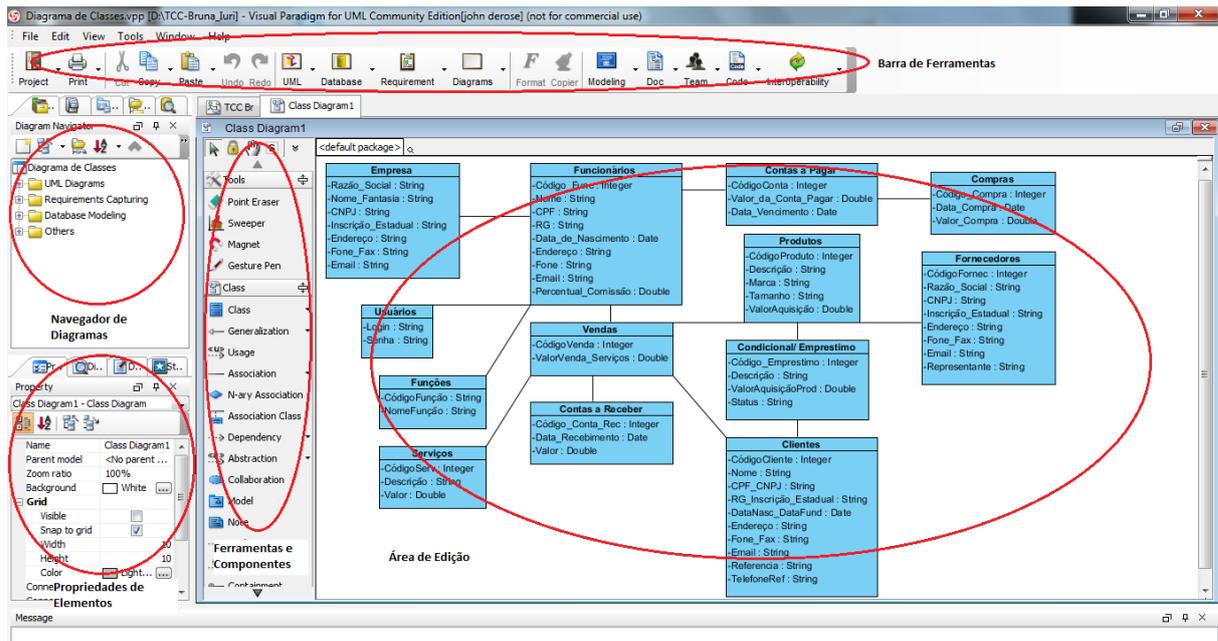


Figura 1 - Tela inicial da ferramenta Visual Paradigm

As partes destacadas na Figura 1 apresentam os principais elementos da tela principal do Visual Paradigm e são:

a) Navegador de diagramas - nesta área da interface são apresentados os diagramas armazenados. A disponibilização dos arquivos em pastas por tipo de diagrama facilita a localização dos mesmos e organização do projeto.

b) Propriedades – apresenta as propriedades do elemento selecionado do diagrama em edição.

c) Área de edição - nesta área são criados graficamente os diagramas por meio dos elementos dispostos na lateral esquerda dessa área.

d) Ferramentas e componentes – área na qual ficam disponíveis os componentes utilizados para compor o diagrama e ferramentas relacionadas.

e) Barra de ferramentas – com os atalhos para as principais funcionalidades do Visual Paradigm.

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

3.1.2 Case Studio

Case Studio 2 (CASESTUDIO, 2012) é uma ferramenta CASE (*Computer Aided Software Engineering*) de modelagem de banco de dados. CASE é uma técnica que visa facilitar o desenvolvimento de sistemas de *software* por meio do uso de ferramentas. Ferramentas CASE podem auxiliar na realização de todas as atividades relacionadas ao processo de *software*, como análise, projeto, codificação, teste e manutenção. A Figura 2 apresenta a tela principal da ferramenta Case Studio.

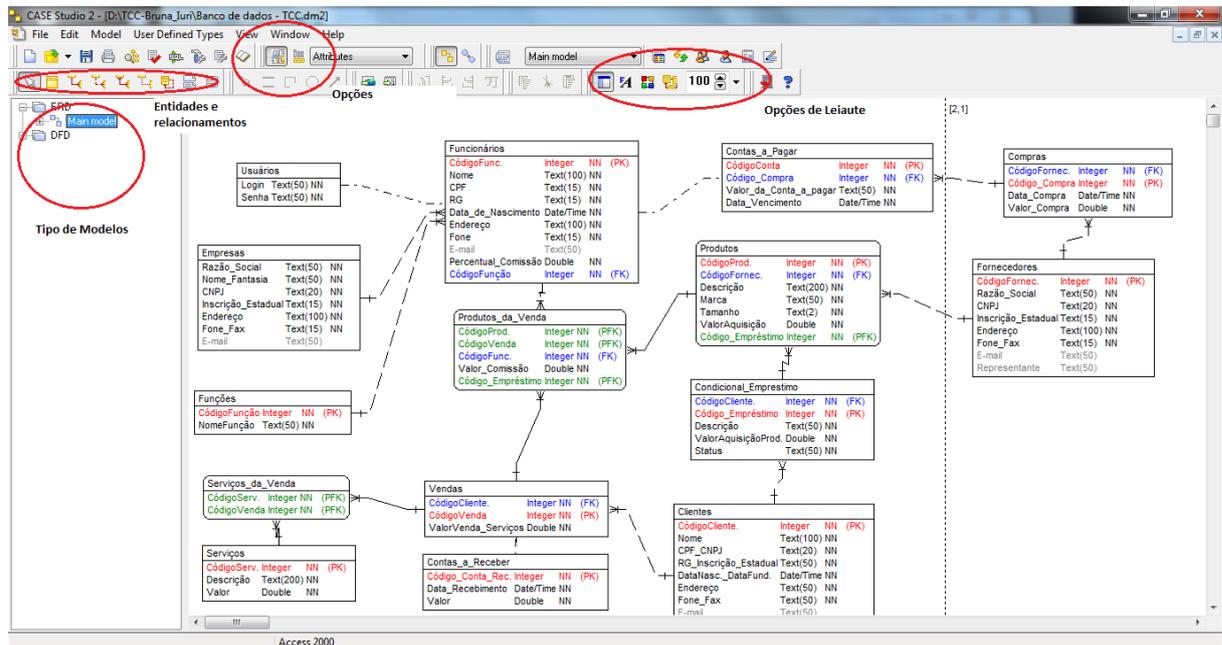


Figura 2 - Tela Principal da ferramenta Case Studio

As áreas circuladas da Figura 2 indicam:

a) Tipo do Modelo – DER (Diagrama de Entidades e Relacionamentos) ou DFD (Diagrama de Fluxo de dados). Os dois diagramas possíveis de serem elaborados com Case Studio.

b) Entidades e Relacionamentos – criação das entidades e de seus relacionamentos para a representação gráfica do banco de dados.

c) Opções Entidades – opções de visualização das entidades (somente entidades, chaves primárias, chaves e atributos).

d) Opções do *Layout* – opções do leiaute do modelo (fontes, esquema de cores, leiaute automático e *zoom*).

Dentre os principais recursos do Case Studio estão (FROLIK, 2006): diagramas de entidade e relacionamento; diagramas de fluxo de dados; geração de *scripts* SQL; engenharia reversa; e geração de documentação detalhada.

Na definição de diagramas de entidade e relacionamento o programa considera opções de bancos de dados individuais, tais como integridade referencial, restrições e domínios, dentre outros. Além da definição de tabelas com atributos e relacionamentos, o Case Studio permite definir critérios, como índices e chaves primárias.

3.1.3 IDE Delphi

A linguagem Delphi, que está vinculada a um ambiente de desenvolvimento, se baseia em uma extensão orientada a objetos da linguagem de programação Pascal, também conhecida como *Object Pascal* (CANTÚ, 2003). Esse ambiente de desenvolvimento é composto de várias ferramentas agregadas.

Como ambiente de desenvolvimento, Delphi é um compilador e também uma IDE para o desenvolvimento de *software*. A linguagem utilizada é *Object Pascal* (Pascal com extensões orientadas a objetos). Dentre as características da linguagem Delphi (CANTÚ, 2000; CANTÚ, 2003) destacam-se:

- a) IDE, o ambiente integrado de desenvolvimento.
- b) RAD que agiliza o desenvolvimento de aplicações.
- c) Construtor visual de interface com o usuário, baseado em formulários e componentes. O uso de componentes facilita a reutilização e a manutenção de código.
- d) Compilador de código nativo.
- e) *Drag-and-drop design*: o código é gerado automaticamente durante a montagem do formulário.
- f) *Tow-way tools*: alternar entre a visualização de um formulário e seu código (*unit*).
- g) Biblioteca de componentes visuais: A VCL consiste de objetos reutilizáveis incluindo objetos padrão de interface com o usuário, gerenciamento de dados, gráficos e multimídia, gerenciamento de arquivos. E é possível visualizar a hierarquia dos objetos na VCL.
- h) Arquitetura aberta. Possibilidade de adicionar componentes e ferramentas de terceiros.
- i) Linguagem de programação orientada a objeto (*Object Pascal* - 4ª geração).
- j) Depurador gráfico. O *debugger* auxilia a encontrar e a eliminar erros no código.
- k) Gerenciador de projetos: oferece uma visualização de todos os formulários e as *units* de um determinado projeto e oferece um mecanismo para gerenciar projetos.

l) Geradores de relatórios: o Quick Report e o Rave Reports são ferramentas para geração de relatórios.

m) BDE e SGBD (*Sistema Gerenciador de Banco de Dados*), Database Explorer.

n) Help *on-line*: sensível ao texto.

A Figura 3 apresenta a tela inicial de IDE da linguagem Delphi.

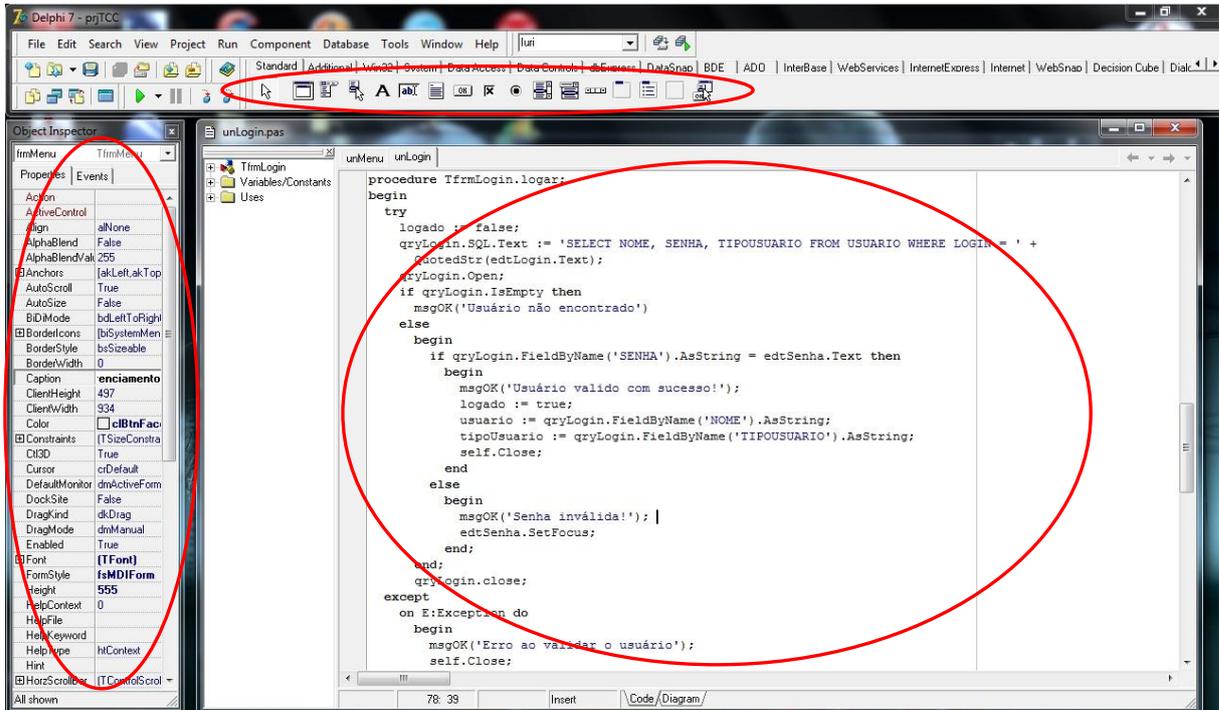


Figura 3 - Tela inicial de IDE da linguagem Delphi

As partes destacadas na Figura 3 são:

a) Paleta de componentes – a paleta de componentes é uma barra de ferramentas com componentes a serem utilizados no desenvolvimento de aplicações. A paleta é constituída de várias guias. Cada guia possui os componentes agrupados de acordo com sua finalidade.

b) *Form designer* - o *form* é uma janela onde a interface gráfica da aplicação é desenvolvida. No *form* são colocados os componentes que farão parte da interface com o usuário.

c) *Object inspector* - define propriedades e eventos para os componentes. O *object inspector* muda de acordo com o componente escolhido. O *object inspector* oferece acesso a todas as propriedades publicadas (*published*) do componente selecionado.

3.1.4 Firebird

O Firebird é um Sistema Gerenciador de Banco de Dados de código fonte aberto e

seus principais recursos são (CANTÚ, 2010):

- Suporte a *stored procedures* e *triggers*;
- Transações compatíveis com ACID (*Atomicity, Consistency, Isolation, Durability*);
- Integridade referencial;
- Linguagem nativa para *stored procedures* e *triggers* (PSQL);
- Suporte para funções externas;
- Diversas formas de acesso ao banco de dados;
- Suporte nativo para diversos sistemas operacionais, incluindo o Windows, Linux, Solaris, MacOS;
- Backups incrementais;
- Tabelas temporárias.

3.1.5 IBExpert

IBExpert é um programa para administração de bancos de dados InterBase e Firebird. Dentre os seus recursos estão (IBEXPERT, 2013):

- Criação de tabelas, chaves e índices;
- Modelagem de dados (*database designer*);
- Editores para todos os objetos do banco de dados (tabelas, *triggers*, *procedures*, *domains*, etc.) com depurador de *triggers* e *stored procedures*, construtor visual de consultas SQL e recursos avançados de edição, tais como: *code insight*, *code completion* e *keyboard templates*.

O IBExpert permite a criação de tabelas sem a necessidade da utilização de comandos DDL (*Data Definition Language*). Todos os atributos de uma tabela, tais como: campos, chave primária (*primary key*), chave estrangeira (*foreign key*), índices, *triggers*, etc., podem ser criados e alterados visualmente, o IBExpert gera automaticamente o *script* para criar o banco de dados.

A Figura 4 apresenta a tela principal do IBExpert.

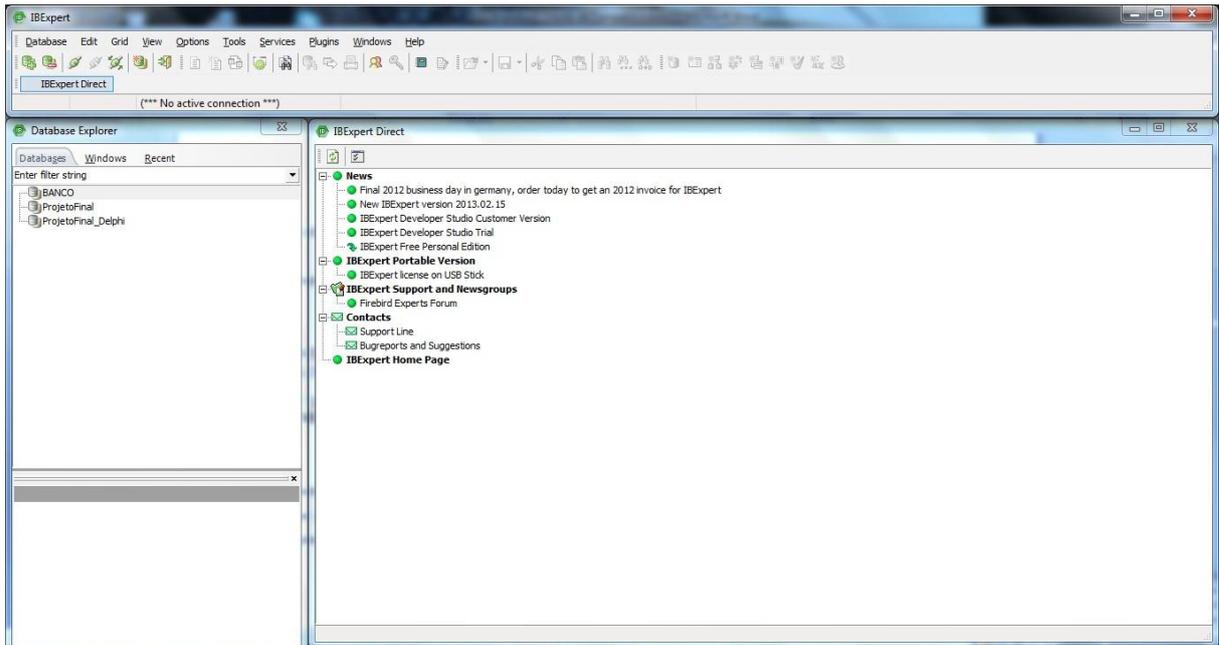


Figura 4 - Tela principal IBExpert

A criação de um banco de dados é realizada através da opção Create DataBase no Menu DataBase. A Figura 5 apresenta a tela de criação de um novo banco de dados.

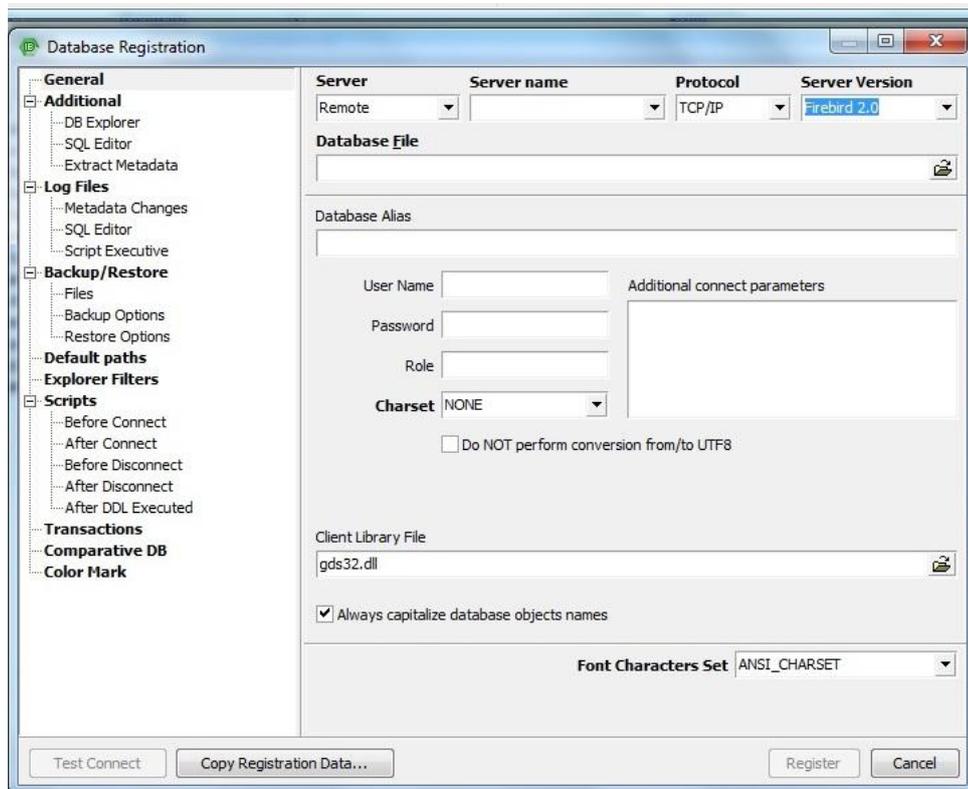


Figura 5 - Tela de registro IBExpert

Os dados a serem cadastrados, de acordo com a Figura 5, na criação do banco de dados são:

- *Server*: indicar o servidor do banco de dados que pode ser local ou remoto.

- *Server name*: para a opção remoto em *Server* é necessário informar o nome ou IP (*Internet Protocol*) do computador servidor do banco de dados.
- *Protocol*: é o protocolo usado para fazer a comunicação com o servidor.
- *Server Version*: para indicar a versão do servidor de banco de dados.
- *DataBase File*: o arquivo do banco de dados.
- *DataBase Alias*: para indicar um nome para a identificação do banco de dados.
- *Username*: SYSDBA que é o administrador padrão do banco de dados Firebird.
- *Password*: preencher nesse campo a senha masterkey.
- *Role*, *Charset* e *Client Library File*: devem ser mantidos os valores padrão.

3.1.6 FastReport

O FastReport é um componente que facilita a geração de relatórios por meio de um editor visual que oferece uma janela de pré-visualização (FAST REPORT, 2013).

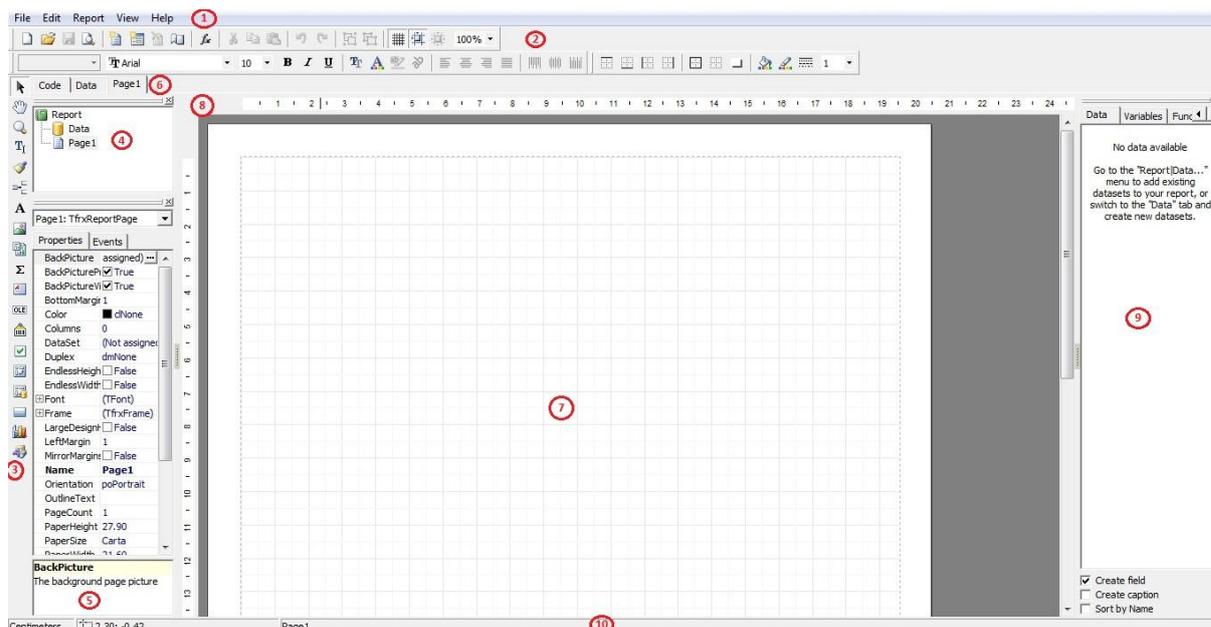


Figura 6 - Tela principal do FastReport

As partes indicadas na Figura 6 são:

- 1) Barra de Menus;
- 2) Barra de Ferramentas;
- 3) Barra de Objetos;
- 4) Árvore de Componentes;
- 5) Janela de Propriedades do Objeto (Object Inspector);
- 6) Páginas do Relatório;

- 7) Espaço de Design do Relatório;
- 8) Réguas;
- 9) Árvore de Dados;
- 10) Barra de Status.

3.1.7 Inno Setup

O Inno Setup é um gerador de instaladores para programas Windows. Esse gerador é gratuito e de código fonte aberto. O Inno Setup permite criar chaves no registro e em arquivos .ini, além de copiar arquivos para alguma pasta e de criar atalhos e desinstaladores de programas (PICAIO, 2013). Várias propriedades podem ser configuradas, dentre elas a de limitar a instalação somente a administradores, registrar DLL/OCX e instalar fontes.

A geração dos instaladores com o Inno Setup ocorre por meio de *script*. O compilador do Inno gera um executável a partir do *script*. Esse executável contém os arquivos, as configurações e o módulo do instalador.

3.2 MÉTODO

As etapas para a modelagem e a implementação do sistema resultado deste trabalho seguiram o modelo sequencial linear proposto por Pressman (2005). O uso desse modelo é justificado porque o sistema é simples e os requisitos do mesmo foram todos definidos no início do processo.

As etapas definidas foram:

a) Levantamento dos requisitos

Os requisitos foram definidos tendo como base a necessidade de um controle de uma loja de confecções que possui um salão de beleza vinculado à mesma. Essa loja de confecções possui funcionários que realizam vendas e prestam serviços aos clientes. As vendas são relacionadas aos produtos como vestuário e cosméticos. Esses produtos são comprados pelo administrador da loja por meio de fornecedores. Os serviços prestados pelos colaboradores da empresa estão vinculados ao salão de beleza e são cortes de cabelo, tinturas, massagens e manicure, dentre outros. Assim, como levantamento de requisitos foi realizado o

levantamento das informações que seriam necessárias manter no sistema para a administração das contas a pagar e a receber e dos empréstimos de mercadorias realizados na loja.

Observando os processos realizados na loja e no salão e em conversas com a proprietária do estabelecimento foi possível definir os requisitos para o sistema.

b) Análise e projeto

Os requisitos levantados foram modelados por meio de diagrama de casos de uso, de classes e de entidades e relacionamentos do banco de dados. Os casos de uso, as classes e as tabelas foram descritos e documentados.

c) Implementação

A implementação do sistema foi realizada com o uso da linguagem Delphi 7. Os relatórios foram elaborados utilizando o FastReport e o executável do programa gerado com o uso da ferramenta Inno Setup.

d) Testes

Os testes foram realizados pelos autores deste trabalho visando identificar erros de codificação e para verificar se as funcionalidades do sistema foram devidamente implementadas.

4 SISTEMA DESENVOLVIDO

Este capítulo apresenta o sistema para gerenciamento de uma loja de confecções e de um salão de beleza modelado e implementado como resultado do desenvolvimento deste trabalho.

4.1 APRESENTAÇÃO DO SISTEMA

Para o gerenciamento de uma loja de confecções e de prestação de serviços de salão de beleza é necessário cadastrar a empresa, assim como dos seus colaboradores e suas respectivas funções. É necessário também cadastrar os produtos que serão vendidos aos clientes. Os produtos comercializados são distribuídos para a empresa por meio de fornecedores os quais devem ser cadastrados no sistema. O sistema desenvolvido possibilitará que clientes levem produtos de forma condicional. E, assim, é necessário fazer o controle desses produtos. Esse sistema também fornece apoio ao gerenciamento do negócio por meio de um controle simples de contas a pagar e a receber.

4.2 MODELAGEM DO SISTEMA

O Quadro 1 apresenta os requisitos funcionais identificados para o sistema.

Identificação	Nome	Descrição
RF001	Cadastrar empresa	O usuário do sistema deve cadastrar a empresa informando dados como razão social, nome fantasia, CNPJ (Cadastro Nacional de Pessoas Jurídicas), Inscrição Estadual, endereço, telefone/fax e <i>email</i> .
RF002	Cadastrar funções	O usuário do sistema deve cadastrar a função que os colaboradores exercem informado o nome da mesma.
RF003	Cadastrar funcionários	O usuário do sistema deve cadastrar os funcionários da empresa informando dados como o nome, CPF (Cadastro de Pessoa Físicas), RG (Registro Geral), data de nascimento, endereço, fone, percentual de comissão que recebe por venda e a função que tal exercerá.
RF004	Cadastrar clientes	O responsável por tal cadastramento terá que informar, caso o cliente seja pessoa física: nome, CPF,

		RG, data de nascimento, endereço, fone/fax, <i>email</i> e referência. Caso o cliente seja empresa jurídica o responsável deve informar: Razão social, CNPJ, Inscrição Estadual, data de fundação, endereço, fone/fax, <i>email</i> , referência e telefone da mesma.
RF005	Cadastrar fornecedores	O usuário responsável terá que informar a razão social, CNPJ, Inscrição Estadual, endereço, fone/fax, <i>email</i> e representante.
RF006	Cadastrar condicional/ empréstimo	O usuário do sistema deve informar o código do cliente que está emprestando, a quantidade de produtos que estão sendo emprestados, assim como a descrição dos produtos e os seus respectivos valores de aquisição.
RF007	Cadastrar compras	O usuário deve cadastrar as compras efetuadas pela loja. Para isso ele deve informar o código do fornecedor que fez a venda para a loja e o valor da compra.
RF008	Cadastrar contas a pagar	Para o cadastramento de contas a pagar o usuário deve informar o código da compra que está sendo paga, a data de vencimento da mesma e o valor a ser pago.
RF009	Cadastrar produtos	O usuário deve cadastrar o código do fornecedor que irá vender o produto a loja, também deve ser informada a descrição do produto, sua marca, tamanho e valor de aquisição do mesmo.
RF010	Cadastrar serviço	Para cadastrar um serviço deve ser informada a descrição do serviço e o valor do mesmo.
RF011	Cadastrar vendas	Para registrar uma venda o usuário deve informar o código do cliente para quem está sendo realizada a venda e também deve ser informado o valor da venda.
RF012	Cadastrar contas a receber	O usuário deve cadastrar as contas a receber informando a data de vencimento da mesma e o seu valor.
RF013	Cadastrar usuários	O usuário administrador deve cadastrar o <i>login</i> e senha dos colaboradores que terão acesso ao sistema.

Quadro 1 - Requisitos funcionais

No Quadro 2 estão os requisitos não funcionais identificados para o sistema. Os requisitos não funcionais explicitam regras de negócio, restrições ao sistema de acesso, por exemplo, requisitos de qualidade, desempenho, segurança e outros.

Identificação	Nome	Descrição
RNF001	Cadastrar empresas	Somente o administrador do sistema poderá realizar esse cadastro, sendo necessário o preenchimento dos campos razão social, nome fantasia, CNPJ, inscrição estadual, endereço e fone/fax.
RNF002	Cadastrar funções	Esse cadastro pode ser realizado somente pelo administrador do sistema sendo que o preenchimento do campo nome da função torna-se obrigatório.

RNF003	Cadastrar funcionários	Cadastro que só pode ser feito pelo administrador do sistema sendo de exclusiva obrigatoriedade o preenchimento dos campos nome, CPF, RG, data de nascimento, endereço, fone, percentual de comissão e código da função exercida.
RNF004	Cadastrar fornecedores	Somente o administrador do sistema pode realizar esse cadastro informando assim obrigatoriamente os dados: razão social, CNPJ, Inscrição Estadual, endereço e fone/fax.
RNF005	Cadastrar condicional/ empréstimo	A cada condicional feito o cliente não poderá levar mais que dez produtos em condicional/empréstimo.
RNF006	Cadastrar compras	Cadastro que deve ser realizado somente pelo administrador do sistema sendo de obrigatoriedade o preenchimento dos dados data da compra, valor da compra não se esquecendo de informar o código do fornecedor que realizou tal venda para a empresa.
RNF007	Cadastrar contas a pagar	Cadastro realizado somente pelo administrador do sistema sendo que é obrigatório o preenchimento dos campos data de vencimento e valor, não se esquecendo de vinculá-lo a qual compra pertence.
RNF008	Cadastrar serviços	Registro feito pelo administrador do sistema tendo a necessidade do preenchimento dos campos descrição e valor.
RNF009	Cadastrar contas a receber	Cadastro efetuado pelo administrador do sistema sendo necessário o preenchimento dos campos data de recebimento e valor.
RNF010	Cadastrar usuários	Cadastro efetuado pelo administrador do sistema sendo necessário o preenchimento das informações <i>login</i> e senha.

Quadro 2 - Requisitos não-funcionais

A Figura 4 apresenta o diagrama de casos de uso relacionados aos requisitos apresentados no Quadro 1.

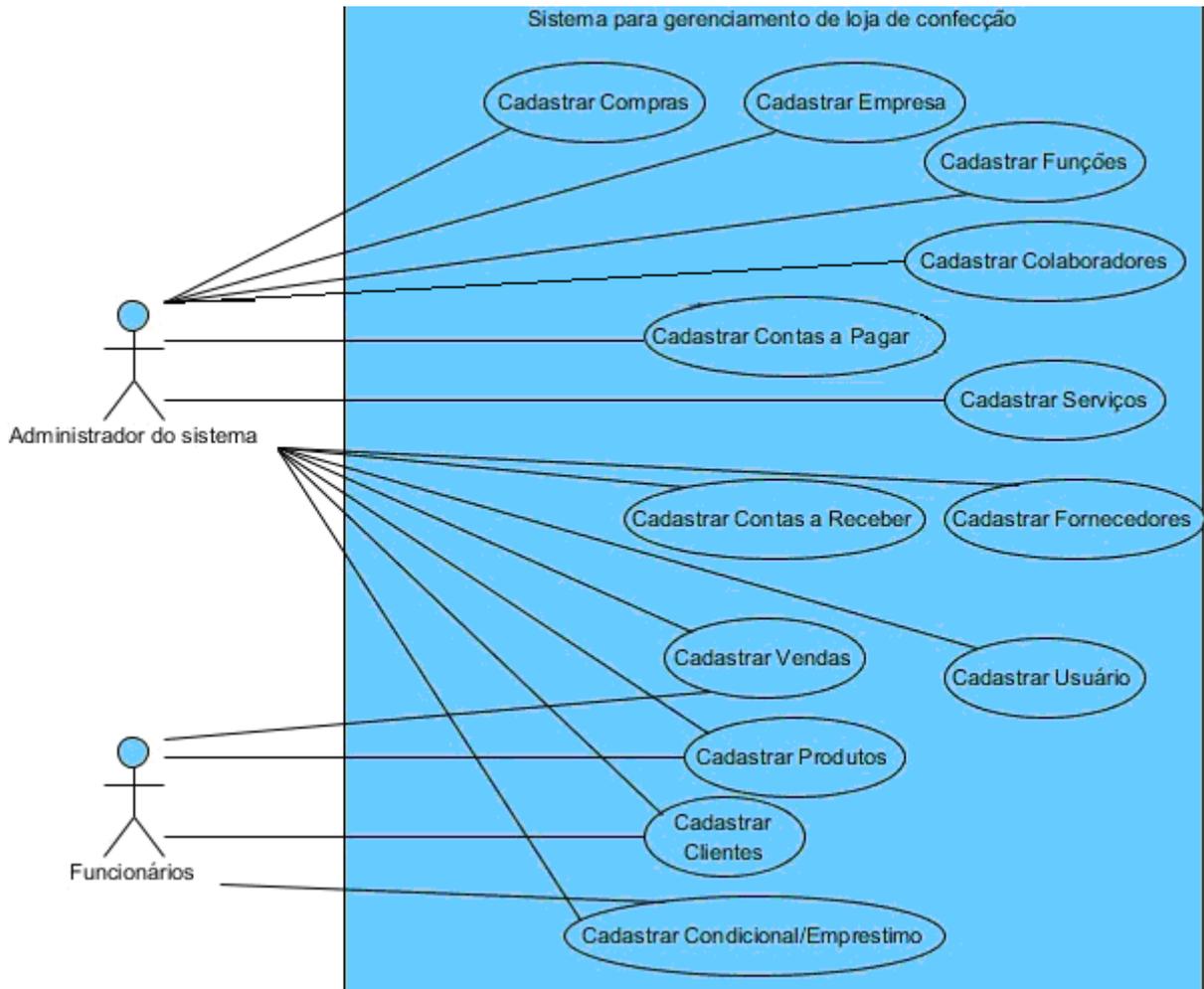


Figura 7 - Diagrama de casos de uso do sistema para lojas de confecções

A seguir, nos Quadros 3 a 6, estão documentados os casos de uso que constam na Figura 7. Cada caso de uso pode implementar um ou mais requisitos e estar vinculado a um ou mais requisitos não funcionais. O Quadro 3 apresenta a descrição da operação de inclusão dos casos de uso cadastrar.

Identificador do requisito: Cadastrar.

Descrição: Este caso de uso permite incluir dados cadastrais.

Evento Iniciador: Tela de cadastros disponíveis.

Atores: Administrador do sistema, Funcionário.

Pré-condição: Não há.

Seqüência de Eventos:

- 1 – Ator Administrador ou Funcionário informa dados para cadastro.
- 2 – Sistema valida os dados e os inclui no banco de dados.
- 3 – Sistema retorna mensagem de cadastramento efetuado com sucesso.

Pós-Condição: Os dados informados deverão ser válidos.

Extensões: Se dados forem informados incorretamente o sistema deverá solicitar que eles sejam informados novamente.

Nome do fluxo alternativo (extensão)	Descrição
Informações inválidas	Se a informação prescrita for inválida o

		usuário deverá informá-la novamente.
Inclusões: Validar dados.		
Requisitos não funcionais:		
Identificador	Nome	Descrição
RNF1.1	Cadastrar	As informações descritas que necessitam de validação só poderão ser cadastradas se forem validadas com sucesso.

Quadro 3 - Caso de uso Cadastrar

No Quadro 4 está a descrição da operação alterar dos casos de uso identificados como “cadastrar” no diagrama da Figura 3.

Identificador do requisito: Alterar dados informados.		
Descrição: Este caso de uso permite alterar dados de cadastros existentes no banco de dados.		
Evento Iniciador: Tela de alterações disponíveis.		
Atores: Administrador do sistema, Funcionário.		
Pré-condição: Não há.		
Seqüência de Eventos:		
1 – Ator solicita apresentação dos dados do cadastro do qual pretende alterar dados.		
2 – Ator realizar as alterações desejadas nos dados.		
3 – Sistema valida se é possível realizar as alterações solicitadas.		
4 – Sistema retorna mensagem de alterações efetuadas com sucesso.		
Pós-Condição: As alterações requisitadas pelo usuário deverão ser realizadas.		
Extensões: Se alterações requisitadas não forem permitidas o sistema deverá informar que é impossível realizá-las.		
Nome do fluxo alternativo (extensão)	Descrição	
Alterações inválidas	Se a alteração requisitada pelo usuário for inválida o sistema deve informar que não será possível fazer determinadas mudanças.	
Inclusões: Validar alterações.		
Requisitos não funcionais:		
Identificador	Nome	Descrição
RNF1.2	Alterar	As alterações requisitadas só poderão sofrer mudanças se estas forem válidas para o manuseio do usuário.

Quadro 4 - Caso de uso Alterar dados informados

No Quadro 5 é apresentada a descrição da operação consultar dos casos de uso de cadastro.

Identificador do requisito: Consultar dados.
Descrição: Este caso de uso permite consultar dados de cadastros do sistema.
Evento Iniciador: Tela de consultas disponíveis.
Atores: Administrador do sistema, Funcionário.
Pré-condição: Não há.
Seqüência de Eventos:

<p>1 – Escolher cadastro e registro para que seus dados sejam apresentados. 2 – Sistema valida se estes dados estão disponíveis para consulta. 3 – Sistema apresenta os dados solicitados. Pós-Condição: os dados obtidos da consulta apresentados. Extensões: se não for possível realizar a consulta de tais dados o sistema deverá informar que certa consulta esta indisponível.</p>		
Nome do fluxo alternativo (extensão)		Descrição
Consultas indisponíveis		Se a consultar estiver indisponível o sistema deverá informar ao usuário tal condição.
<p>Inclusões: Validar consultas Requisitos não funcionais:</p>		
Identificador	Nome	Descrição
RNF1.3	Consulta	As informações descritas só poderão ser consultadas se estas estiverem disponíveis para tal ato.

Quadro 5 - Caso de uso Consultar dados

O Quadro 6 apresenta a descrição do caso de uso excluir dados.

<p>Identificador do requisito: Excluir dados. Descrição: Este caso de uso permite excluir dados de tabelas de banco de dados. Evento Iniciador: Tela de exclusões disponíveis. Atores: Administrador do sistema, Funcionário. Pré-condição: Não há. Seqüência de Eventos: 1 – O ator informa o cadastro e respectivo registro do qual quer excluir dados. 2 – Sistema valida se os dados podem ser excluídos. 3 – Sistema retorna mensagem de exclusão efetuada com sucesso. Pós-Condição: As exclusões requisitadas deverão estar disponíveis para o usuário. Extensões: Se não é possível realizar a exclusão solicitada pelo usuário, o sistema deverá informar a impossibilidade de realizá-la.</p>		
Nome do fluxo alternativo (extensão)		Descrição
Exclusões inválidas		Se a exclusão for inválida o sistema deverá informar ao usuário.
<p>Inclusões: Validar dados. Requisitos não funcionais:</p>		
Identificador	Nome	Descrição
RNF 1.4	Excluir	Só poderá ocorrer determinada exclusão se o usuário estiver apto a realizar tal ato.

Quadro 6 - Caso de uso Excluir dados

A Figura 8 apresenta o diagrama de classes do sistema.

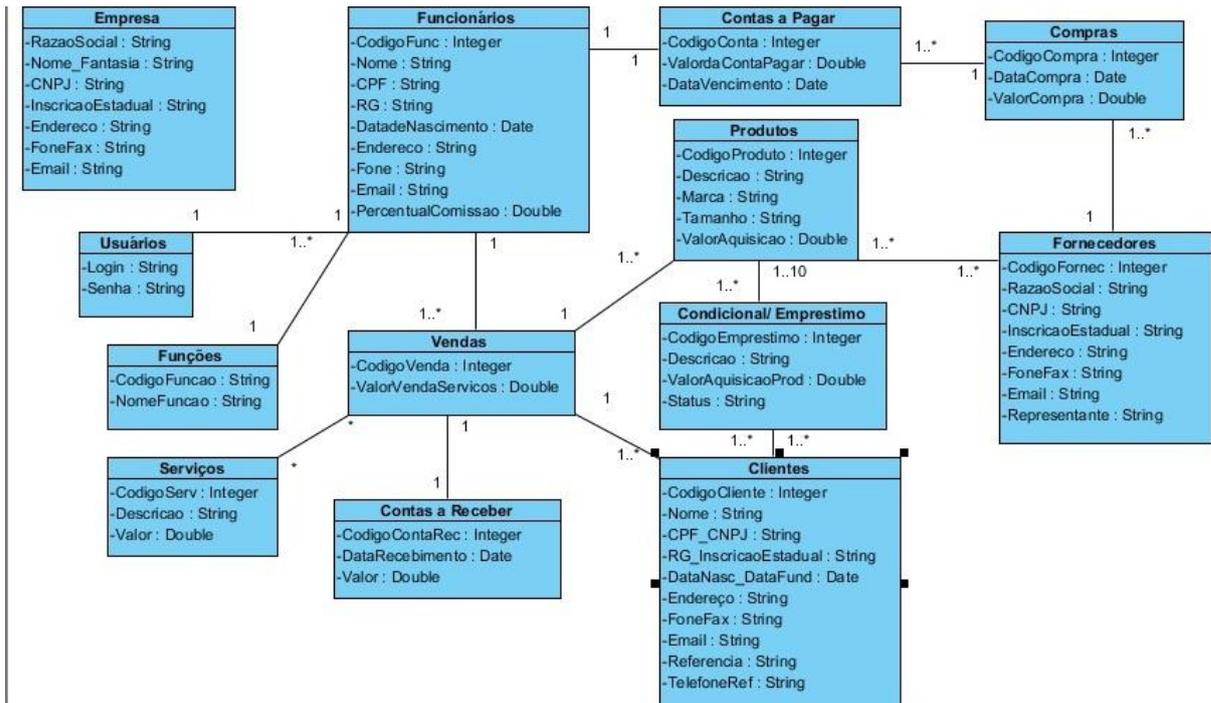


Figura 8 - Diagrama de classes

A seguir, nos Quadros 7 a 18, está a descrição das classes apresentadas na Figura 8.

Identificação:	Empresas
Descrição:	É previsto que nesta classe sejam requisitados os dados da empresa empregadora para que assim o cadastro seja efetuado com sucesso.
Atributos:	Razão Social (<i>String</i>): nome devidamente registrado sob o qual a pessoa jurídica se individualiza e exerce suas atividades. Nome Fantasia (<i>string</i>): designação popular de título do estabelecimento. CNPJ (<i>string</i>): Cadastro Nacional de Pessoa Jurídica. Inscrição estadual (<i>string</i>): registro do contribuinte no cadastro do ICMS (Imposto sobre Circulação de Mercadorias e Prestação de Serviços) mantido pela Receita Federal. Endereço (<i>string</i>): localização da empresa. Fone/Fax (<i>string</i>): número de contato com a empresa. Email (<i>string</i>): e-mail para contato com a empresa.
Métodos:	Cadastrar, Alterar, Consultar e Excluir

Quadro 7 - Classe Empresas

Identificação:	Funções
Descrição:	Essa classe é responsável pelos dados cadastrais das funções que são exercidas pelos colaboradores/ funcionários da empresa.
Atributos:	Código da Função (<i>integer</i>): número de identificação da função. Nome da Função (<i>string</i>): nome da função.
Métodos:	Cadastrar, Alterar, Consultar e Excluir

Quadro 8 - Classe Funções

Identificação:	Funcionário
Descrição:	Classe responsável pelo cadastramento dos colaboradores da empresa.
Atributos:	Código do Funcionário (<i>integer</i>): número de identificação do funcionário. Nome (<i>string</i>): Nome do funcionário. CPF (<i>string</i>): número do cadastro de pessoa física do funcionário. RG (<i>string</i>): número do registro geral do funcionário. Data de Nascimento (<i>date</i>): data de nascimento do funcionário. Endereço (<i>string</i>): localização residencial. Fone (<i>string</i>): telefone para contato. E-mail (<i>string</i>): e-mail para contato. Percentual de comissão (<i>double</i>): valor percentual que o funcionário ganha por venda efetuada.
Métodos:	Cadastrar, Alterar, Consultar, Excluir e Vincular Funções
Observações:	Esta classe está vinculada com a classe “Funções”

Quadro 9 - Classe Funcionários

Identificação:	Clientes
Descrição:	Essa classe tem como objetivo manter o cadastro os clientes da empresa.
Atributos:	Código do cliente (<i>integer</i>): número de identificação do cliente. Nome (<i>string</i>): nome do cliente. CPF_CNPJ (<i>string</i>): número do cadastro de pessoa física ou número do cadastro nacional de pessoa jurídica. RG_Inscrição Estadual (<i>string</i>): número do registro geral ou número da inscrição estadual do cliente. Data de Nascimento_ Data de Fundação (<i>date</i>): data de nascimento ou data de fundação do cliente. Endereço (<i>string</i>): localização do cliente. Fone/Fax (<i>string</i>): telefone para contato com o cliente. E-mail (<i>string</i>): e-mail para contato. Referencia (<i>string</i>): referências sobre o cliente. Telefone da Referencia (<i>string</i>): telefone para contato com a referencia.
Métodos:	Incluir, Alterar, Consultar e Excluir.

Quadro 10 - Classe Clientes

Identificação:	Fornecedores
Descrição:	Essa classe prevê os registros de fornecedores da empresa.
Atributos:	Código do Fornecedor (<i>integer</i>): número de identificação do fornecedor. Razão Social (<i>string</i>): nome devidamente registrado sob o qual a pessoa jurídica se individualiza e exerce suas atividades. CNPJ (<i>string</i>): número do cadastro nacional de pessoa jurídica. Inscrição estadual (<i>string</i>): registro do contribuinte no cadastro do ICMS mantido pela Receita Federal. Endereço (<i>string</i>): localização do fornecedor. Fone/Fax (<i>string</i>): telefone para contato com o fornecedor. E-mail (<i>string</i>): e-mail para contato. Representante (<i>string</i>): nome do funcionário que representa o fornecedor

	na região em que a empresa se encontra.
Métodos:	Incluir, Alterar, Consultar e Excluir.

Quadro 11 - Classe Fornecedores

Identificação:	Condicionais/ Empréstimos
Descrição:	Classe responsável pelos dados cadastrais dos condicionais/empréstimos.
Atributos:	Código do Empréstimo (<i>string</i>): número de identificação do condicional/empréstimo. Descrição (<i>string</i>): descrição dos produtos emprestados. Valor de Aquisição dos Produtos (<i>double</i>): valor de aquisição dos produtos emprestados. Status (<i>string</i>): estado (<i>status</i>) do empréstimo.
Métodos:	Incluir, Alterar, Consultar e vincular “Cliente”.
Observações:	Esta classe está vinculado com a classe “Cliente”.

Quadro 12 - Classe Condicional/Empréstimos

Identificação:	Compras
Descrição:	Classe responsável pelos dados cadastrais das compras efetuadas pela empresa.
Atributos:	Código de compra (<i>integer</i>): número de identificação da compra. Data de compra (<i>date</i>): data que foi efetuada a compra. Valor da compra (<i>double</i>): valor da compra.
Métodos:	Incluir, Alterar, Consultar, Excluir e vincular “Fornecedor”.
Observações:	Classe vinculada com a classe “Fornecedor”.

Quadro 13 - Classe Compras

Identificação:	Contas a pagar
Descrição:	Classe responsável pelos dados cadastrais das contas a pagar da empresa.
Atributos:	Código da conta (<i>integer</i>): número de identificação da conta. Valor da conta a pagar (<i>double</i>): valor da conta a pagar. Data de vencimento (<i>date</i>): data de vencimento da conta.
Métodos:	Incluir, Alterar, Consultar, Excluir e vincular compra.
Observações:	Classe vinculada a classe “Compra”.

Quadro 14 - Classe Contas a pagar

Identificação:	Produtos
Descrição:	Classe responsável pelos dados cadastrais dos produtos oferecidos pela empresa.
Atributos:	Código Produto (<i>integer</i>): número de identificação do produto. Descrição (<i>string</i>): descrição do produto. Marca (<i>string</i>): marca do produto. Tamanho (<i>string</i>): tamanho do produto.

	Valor aquisição (<i>double</i>): valor de aquisição do produto.
Métodos:	Incluir, Alterar, Consultar, Excluir e vincular fornecedores.
Observações:	Classe vinculada com a classe “Fornecedores”.

Quadro 15 - Classe Produtos

Identificação:	Serviços
Descrição:	Classe responsável pelos dados cadastrais dos serviços prestados pela empresa.
Atributos:	Código do serviço (<i>integer</i>): número de identificação do serviço. Descrição (<i>string</i>): descrição do serviço prestado. Valor (<i>double</i>): valor do serviço.
Métodos:	Incluir, Alterar, Consultar e Excluir.

Quadro 16 - Classe Serviços

Identificação:	Vendas
Descrição:	Classe responsável pelos dados cadastrais das vendas realizadas pela empresa.
Atributos:	Código Venda (<i>integer</i>): número de identificação da venda. Valor (<i>double</i>): valor da venda.
Métodos:	Incluir, Alterar, Consultar, Excluir e vincular cliente.
Observações:	Classe vinculada com a classe “Cliente”

Quadro 17 - Classe Vendas

Identificação:	Contas a receber
Descrição:	Classe responsável pelos dados cadastrais das contas a receber da empresa.
Atributos:	Código conta a receber (<i>integer</i>): número de identificação da conta a receber. Descrição (<i>string</i>): descrição da conta a receber. Valor (<i>double</i>): valor da conta a receber.
Métodos:	Incluir, Alterar, Consultar e Excluir.

Quadro 18 - Classe Contas a receber

Identificação:	Usuários
Descrição:	Classe responsável pelos dados cadastrais dos usuários do sistema.
Atributos:	<i>Login</i> (<i>string</i>): <i>login</i> do usuário do sistema. <i>Senha</i> (<i>string</i>): <i>senha</i> do usuário do sistema.
Métodos:	Incluir, Alterar, Consultar e Excluir.

Quadro 19 - Classe Usuários

A Figura 9 apresenta o diagrama de entidades e relacionamentos do banco de dados.

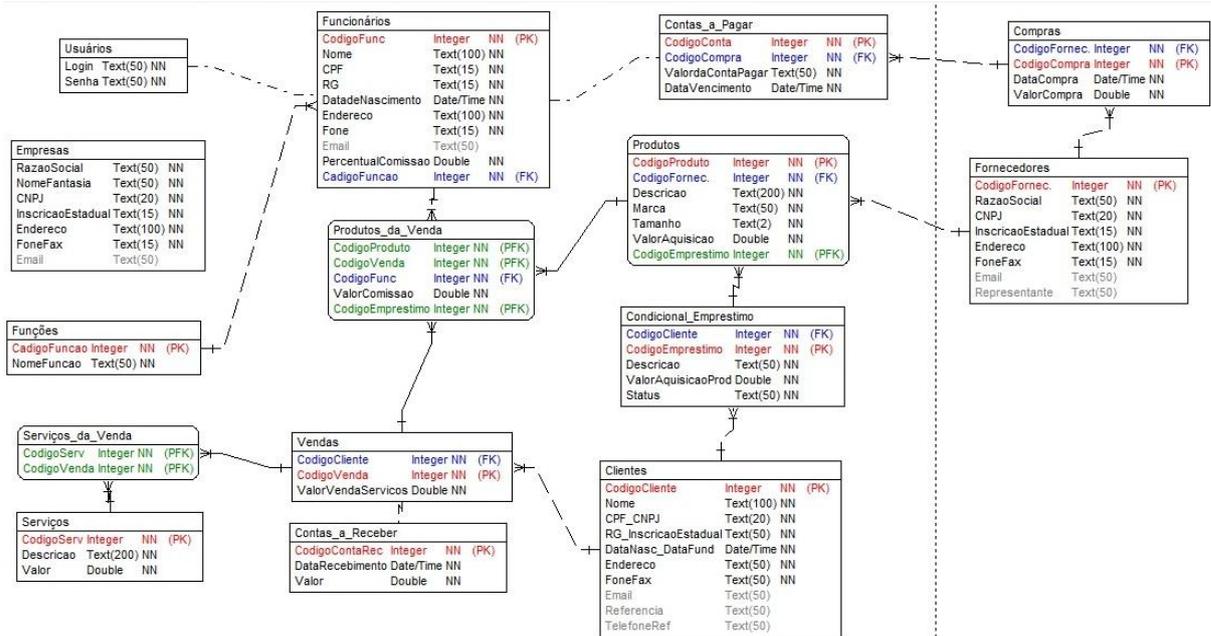


Figura 9 - Diagrama de entidade relacionamento do banco de dados

Os Quadros 19 a 30 apresentam a descrição das tabelas que compõem o banco de dados, conforme expõe a Figura 9.

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
RazaoSocial	String	Não	Não	Não	
NomeFantasia	String	Não	Não	Não	
CNPJ	String	Não	Não	Não	
InscricaoEstadual	String	Não	Não	Não	
Endereco	String	Não	Não	Não	
FoneFax	String	Não	Não	Não	
Email	String	Sim	Não	Não	

Quadro 19 - Tabela empresa

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
CodigoFuncao	Integer	Não	Sim	Sim	Auto-denominado
NomeFuncao	String	Não	Não	Não	

Quadro 20 - Tabela funções

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
CodigoFunc	<i>Integer</i>	Não	Sim	Sim	Auto-denominado
Nome	<i>String</i>	Não	Não	Não	
CPF	<i>String</i>	Não	Não	Não	
RG	<i>String</i>	Não	Não	Não	
DatadeNascimento	<i>Date</i>	Não	Não	Não	
Endereco	<i>String</i>	Não	Não	Não	
Fone	<i>String</i>	Não	Não	Não	
Email	<i>string</i>	Sim	Não	Não	
PercentualComissao	<i>Double</i>	Não	Não	Não	

Quadro 21 - Tabela funcionários

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
CodigoCliente	<i>integer</i>	Não	Sim	Sim	Auto-denominado
Nome	<i>String</i>	Não	Não	Não	
CPF_CNPJ	<i>String</i>	Não	Não	Não	
RG_InscricaoEstadual	<i>String</i>	Não	Não	Não	
DataNasc_DataFund	<i>Date</i>	Não	Não	Não	
Endereco	<i>String</i>	Não	Não	Não	
FoneFax	<i>String</i>	Não	Não	Não	
Email	<i>String</i>	Sim	Não	Não	
Referencia	<i>String</i>	Sim	Não	Não	
TelefoneRef	<i>String</i>	Sim	Não	Não	

Quadro 22 - Tabela clientes

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
CodigoFornecedor	<i>Integer</i>	Não	Sim	Sim	Auto-denominado
RazaoSocial	<i>String</i>	Não	Não	Não	
CNPJ	<i>String</i>	Não	Não	Não	
InscricaoEstadual	<i>String</i>	Não	Não	Não	
Endereco	<i>String</i>	Não	Não	Não	
FoneFax	<i>String</i>	Não	Não	Não	
Email	<i>String</i>	Sim	Não	Não	
Representante	<i>String</i>	Sim	Não	Não	

Quadro 23 - Tabela fornecedores

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
CodigoEmprestimo	<i>Integer</i>	Não	Sim	Sim	Auto-denominado
Descricao	<i>String</i>	Não	Não	Não	
ValorAquisicaoProd	<i>String</i>	Não	Não	Não	
Status	<i>String</i>	Não	Não	Não	

Quadro 24 - Tabela empréstimo/condicional

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
CodigoCompra	<i>Integer</i>	Não	Sim	Sim	Auto-denominado
DataCompra	<i>Date</i>	Não	Não	Não	
ValorCompra	<i>Double</i>	Não	Não	Não	

Quadro 25 - Tabela compras

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
CodigoCompra	<i>integer</i>	Não	Sim	Sim	Auto-denominado
ValordaContaPagar	<i>double</i>	Não	Não	Não	
DataVencimento	<i>date</i>	Não	Não	Não	

Quadro 26 - Tabela contas a pagar

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
CodigoProduto	<i>Integer</i>	Não	Sim	Sim	Auto-denominado
Descricao	<i>String</i>	Não	Não	Não	
Marca	<i>String</i>	Não	Não	Não	
Tamanho	<i>String</i>	Não	Não	Não	
ValorAquisicao	<i>Double</i>	Não	Não	Não	

Quadro 27 - Tabela produtos

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
CodigoServ	<i>Integer</i>	Não	Sim	Sim	Auto-denominado
Descricao	<i>String</i>	Não	Não	Não	
Valor	<i>Double</i>	Não	Não	Não	

Quadro 28 - Tabela serviços

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
CodigoVenda	<i>Integer</i>	Não	Sim	Sim	Auto-denominado
ValorVendaServicos	<i>Double</i>	Não	Não	Não	

Quadro 29 - Tabela vendas

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
CodigoContaRec	<i>Integer</i>	Não	Sim	Sim	Auto-denominado
DataRecebimento	<i>Date</i>	Não	Não	Não	
Valor	<i>Double</i>	Não	Não	Não	

Quadro 30 - Tabela contas a receber

Campo	Tipo	Nulo	Chave primária	Chave estrangeira	Observações
Login	<i>String</i>	Não	Não	Não	
Senha	<i>String</i>	Não	Não	Não	

Quadro 31 - Tabela Usuários

4.3 DESCRIÇÃO DO SISTEMA

A Figura 10 apresenta a tela de *login* e senha para acesso ao sistema. Essa tela permite a validação se o usuário existe. Os dois campos dessa tela são de preenchimento obrigatório.

Figura 10 - Tela de *login* e senha

Na Figura 11 é apresentada a tela inicial do software. Nela estão os menus Cadastros, Cadastros Administrativos (que somente é habilitado para o usuário que possui *login* e senha de administrador), Relatórios, Gráficos, Fazer Logoff e Ajuda. Na parte inferior da tela é apresentado o nome do usuário logado, a data e o horário do sistema operacional.



Figura 11 - Tela inicial do sistema

Na imagem da Figura 12 são apresentadas as opções no menu Cadastros, sendo elas o cadastramento de Clientes, Vendas, Empréstimos, Produtos e por fim a opção Sair.



Figura 12 - Menu cadastros

A Figura 13 apresenta as opções que o menu Cadastros Administrativos possui: Usuários, Empresas, Funções, Funcionários, Fornecedores, Compras, Serviços, Contas a Pagar, Contas a Receber e a opção Sair. As opções desse menu só estarão disponíveis aos usuários com perfil administrador.

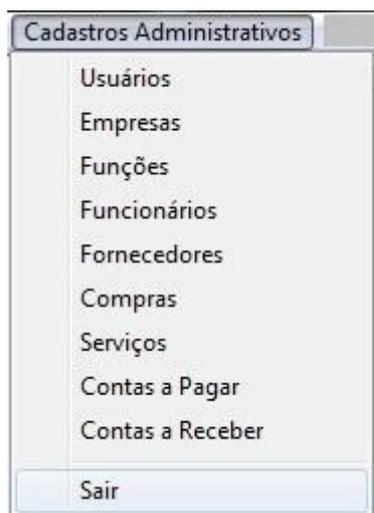


Figura 13 - Menu cadastros administrativos

O menu Relatórios pode ser visualizado na Figura 14. Nesta opção o usuário pode verificar os relatórios dos Clientes, Fornecedores, Produtos, Serviços, Contas a Pagar e Contas a Receber cadastrados no sistema.



Figura 14 - Menu relatórios

A Figura 15 mostra o menu Gráficos no qual o usuário pode visualizar graficamente as contas a pagar e a receber cadastradas no sistema.

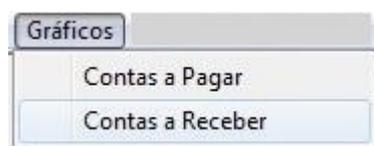


Figura 15 - Menu gráficos

Na opção Fazer *Logoff*, o usuário pode acessar o aplicativo com outro *login* e senha sem precisar finalizar o sistema. No menu Ajuda são encontradas informações sobre o *software* e seus desenvolvedores.

A tela de Cadastro de Clientes é representada pela Figura 16. No lado esquerdo dessa tela estão os botões Novo, Alterar, Excluir, Cancelar, Salvar e Fechar além das informações que devem ser obrigatoriamente cadastradas como Nome, CPF/CNPJ, RG/Inscrição Estadual,

Data de Nascimento/Data de Fundação, Endereço, Fone/Fax, *E-mail*, Referência e Telefone da Referência lembrando que o campo Código do Cliente é auto-incremento. Na parte inferior da tela, há um *grid* que permite visualizar todos os cadastrados já efetuados.

The screenshot shows a web application window titled "Sistema de Gerenciamento - Cadastro de Clientes". The interface includes a menu bar with options like "Cadastro", "Cadastros Administrativos", "Relatórios", "Gráficos", "Ajuda", and "Exec. Logout". Below the menu is a form for entering client data. The form fields are as follows:

- Mantençaõ de Dados: Código do Cliente (1)
- Nome: Guilherme Carneil
- CPF/CNPJ: 999.000.111.22
- RIS/Inscrição Estadual: 2.223.456.1
- Data de Nascimento/Data de Fundação: 24/07/1992
- Endereço: Rua Tapajó, 513
- Fone/Fax: (46)9990-7788
- E-mail: guilherme@gmail.com
- Referência: Bruna, Telefone da Referência: (46)9990-8645

At the bottom of the form is a "Consulta de Dados" section containing a data grid with the following content:

Código	Nome	CPF/CNPJ	RIS/Inscrição Estadual	Data de Nascimento/Fundação	Endereço
1	Guilherme Carneil	999.000.111.22	2.223.456.1	24/07/1992	Rua Tapajó, 513

The status bar at the bottom indicates the user is "Usuário Admin", the date is "domingo, 21 de abril de 2013", and the time is "23:23:05".

Figura 16 - Tela cadastro de clientes

A Figura 17 apresenta a tela de cadastrado das vendas efetuadas. Nela há os botões Novo, Alterar, Excluir, Cancelar, Salvar e Fechar, além dos campos que devem ser cadastrados obrigatoriamente como Valor e vinculação do Cliente para o qual foi feita a venda. A informação Código da Venda é auto-incremento. Na parte inferior da tela está o *grid* que apresenta todos os cadastros já realizados.

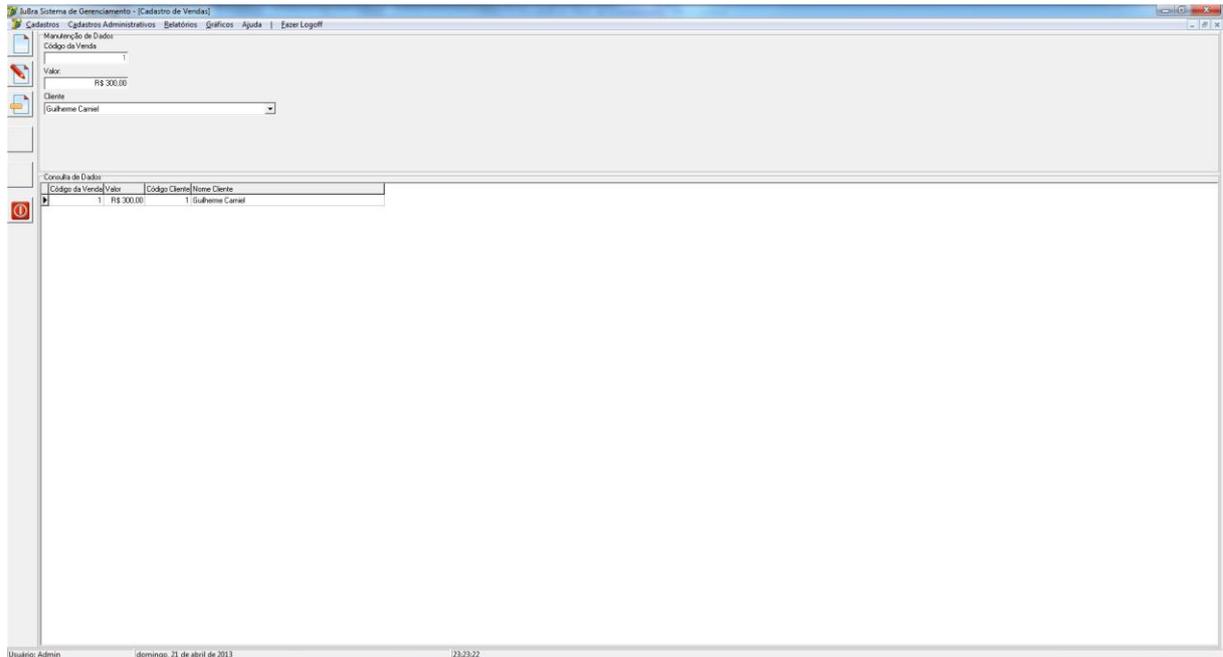


Figura 17 - Tela cadastros de vendas

Na Figura 18 está a tela de cadastros dos empréstimos. Na parte esquerda da tela há os botões Novo, Alterar, Excluir, Cancelar, Salvar e Fechar. Há também as informações obrigatórias nos cadastros: Nome Cliente que deve ser vinculado com os Clientes já cadastrados no sistema, Valor, Status, Descrição e Produto, campo que também deve ser vinculado com os produtos já cadastrados no sistema. E na parte inferior da tela são apresentados os cadastros já existentes e que podem ser editados ou excluídos.

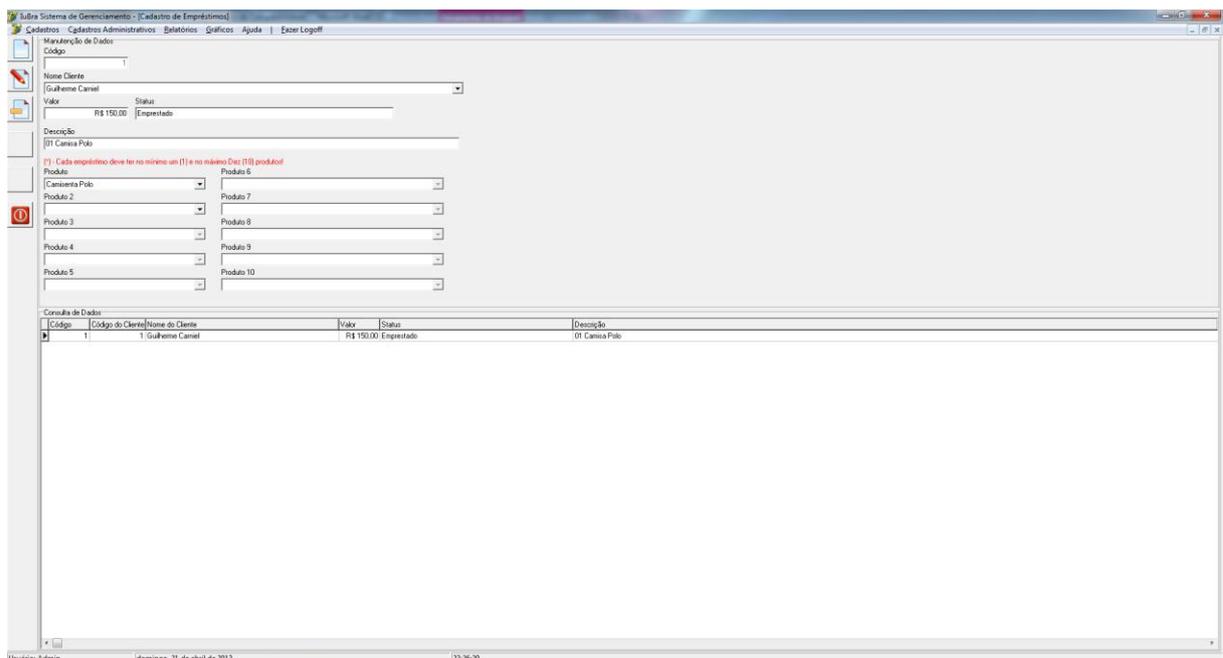


Figura 18 - Tela cadastros de empréstimos

Na Figura 19 é apresentada a tela de cadastro de produtos. Nela há os botões padrões para cadastro e as informações obrigatórias como Fornecedor que deve ser vinculado com os fornecedores já cadastrados no sistema, Descrição, Marca, Tamanho e Valor. Essa tela também permite a visualização dos dados dos cadastros já realizados.

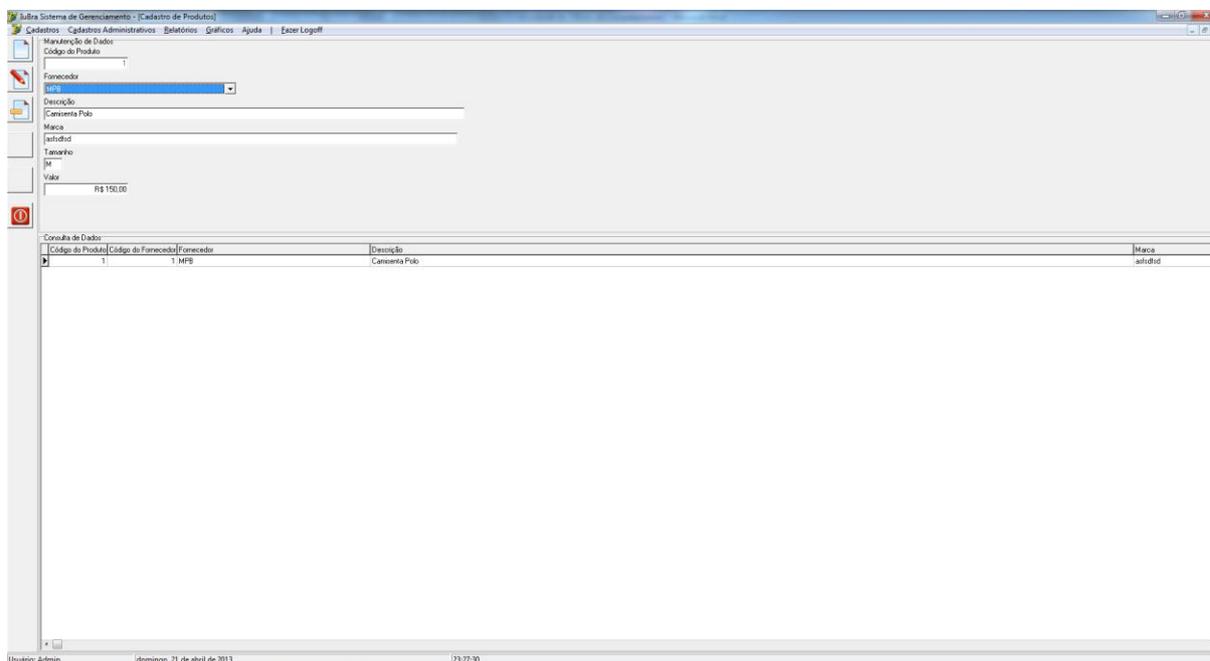


Figura 19 - Tela cadastros de produtos

A tela de cadastros de usuários pode ser visualizada na Figura 20. Nela, além dos botões padrões do cadastro, há as informações *login*, senha e o vínculo dos funcionários já cadastrados e um *checkbox* que quando marcado disponibiliza aos usuários recém-cadastrados a possibilidade de ter acesso a todas as informações do sistema. Por último há está o *grid*, com a listagem dos cadastros já realizados.

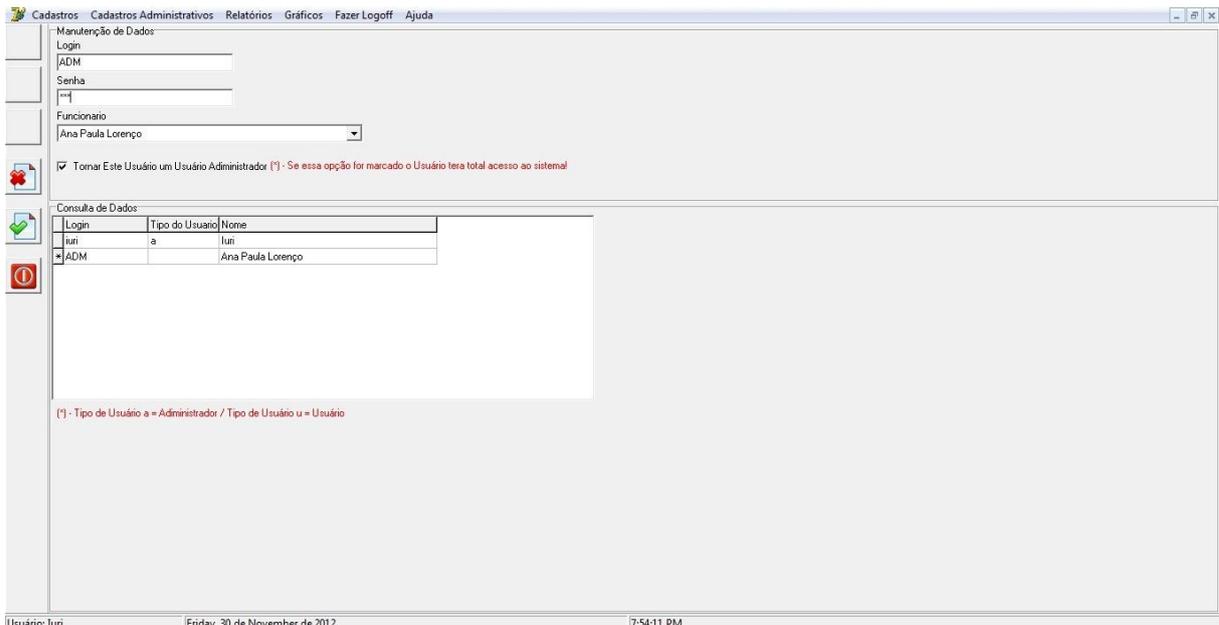


Figura 20 - Tela cadastros de login e senha

Na Figura 21 pode ser visualizada a tela de cadastramento de Empresas. No lado esquerdo da tela estão os botões padrão de cadastros. Na parte central encontra-se os dados a serem cadastrados como Razão Social, Nome Fantasia, CNPJ, Inscrição Estadual, Endereço, Fone/Fax e *E-mail*. Essa tela apresenta a listagem dos registros já realizados.

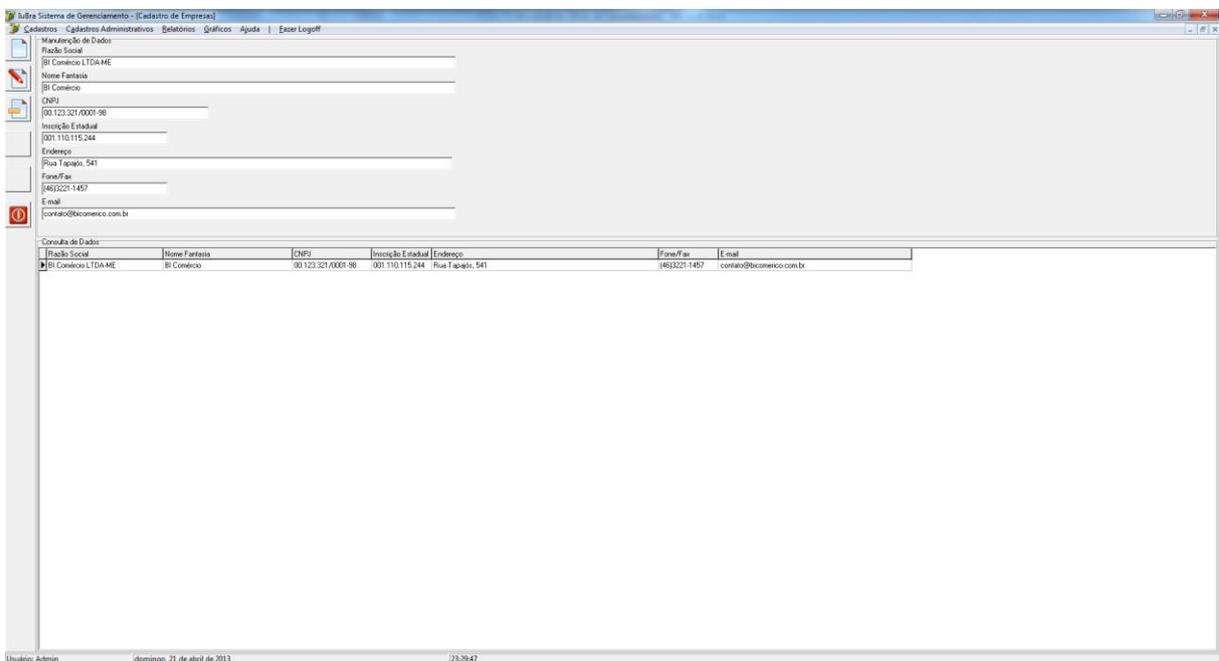


Figura 21 - Tela cadastros de empresas

A tela da Figura 22 disponibiliza ao usuário o cadastramento das funções que os colaboradores exercerão. Nesta tela são encontrados os botões padrões de um cadastro, as informações cadastrais como Código da Função que é auto-incremento e o Nome da Função.

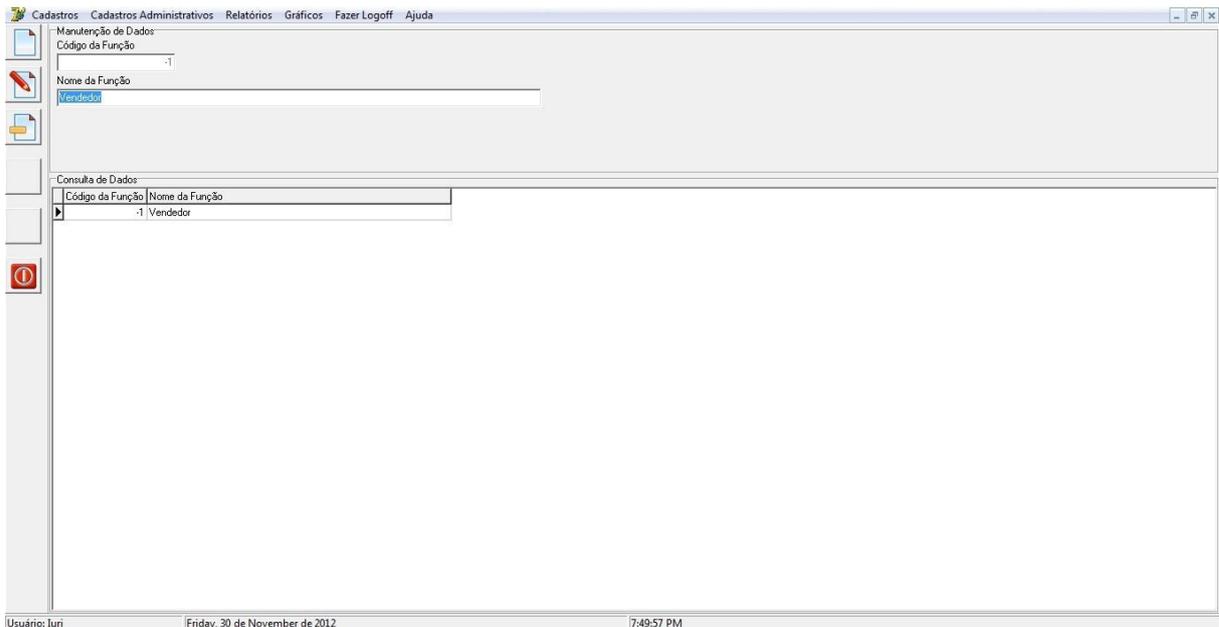


Figura 22 - Tela cadastros de funções

A Figura 23 ilustra a tela de cadastro dos colaboradores. Nela estão os botões padrão para o cadastro que são Novo, Alterar, Excluir, Cancelar, Salvar e Fechar. Também podem ser visualizados os dados a serem cadastrados como Código do Funcionário, que é auto-incremento, Nome, CPF, RG, Data de Nascimento, Endereço, Fone, E-mail, Função (campo que deve ser vinculado com as funções já cadastradas no sistema) e Percentual de Comissão. Na parte inferior da tela está a listagem dos cadastros já realizados.

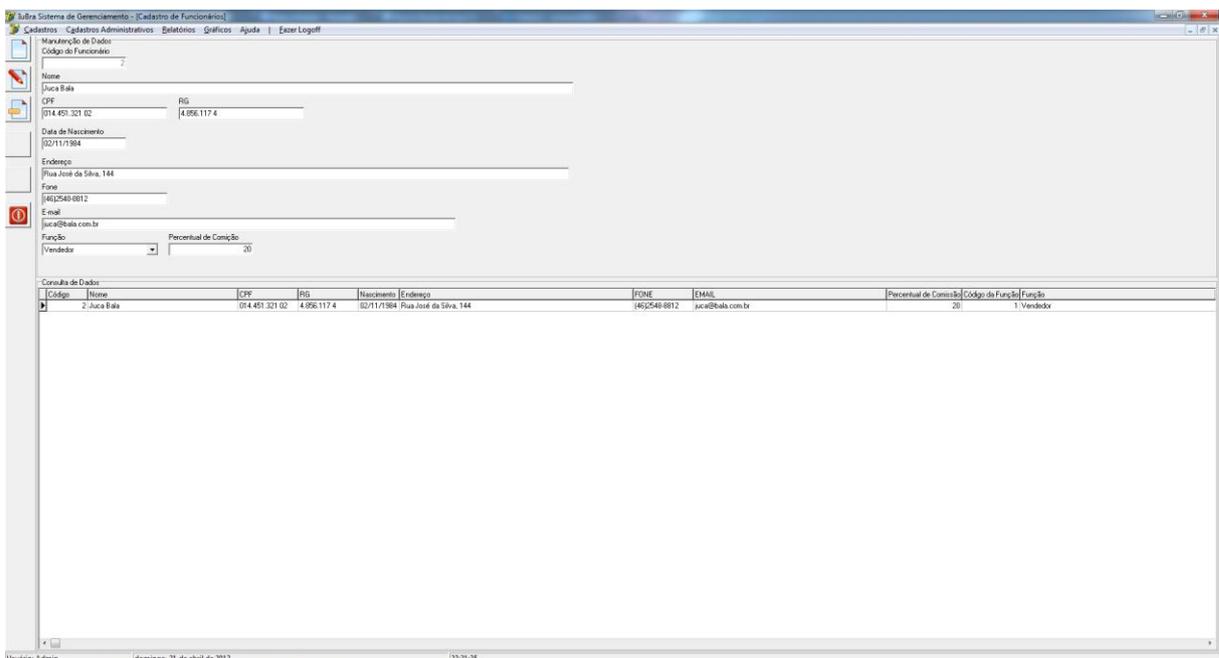


Figura 23 - Tela cadastros de funcionários

A Figura 24 apresenta ao usuário a tela para o cadastramento dos fornecedores. Nesta tela são encontrados os botões padrão de cadastro e as informações para cadastramento que são: Código do Fornecedor, auto-incremento, Razão Social, CNPJ, Inscrição Estadual, Endereço, Fone/Fax, *Email* e Representante.

Código	Razão Social	CNPJ	Inscrição Estadual	Endereço	Fone/Fax	E-mail
1	Brix LTDA	00.000.111/0909-22	Isento	Av. Majé	(46)7777-7777	brix@brix.com.br

Usuário: luni Friday, 30 de November de 2012 7:55:29 PM

Figura 24 - Tela cadastros de fornecedores

Na Figura 25 está o cadastro das compras efetuadas. Nessa tela estão os botões padrão do cadastro, as informações a serem cadastradas, Código Compras (que é auto-incremento), Fornecedor (que deve ser vinculado com os fornecedores já cadastrados no sistema), Data da Compra e Valor da Compra. Na parte inferior da tela ficam visíveis todos os cadastrados já realizados no *software*.

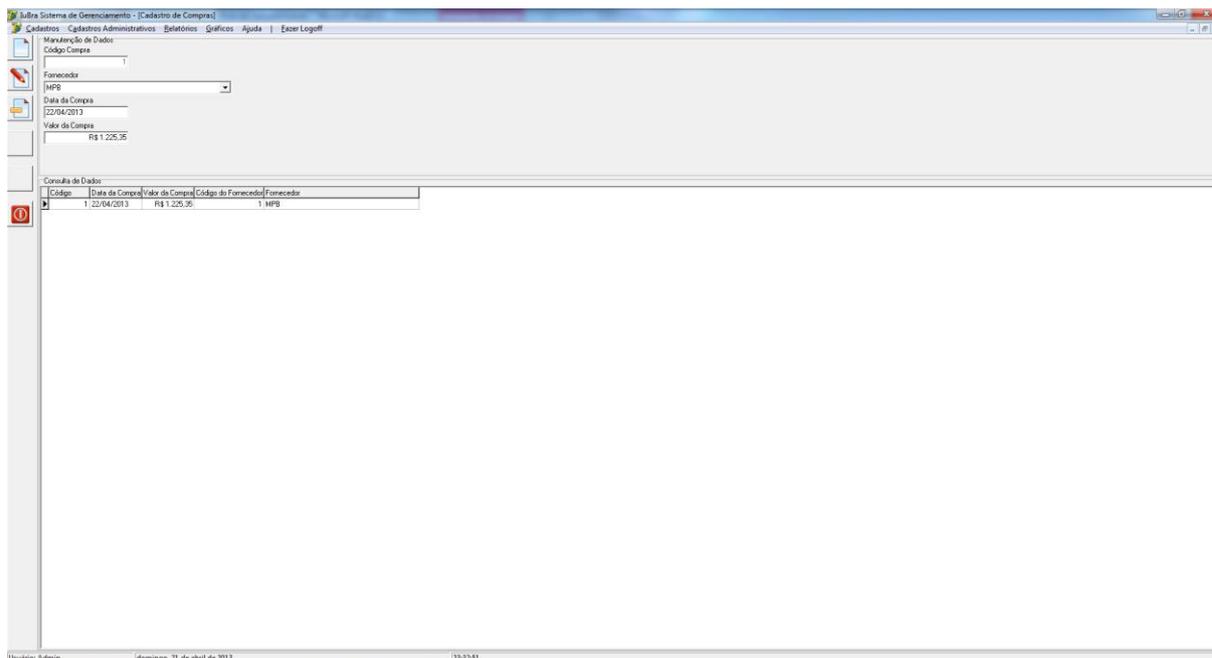


Figura 25 - Tela cadastros de compras

A imagem da Figura 26 apresenta o cadastro dos serviços prestados pela empresa. Nesta tela há os botões padrão do cadastro, as informações a serem cadastradas, Código (que é auto-incremento), Descrição e Valor.

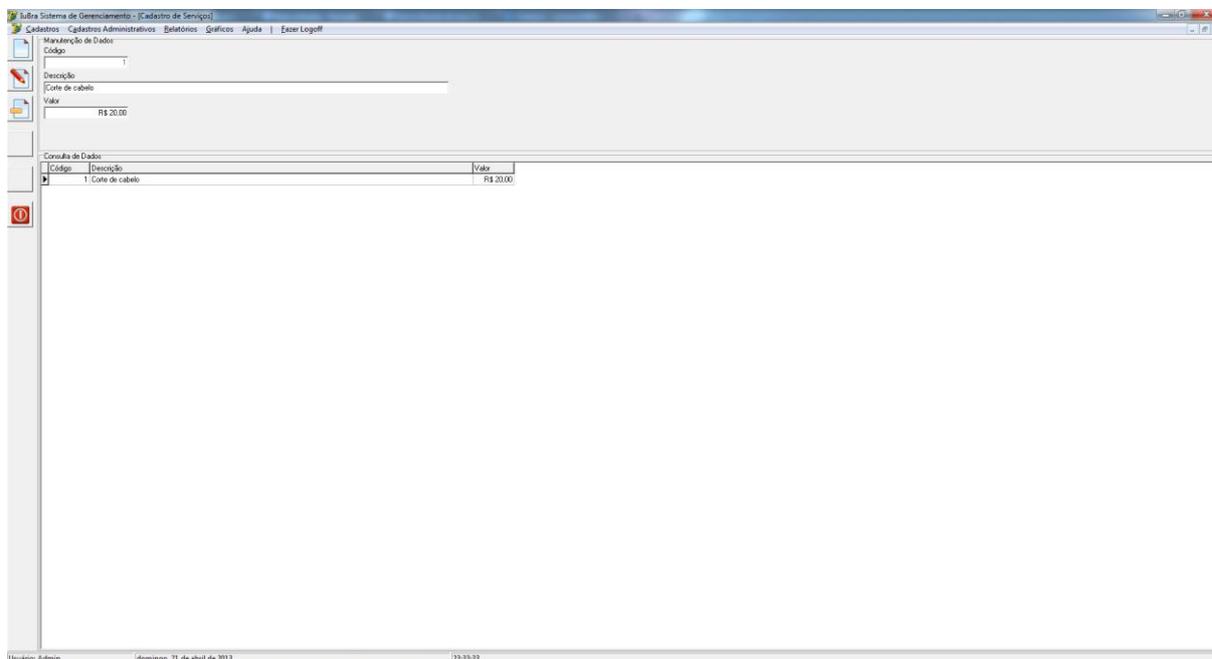


Figura 26 - Tela cadastros de serviços

Pode ser visualizada na Figura 27 a tela de cadastro das contas a pagar. Nela estão os botões padrões de um cadastro, as informações a serem cadastradas, Código Conta (auto-incremento), Código Compra (que deve ser vinculado com as compras já cadastradas), Valor,

Status (se a conta já está paga ou não paga) e vencimento. Assim, como nas demais telas de cadastros, por fim, há um *grid* que mostra todas as informações já cadastradas.

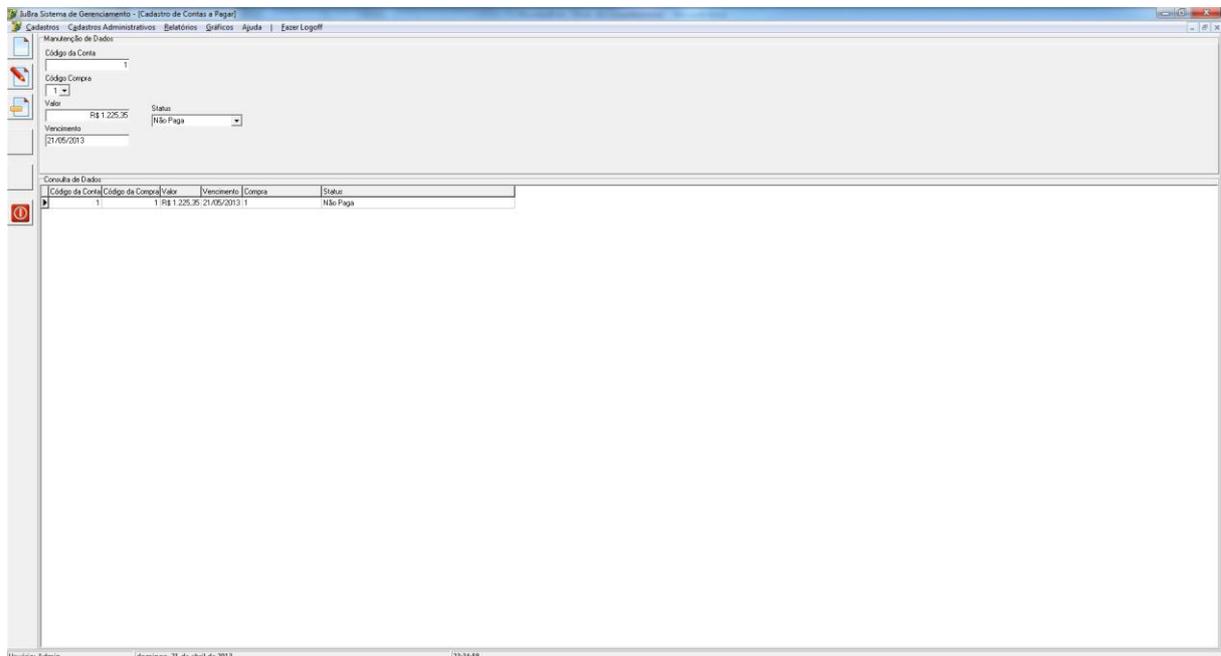


Figura 27 - Tela cadastros de contas a pagar

A Figura 28 apresenta ao usuário o cadastramento das Contas a Receber. No canto direito da mesma estão os botões padrão dos cadastros, no meio da tela estão as informações a serem cadastradas, Código, auto-incremento, Data de Recebimento, Valor e *Status* da Contas, recebida ou não recebida e na parte inferior da mesma está o *grid* que mostra todas as informações já cadastradas.

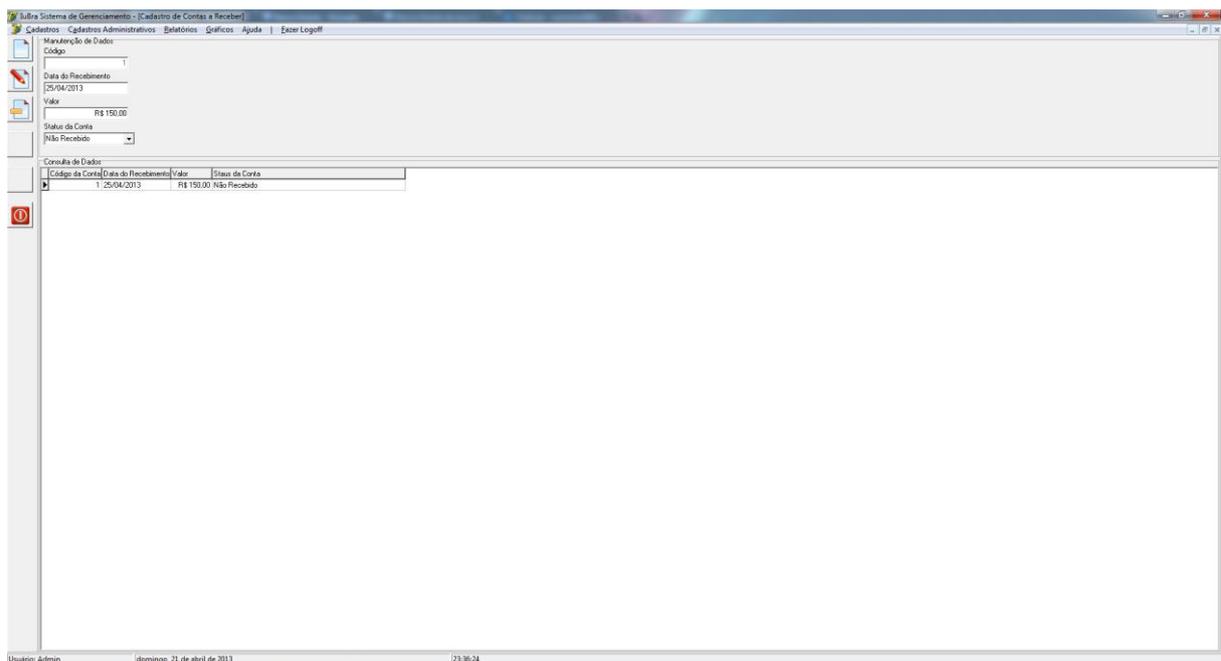


Figura 28 - Tela cadastros de contas a receber

Na Figura 29 pode ser visualizado o Relatório de Clientes. Este relatório mostra ao usuário as informações como Nome, Endereço, Telefone/Fax e *E-mail* cadastrados na tela de cadastros de Clientes.

Relatório de Clientes			
Nome	Endereço	Telefone/Fax	E-mail
Guilherme Camiel	Rua Tapajós, nº513	(40)9988-7788	bruuna_scheffer@hotmail.com
Bruna Scheffer Dias	Rua Arnaldo Jacobsen	(46)9909-8645	bruuna_scheffer@hotmail.com
			Total de Registros: 2

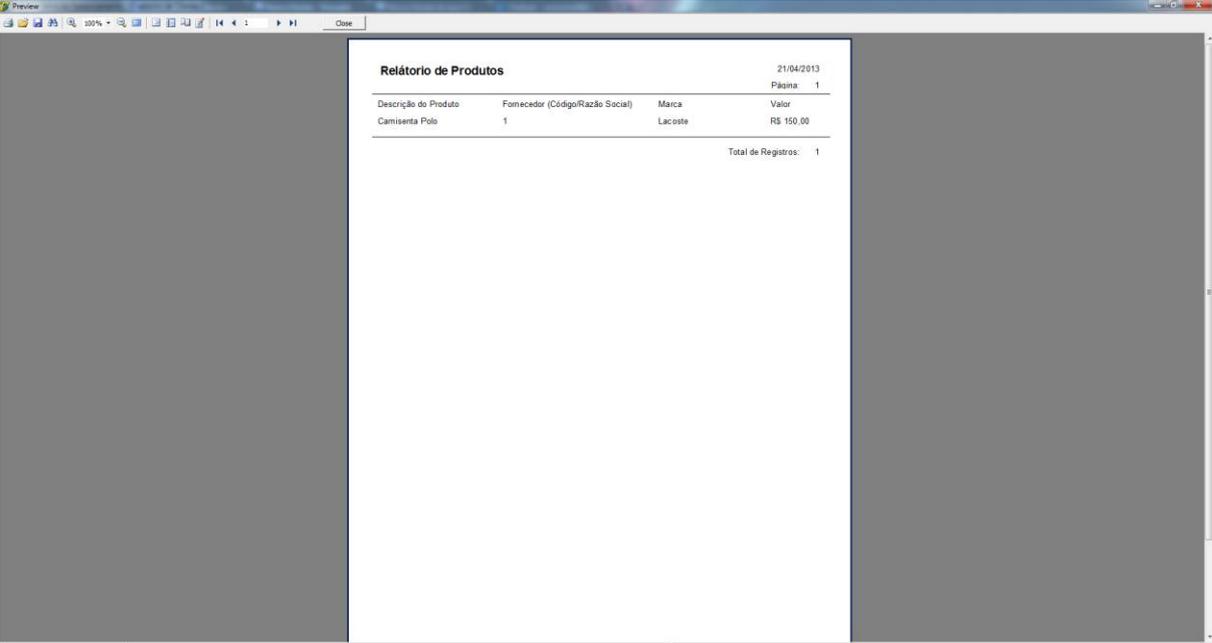
Figura 29 - Relatório de clientes

Na Figura 30 está o Relatório de Fornecedores e as informações cadastradas na tela de cadastro de Fornecedores como Razão Social, Endereço, Representante e Telefone/Fax.

Relatório de Fornecedores			
Razão Social	Endereço	Representante	Telefone/Fax
Brix LTDA	Av. Majé	Luiz Carlos Santos	(46)7777-7777
Dooping	Av. Balanço	Claudia Gonçalves	(09)9999-9999
			Total de Registros: 2

Figura 30 - Relatório de fornecedores

A imagem da Figura 31 mostra o Relatório de Produtos e os dados cadastrados na tela de cadastro dos mesmos, como Descrição, Fornecedor (Código/Razão Social), Marca e Valor.



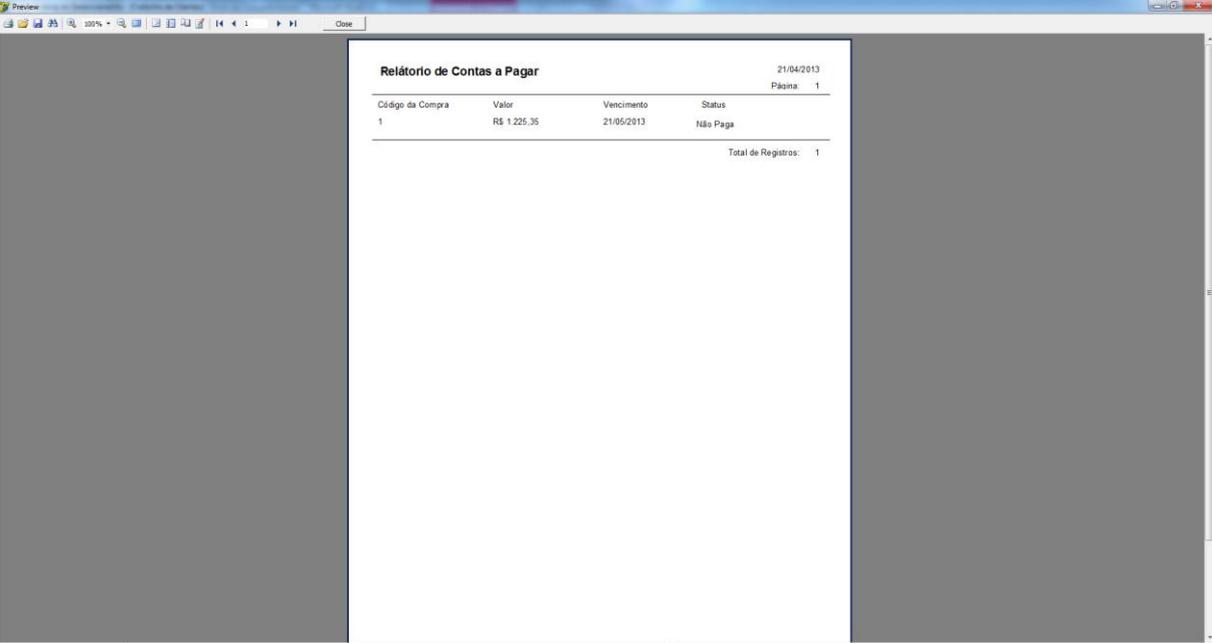
The screenshot shows a software window titled "Relatório de Produtos" with a date of 21/04/2013 and page number 1. The window contains a table with the following data:

Descrição do Produto	Fornecedor (Código/Razão Social)	Marca	Valor
Camisenta Polo	1	Lacoste	R\$ 150,00

Total de Registros: 1

Figura 31 - Relatório de produtos

O Relatório de Contas a Pagar pode ser visualizado pela Figura 32. Nele há as informações de Código da Compra, Valor, Vencimento e *Status*, informações essas cadastradas na tela de cadastros de contas a pagar.



The screenshot shows a software window titled "Relatório de Contas a Pagar" with a date of 21/04/2013 and page number 1. The window contains a table with the following data:

Código da Compra	Valor	Vencimento	Status
1	R\$ 1.225,35	21/05/2013	Não Paga

Total de Registros: 1

Figura 32 - Relatório de contas a pagar

O Relatório de Contas a Receber é exibido na Figura 33 assim como suas informações como Código da Conta, Valor, Vencimento e Status.

Relatório de Contas a Receber			
Código da Conta	Valor	Vencimento	Status
1	R\$ 150,00	25/04/2013	Não Recebido
			Total de Registros: 1

Figura 33 - Relatório de contas a receber

Na Figura 34 o usuário pode visualizar o gráfico referente a todas as Contas a Pagar cadastradas no sistema de acordo com o dia em que foram cadastradas.

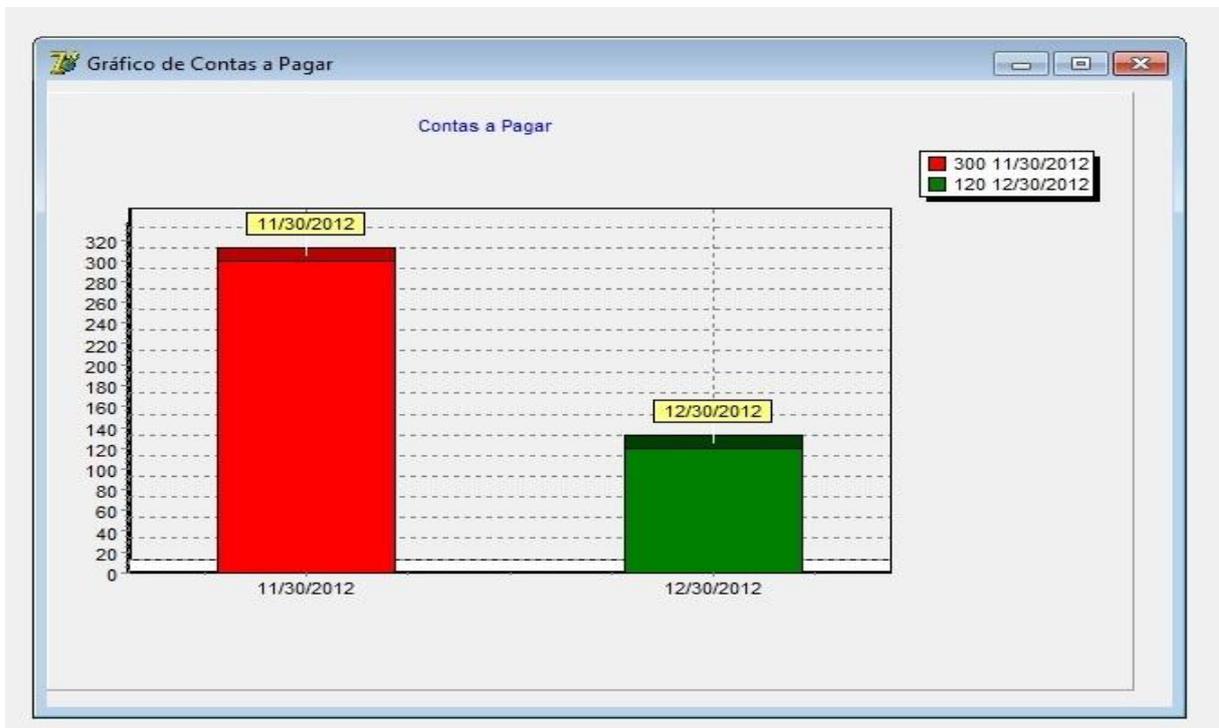


Figura 34 - Gráfico de contas a pagar

Na Figura 35 pode ser visualizado o gráfico de todas as Contas a Receber cadastradas no sistema de acordo com o dia em que foram cadastradas.

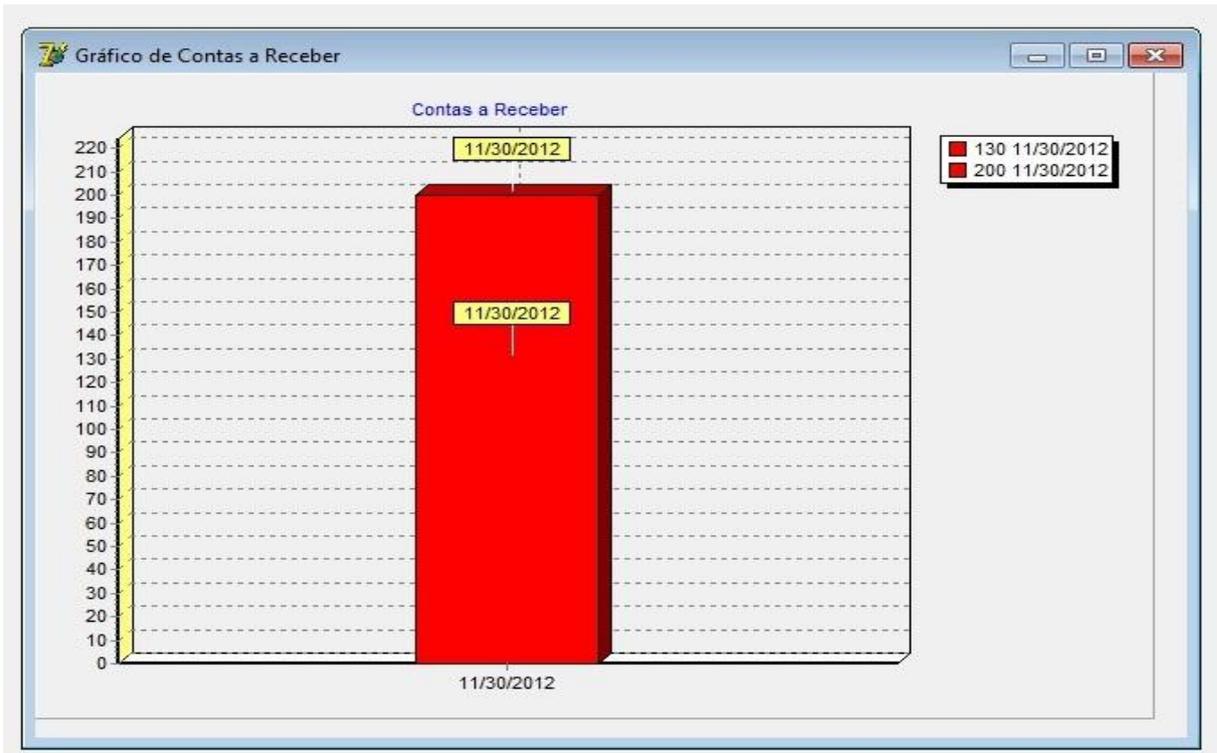


Figura 35 - Relatório de contas a receber

4.4 IMPLEMENTAÇÃO DO SISTEMA

Para que seja possível realizar a conexão com o banco de dados, foi criada uma *procedure* privada denominada *procedure ConectaBanco*. Essa *procedure* é apresentada na Listagem 1.

```
procedure TfrmMenu.ConectaBanco;
begin
    dm.SQLC.Connected := false;
    dm.SQLC.Params.Clear;
    dm.SQLC.Params.LoadFromFile('BD/configFirebird.ini');
    dm.SQLC.Connected := true;
end;
```

Listagem 1 - Procedure ConectaBanco

Essa *procedure* da Listagem 1 realiza a conexão com o banco de dados utilizando o formulário DM (*Data Module*), no qual estão os componentes necessários para realizar a conexão. Após fechar qualquer conexão existente e zerar todos os parâmetros da componente SQLC (*Structured Query Language Connection*), carrega-se os parâmetros de um arquivo de extensão “.ini” no qual estão todos os parâmetros necessários para a conexão funcionar corretamente. A linha de código “dm.SQLC.Connected := true” tem a função de ativar a

conexão após os parâmetros serem carregados, a mesma só ira funcionar se todos os parâmetros estiverem corretos.

Na implementação da Listagem 2, a *procedure TfrmLogin.logar* é acionada quando o botão logar da tela “Login” é clicado.

```

procedure TfrmLogin.logar;
begin
  try
    logado := false;
    qryLogin.SQL.Text := 'SELECT NOME, SENHA, TIPOUSUARIO FROM
USUARIO WHERE LOGIN = ' +
      QuotedStr(edtLogin.Text);
    qryLogin.Open;
    if qryLogin.IsEmpty then
      msgOK('Usuário não encontrado')
    else
      begin
        if qryLogin.FieldName('SENHA').AsString = edtSenha.Text
then
          begin
            msgOK('Usuário valido com sucesso!');
            logado := true;
            usuario := qryLogin.FieldName('NOME').AsString;
            tipoUsuario :=
qryLogin.FieldName('TIPOUSUARIO').AsString;
            self.Close;
          end
        else
          begin
            msgOK('Senha inválida!');
            edtSenha.SetFocus;
          end;
        end;
      qryLogin.close;
    except
      on E:Exception do
        begin
          msgOK('Erro ao validar o usuário');
          self.Close;
        end;
      end;
    end;
end;

```

Listagem 2 - Procedure TfrmLogin.logar

O código apresentado na Listagem 2 tem uma variável booleana para identificar se o usuário está logado no sistema ou não. O primeiro passo é atribuir valor “*false*” para essa variável “logado” e em seguida recuperar do banco de dados todas as informações referentes ao usuário digitado no campo “Usuário”. Caso o usuário indicado não existir, mostra-se uma mensagem informando que o usuário não foi encontrado. Caso contrário o sistema verifica se a senha informada é igual à senha cadastrada no banco de dados. Se essa informação for

verdadeira, atribui-se valor “true” para a variável “logado” e mostra-se a mensagem informando que o usuário foi validado.

A Listagem 3 apresenta a procedure *TfrmMenu.trocardeUsurio1Click(Sender: TObject)*.

```

procedure TfrmMenu.trocardeUsurio1Click(Sender: TObject);
var
  x: byte;
begin
  if MDIChildCount > 0 then
    for x:= 0 to Pred(MDIChildCount) do
      MDIChildren[x].Close;
  try
    ConectaBanco;
    //msgOk('Banco conectado!');
    Application.ShowMainForm := false;
    frmLogin := TfrmLogin.Create(nil);
    frmLogin.ShowModal;
    StatusBar1.Panels[0].Text := 'Usuário: '+frmLogin.usuario;
    //Valida se o usuario é Administrador do sistema, para
desabilitar alguns cadastros
    VerificarUsuario;
    FreeAndNil(frmLogin);
    Application.ShowMainForm := true;
    StatusBar1.Panels[1].Text := FormatDateTime('dddd, dd "de" mmmm
"de" yyyy', Date);

  except
    on E: Exception do
      begin
        msgOK('Erro ao conectar o banco: '+ E.Message);
        Application.Terminate;
      end;
  end;
end;
end;

```

Listagem 3 - TfrmMenu.trocardeUsurio1Click(Sender: TObject)

A *procedure* privada da Listagem 3 tem como função fechar todos os formulários filhos abertos dentro do formulário pai (Menu) e chamar novamente a função de conexão com o banco. Esta *procedure* permite a troca de usuários sem a necessidade de fechar o programa.

A implementação da Listagem 4 representa a função privada “DadosValidos” que é implementada em todos os cadastros, nesse código de exemplo é mostrado a função que valida os dados do cadastro de clientes.

```

function TfrmCadClientes.DadosValidos: boolean;
begin
  Result := true;
  if(edtCodigo.Text = '') then
    begin
      msgOk('Informe o Código do Cliente');
    end;
end;

```

```

        edtCodigo.SetFocus;
        result := false;
    end
else if(edtNome.Text = '') then
    begin
        msgOk('Informe o Nome do Cliente');
        edtNome.SetFocus;
        result := false;
    end
else if(edtCPF.Text = '') then
    begin
        msgOk('Informe o CPF/CNPJ do Cliente');
        edtCPF.SetFocus;
        result := false;
    end
else if(edtRG.Text = '') then
    begin
        msgOk('Informe o RG/Inscrição Estadual do Cliente');
        edtRG.SetFocus;
        result := false;
    end
else if(edtData.Text = '') then
    begin
        msgOk('Informe a Data de Nascimento/Fundação do Cliente');
        edtData.SetFocus;
        result := false;
    end
else if(edtEndereco.Text = '') then
    begin
        msgOk('Informe o Endereço do Cliente');
        edtEndereco.SetFocus;
        result := false;
    end
end;
end;

```

Listagem 4 - TfrmCadClientes.DadosValidos

Na função da Listagem 4, é verificado se todos os campos que não podem ser nulos estão com conteúdo informado. Isso é realizado para manter a integridade do banco de dados e para evitar qualquer erro devido à falta de informação em algum cadastro específico. Essa função não permite a compilação nem execução do projeto se ela não estiver presente em todos os cadastros.

No cadastro de produtos ao clicar no botão "Novo", chama-se a função de estado dos botões que habilitará e desabilitará algumas das demais opções. Em seguida o foco é colocado no primeiro componente e com o comando "cds.Append" é adicionado um registro em branco na última posição, sem salvar as informações no banco de dados. Isso será feito somente quando o botão "Salvar" for acionado. A Listagem 5 apresenta o código do botão "Novo".

```

procedure TfrmPadraoDB.btnNovoClick(Sender: TObject);
begin

```

```

EstadoBotoes (tebIncluir);
SelectFirst; //coloca o foco no primeiro componente
cds.Append;
end;

```

Listagem 5 - TfrmPadraoDB.btnNovoClick

Ao clicar no botão “Alterar”, é chamada a função de estado dos botões que irá novamente habilitar e desabilitar algumas das demais opções, após isso com o comando “cds.Edit” é informado que o registro em questão está em edição. Porém, sem realizar nenhuma alteração no banco de dados. As alterações serão realizadas somente quando o botão “Salvar” for acionado. O código do botão “Alterar” está na Listagem 6.

```

procedure TfrmPadraoDB.btnAlterarClick(Sender: TObject);
begin
  EstadoBotoes (tebAlterar);
  cds.Edit;
end;

```

Listagem 6 - TfrmPadraoDB.btnAlterarClick

Ao clicar no botão “Salvar”, a função “Dados Validos” que verifica se todos os dados obrigatórios estão preenchidos é chamada. Se tudo estiver de acordo, os dados são salvos no banco de dados com o comando “cds.Post”. Caso nenhum erro ocorra, a função de estado dos botões é chamada, habilitando os devidos botões. Caso algum erro ocorra, uma mensagem será apresentada. Na listagem 7 está o código do botão “Salvar”.

```

procedure TfrmPadraoDB.btnSalvarClick(Sender: TObject);
begin
  if DadosValidos then
  begin
    cds.Post;
    if cds.ApplyUpdates(0) = 0 then
      EstadoBotoes (tebCancelar)
    else
      msgOK('Erro ao gravar o registro!');
  end;
end;

```

Listagem 7 - TfrmPadraoDB.btnSalvarClick

Ao clicar no botão “Excluir”, chama-se a função de estado dos botões que habilitará e desabilitará algumas das demais opções e realiza-se a verificação se realmente existe um registro selecionado. Se tudo estiver correto, uma mensagem de confirmação para realmente deletar o registro é apresentada. Se a exclusão for confirmada chama-se mais uma vez a função de estado de botões e deleta-se o registro com o comando “cds.Delete”. Logo após realiza-se a verificação de que o registro foi excluído com sucesso. O código do botão “Excluir” está na Listagem 8.

```
procedure TfrmPadraoDB.btnExcluirClick(Sender: TObject);
begin
  EstadoBotoes(tebExcluir);
  if (not cds.IsEmpty) and
    (msgSN('Deseja excluir o registro selecionado?') = idyes) then
    begin
      EstadoBotoes(tebCancelar);
      cds.Delete;
      if cds.ApplyUpdates(0) <> 0 then
        msgOK('Erro excluir o registro!');
    end
  else
    EstadoBotoes(tebCancelar);
end;
```

Listagem 8 - TfrmPadraoDB.btnExcluirClick

5 CONCLUSÃO

O desenvolvimento deste trabalho objetivou a criação de um *software* para o controle de uma loja de confecções que possui um salão de beleza vinculado. Esse sistema provê o controle de compras, vendas, condicional e serviços de salão de beleza.

Para o desenvolvimento deste trabalho foram utilizadas as ferramentas Case Studio e Visual Paradigm na modelagem do sistema. Essas ferramentas foram de grande ajuda para realizar a análise do sistema. Elas facilitaram o desenvolvimento do diagrama casos de uso, de classes e de entidades e relacionamentos. Permitindo identificar as entidades de implementação (classe) e as entidades do banco de dados (tabelas).

Outra ferramenta utilizada no desenvolvimento do programa foi Delphi 7 que permite o desenvolvimento de *software* com a linguagem Delphi. Essa foi a linguagem utilizada para implementar o programa resultado deste trabalho. A escolha desta linguagem foi decorrente das facilidades que a mesma possui, considerando o escopo de conhecimento dos autores deste trabalho.

As facilidades encontradas em torno das ferramentas utilizadas foram em decorrência que os autores do trabalho já haviam desenvolvido escopos semelhantes com as mesmas, facilitando, assim, o desenvolvimento da análise e programação do *software*. Por outro lado as dificuldades na utilização de tais ferramentas ocorreram durante o desenvolvimento dos relatórios do sistema. Pesquisa na Internet foram realizadas para entendimento do funcionamento e leiaute das ferramentas.

O sistema desenvolvido permitirá ao usuário um melhor controle nos empréstimos de mercadorias, vendas e compras realizadas na loja de confecção e assim como dos serviços oferecidos no salão de beleza. O *software* permitirá, ainda, o cadastramento de clientes, fornecedores, usuários, funções, serviços realizados no salão de beleza, empresa, produtos que serão vendidos na loja de confecção e compras realizadas pela loja. Essas funcionalidades oferecem aos usuários do sistema uma melhor administração dos serviços realizados e dos produtos comprados e vendidos.

REFERÊNCIAS

ARANTES, Carlos Guilherme. **Nova versão 2009 do Delphi**. Disponível em: <http://www.guilherme.inf.br/index.php?option=com_content&view=article&id=73:nova-versao-2009-do-delphi&catid=68:artigos-sobre-delphi&Itemid=77>. Acesso em: 20 de nov. de 2012.

CANTÚ, Marco. **Dominando o Delphi 6 "A Bíblia"**. São Paulo. Makron Books, 2000.

CANTÚ, Marco. **Dominando o Delphi 7 "A Bíblia"**. São Paulo: Makron Books, 2003.

CANTÚ, Carlos, H. **Conheça o Firebird em 2 minutos**. Disponível em: <http://www.firebirdnews.org/docs/fb2min_ptbr.html>. Acesso em: 27 fev. 2013.

CASESTUDIO. **CaseStudio**. Disponível em: <<http://www.casestudio.com>>. Acesso em: 12 mar. 2012.

EMABARCADERO. **Delphi XE3**. Disponível em: <<http://www.embarcadero.com/br/products/delphi>>. Acesso em: 17 set. 2012.

FACUNTE, Emerson. **Delphi Internet e Banco de Dados 7**. Rio de Janeiro: Brasport, 2003.

FAST REPORT. **Fast Report**. Disponível em: <http://www.fast-report.com/en/product/fast-report-vcl/?pk_campaign=google&pk_kwd=delphivcl&gclid=CMmD_r-G2bUCFQk4nAodCzgApw>. Acesso em: 28 fev. 2013.

FROLIK V. **CASE Studio 2. User's manual**, 2006. Disponível em: <http://www.casestudio.com/download/doc/cs2_manual.pdf>. Acesso em: 17 mar. 2012.

IBEXPERT. **IBExpert developer studio**. Disponível em: <<http://www.ibexpert.com/>>. Acesso em: 14 nov. 2012.

LAMIN, Johathan. **Conhecendo o Delphi**. Disponível em: <http://www.oficinadanet.com.br/artigo/745/conhecendo_o_delphi>. Acesso em: 19 de nov. 2012.

LANUSSE, Andreano. **CodeGear™ acelera o desenvolvimento para web com a nova versão do Delphi para PHP**. Disponível em: <<http://edn.embarcadero.com/article/37949>>. Acesso em 20 de mar. 2013.

LIMA, Márcia. **Embarcadero lança Delphi 2010**. Disponível em: <<http://www.baguete.com.br/noticias/geral/11/09/2009/embarcadero-lanca-delphi-2010>>. Acesso em: 21 de nov. 2012.

MARTINS, Vidal. **Visão geral sobre Delphi**. Disponível em: <<http://www.batebyte.pr.gov.br/modules/conteudo/conteudo.php?conteudo=185>>. Acesso em: 18 de nov. 2012.

PICÃO, Marcos Elias. **Criando instaladores para Windows com o Inno Setup**. Disponível em: <<http://www.hardware.com.br/tutoriais/inno-setup/>>. Acesso em; 28 fev. 2013.

PRESSMAN, R. **Engenharia de software**. São Paulo: MacGraw-Hill, 2005.

SEBESTA, R. W. **Conceitos de linguagem de programação**, 5a. ed., Porto alegre: Bookman, 2003.

SONNINO, Bruno. **Kylix Delphi para Linux: guia prático de programação**. Makron Books Pearson, 2001.

VISUAL PARADIGM. **Visual Paradigm**. Disponível em: <<http://www.visual-paradigm.com/>>. Acesso em: 23 jan. 2012.