



FACULDADE DE PINDAMONHANGABA

Raquel Aparecida Marcondes de Souza

SISTEMA PARA CONTROLE DE VENDAS E ESTOQUE

**Pindamonhangaba - SP
2009**



Raquel Aparecida Marcondes de Souza

SISTEMA PARA CONTROLE DE VENDAS E ESTOQUE

Monografia apresentada como parte dos requisitos para obtenção do Diploma de Bacharel em Sistemas de Informação pelo Curso de Bacharelado em Sistemas da Informação da Faculdade de Pindamonhangaba.

Prof^a. MSc. Alindacir Maria Dalla Vecchia Grassi

Prof^a. Bárbara Alessandra Gonçalves Pinheiro Yamada

**Pindamonhangaba - SP
2009**



RAQUEL APARECIDA MARCONDES DE SOUZA
SISTEMA PARA CONTROLE DE VENDAS E ESTOQUE

Monografia apresentada como parte dos requisitos para obtenção do Diploma de Bacharel em Sistemas de Informação pelo Curso de Bacharelado em Sistemas da Informação da Faculdade de Pindamonhangaba.

Data: _____

Resultado: _____

BANCA EXAMINADORA

Prof. _____ Faculdade de Pindamonhangaba

Assinatura _____

Prof. _____ Faculdade de Pindamonhangaba

Assinatura _____

Prof. _____

Assinatura _____

AGRADECIMENTOS

Agradeço primeiramente a Deus pela oportunidade de conseguir uma bolsa para poder realizar meus estudos e por todas as dificuldades que Ele foi me ajudando a superar nestes quatro anos.

Ao presidente Luís Inácio Lula da Silva e sua equipe de governo pelo projeto ProUni, que me permitiu a concessão de uma bolsa integral nesta faculdade.

Às minhas professoras Alindacir Maria Dalla Vecchia Grassi e Bárbara Alessandra Gonçalves Pinheiro Yamada que não só me orientaram com muita dedicação, como também me deram forças para continuar nos momentos de maior desânimo.

Ao meu professor Fabiano Sabha que, apesar do pouco tempo que nos deu aula, demonstrou ser muito responsável e íntegro, sempre prestativo e atencioso.

Às minhas companheiras de estágio Juliete Gomes de Amorim e Mariana Rosa Silva que muito colaboraram no desenvolvimento do protótipo do sistema aqui apresentado, seja na modelagem ou na própria linguagem de programação, bem como nas dificuldades que foram surgindo ao longo do desenvolvimento.

RESUMO

Atualmente, a maior parte das empresas comerciais, desde as micro empresas até as grandes redes, possuem um sistema de *software* para gerenciar todas as suas atividades. Neste sentido, este trabalho apresenta um protótipo para gerenciamento de vendas e estoque de uma pequena empresa comercial, sem fins lucrativos: o Bazar do Sagrado Coração de Jesus. A inexistência de um sistema informatizado para esse gerenciamento pode gerar muitas inconsistências e redundância de dados, além de atrasos para se obter informações importantes para a tomada de decisões e até mesmo a falência de empresas. O objetivo é informatizar o gerenciamento das principais atividades de uma empresa comercial, provendo com as informações armazenadas, condições para um controle financeiro e de estoque mais ágil, preciso e verídico. O sistema abrange funcionalidades como Gestão de Acesso, Gestão de Cadastro, Gestão de Movimentação Financeira e de Estoque e Gestão de Relatórios, as quais foram modeladas nos diagramas de Fluxo de Dados e Entidade-Relacionamento, disponibilizados na ferramenta *Case Studio*. O sistema foi implementado na linguagem *Object Pascal* por meio do IDE Delphi 7, com programação estruturada.

Palavras-chave: Agilidade. Estoque. Gerenciamento. *Software*. Vendas.

System to manage sales and stock

ABSTRACT

Nowadays, almost all commercial companies from small business to large ones, have a software system to manage all their tasks. In this sense, the present work aims in demonstrating a prototype to companies for managing their sales and stocks for a non-profit company called The Bazar do Sagrado Coração de Jesus. The lack of a computerized system for management can generate many inconsistencies, redundance data, and delays for important information of making decisions and even going out of business. The objective is to automate the management of the main activities in this company, providing it with the information stored, a financial control and faster, accurate and reliable stock. The system includes features such as Access Management, Management of Registration, Financial Transactions Management, Stock Reports Management. It was modeled in Data Flow and Entity-Relationship Diagrams, available in Case Studio tool. The system has been implemented in Object Pascal using the Delphi 7 IDE with structured programming.

Keywords: Agility. Stock. Management. Software. Sales.

LISTA DE TABELAS

Tabela 5.1 – Cartões de história resumidos.....	38
Tabela 5.2 – Cartões de história com prioridades, estimativa e interações.....	39
Tabela 5.3 – Requisitos Funcionais.....	41
Tabela 5.4 – Requisitos não Funcionais	44

LISTA DE FIGURAS

Figura 3.1- Aplicação estruturada em Delphi.....	17
Figura 3.2 - Tela inicial do Delphi versão 7.....	18
Figura 3.3 - Tela inicial do <i>Microsoft Office Access 2003</i>	21
Figura 3.4 - Tela inicial do <i>BDE Administrator</i>	22
Figura 3.5- <i>Case Studio 2</i> com projeto visualizado em módulos	25
Figura 4.1 – Modelo de Cartão de História no XP	29
Figura 4.2 – Modelo de Cartão de Tarefa no XP.....	30
Figura 4.3 – Ciclo de vida nos projetos XP	32
Figura 5.1 – Diagrama de Fluxo de Dados do Sistema SCVE-BSCJ.....	45
Figura 5.2 – Decomposição do Processo: Processar Vendas	46
Figura 5.3 – Decomposição do Processo: Gerenciar Usuários	46
Figura 5.4 – Diagrama Entidade-Relacionamento do Sistema SCVE-BSCJ.....	47
Figura 5.5 – Esquema Relacional do Sistema SCVE-BSCJ	48
Figura 6.1 – Tela de Abertura do SCVE-BSCJ	51
Figura 6.2 – Alterar Senha.....	51
Figura 6.3 – Menu Principal.....	52
Figura 6.4 – Cadastro de Artesãos	53
Figura 6.5 – Cadastro de Produtos	53
Figura 6.6 – Cadastro de Usuários.....	54
Figura 6.7 – Lista de Artesãos	54
Figura 6.8 – Lista de Produtos	55
Figura 6.9 – Lista de Usuários.....	55
Figura 6.10 – Formulário de Despesas do Bazar	56
Figura 6.11 – Formulário de Receitas do Bazar	56
Figura 6.12 – Formulário de Fechamento do Caixa Diário.....	57
Figura 6.13 – Formulário de Pagamento ao Artesão	57
Figura 6.14 – Recibo de Pagamento ao Artesão	58
Figura 6.15 – Formulário de Vendas	58
Figura 6.16 – Formulário de Itens de Venda	59
Figura 6.17 – Formulário de Entrada de Produtos	59
Figura 6.18 – Recibo de Entrega de Produtos.....	60
Figura 6.19 – Formulário de Devolução de Produtos.....	60
Figura 6.20 – Recibo de Devolução de Produtos.....	61
Figura 6.21 – Relatório de Produtos com tempo de permanência vencido	61
Figura 6.22 – Relatório de Produtos em Estoque	62
Figura 6.23 – Relatório de Caixa Diário.....	62
Figura 6.24 – Resumo Mensal de Fluxo de Caixa.....	63
Figura 6.25 – Relatório de Repasse ao Bazar.....	63
Figura 6.26 – Formulário de <i>Backup</i>	64
Figura 6.27 – Capa do Manual de Orientação ao Usuário	64
Figura 6.28 – Formulário de Informações do Sistema.....	65

SUMÁRIO

1 INTRODUÇÃO	9
1.1 Objetivo do Trabalho	10
1.2 Etapas desenvolvidas	11
1.3 Estrutura do Trabalho	11
2 CONTEXTUALIZAÇÃO DO SISTEMA	13
2.1 Descrição do modelo atual	13
2.2 Proposta	14
3 FERRAMENTAS UTILIZADAS	16
3.1 Delphi	16
3.2 Access	19
3.3 BDE Administrator	21
3.4 Case Studio	23
4 MODELO DE PROCESSO DE DESENVOLVIMENTO DE SOFTWARE UTILIZADO	26
4.1 Definição de Modelo de Processo de Desenvolvimento de Software	26
4.2 Extreme Programming (XP)	27
4.2.1 Mecanismo de Desenvolvimento no XP	32
5 MODELAGEM DO SISTEMA SCVE-BSCJ	36
5.1 Visão do Sistema	36
5.1.1 Composição e tarefas da equipe	37
5.1.2 Escopo do Produto	37
5.1.3 Histórias do usuário	37
5.1.4 Estimativa, priorização e planejamento	38
5.1.5 Funções do Produto	40
5.1.6 Testes de Aceitação	48
5.1.7 Fim do projeto	49
6 PROTÓTIPO DO SISTEMA	50
6.1 Apresentação do Protótipo do Sistema	50
7 CONCLUSÕES E CONSIDERAÇÕES FINAIS	66
REFERÊNCIAS BIBLIOGRÁFICAS	68
REFERÊNCIAS BIBLIOGRÁFICAS COMPLEMENTARES	71

1 INTRODUÇÃO

Nas décadas de 80 e 90, várias manchetes de revistas e jornais, como por exemplo: “*Software: A Nova Força Propulsora*”, “*Armadilha do Software – Automatizar ou Não?*” e “*Criar Software Novo: Era Uma Tarefa Agonizante...*”, demonstravam a nova compreensão da importância do *software* de computador – suas oportunidades e os perigos que apresentavam (PRESSMAN, 1995, p. 3). “O grande desafio da década de 90 é melhorar a qualidade (e reduzir o custo) de soluções baseadas em computador – soluções que são implementadas com *software*” (PRESSMAN, 1995, p. 4). Essa preocupação com a qualidade e custo do *software* ainda persiste nos dias atuais, surgindo cada vez mais técnicas para auxiliar o desenvolvimento do *software*, constituindo uma disciplina chamada Engenharia de *Software*, que utiliza técnicas de engenharia, como teorias, métodos e ferramentas, para produzir um *software*, desde os estágios iniciais de especificação do sistema até a manutenção desse sistema, trabalhando sempre de acordo com as restrições organizacionais e financeiras (SOMMERVILLE, 2003).

Os sistemas de *software* são utilizados em todos os lugares e setores, desde equipamentos elétricos, operações da indústria manufatureira, escolas e universidades, setor de assistência à saúde, finanças e governo, ou até mesmo como entretenimento.

Segundo Pressman (1995, p. 20), “o processamento de informações comerciais é a maior área particular de aplicação de *software*”. Sistemas distintos, como por exemplo, folha de pagamento, contas a pagar e a receber, controle de estoque, dentre outros, se integraram, formando *software* de sistema de informações administrativas, que acessam um ou mais banco de dados que contém informações comerciais, facilitando as operações comerciais e as tomadas de decisões administrativas.

Portanto, torna-se evidente as vantagens de sistemas informatizados para o controle de gerenciamento de qualquer atividade comercial, como por exemplo, redução de custos, acesso rápido às informações, garantia de integridade e veracidade da informação, além de garantia de segurança de acesso à informação e eliminação da redundância de dados (SOMMERVILLE, 2003).

O descontrole do estoque gera muitos prejuízos para as empresas de atividade comercial, visto que as mesmas podem perder oportunidades de venda por não terem acesso rápido a dados do seu estoque. Além disso, a falta de um rigoroso controle financeiro pode levar a falência da empresa.

A falta de gerência das informações no Bazar do Sagrado Coração de Jesus gera vários problemas, como: descontrole de estoque, dos usuários do sistema, dos artesãos e de caixa.

O controle de mercadorias em estoque é demorado e impreciso, não havendo controle de tempo de permanência das mercadorias no bazar, culminando no acúmulo de estoque por mercadorias “encalhadas” e não há um controle financeiro do bazar, gerando atraso no pagamento dos artesãos, falta de informação de quem vendeu o quê e inexistência de informações financeiras para tomada de decisões, como por exemplo, compra de algum item necessário ao bazar.

Um sistema informatizado proporcionará ao bazar maior agilidade e precisão no controle de estoque, controle total de entrada e saída de dinheiro, de usuários do sistema e das mercadorias vendidas pelos mesmos, dos artesãos e das mercadorias entregues por eles e um estoque mais selecionado (controle de tempo de permanência das mercadorias no bazar).

Torna-se então necessário o desenvolvimento de um sistema que armazene de forma confiável, informações dos produtos, usuários e artesãos do Bazar do Sagrado Coração de Jesus, fornecendo, sempre que necessário, relatórios financeiros e de estoque.

1.1 Objetivo do Trabalho

Projetar e implementar um sistema para controle de vendas e estoque do Bazar do Sagrado Coração de Jesus – SCVE-BSCJ, o qual gerenciará todo o funcionamento do bazar para obter informações detalhadas, de forma rápida e segura, sobre cada produto, usuário e artesão, além de quantidade de produtos em estoque, tempo de permanência dos mesmos, valores devidos aos artesãos e valor em caixa.

Os fatos apresentados neste trabalho são reais, porém os nomes são fictícios para preservar a Entidade.

1.2 Etapas desenvolvidas

A primeira etapa desenvolvida foi o levantamento do sistema utilizado pelo Bazar do Sagrado Coração de Jesus, dos problemas encontrados no mesmo e a identificação das necessidades do cliente, obtidos por meio de informações coletadas em entrevista e reuniões com o cliente.

Definiu-se então, os objetivos e escopo do projeto, que permitiu a escolha do modelo de processo de desenvolvimento de *software* que melhor se aplicasse ao SCVE-BSCJ. Para isso foram estudados e analisados alguns modelos.

Em seguida, foi utilizada a Análise Estruturada de Sistema para modelar os requisitos do usuário, por meio do DFD - Diagrama de Fluxo de Dados e DER – Diagrama Entidade-Relacionamento.

Iniciou-se, então, a programação do protótipo do Sistema SCVE-BSCJ, com aplicação de testes a cada nova funcionalidade implementada. Somente se passava à implementação de uma nova funcionalidade, após todos os testes terem sido efetuados com sucesso.

Em paralelo a todas estas atividades, foi redigida a presente monografia.

1.3 Estrutura do Trabalho

Este trabalho está dividido em 7 Capítulos, a saber:

- **Capítulo 1 - Introdução:** Apresenta uma visão geral de todo o trabalho;
- **Capítulo 2 - Contextualização do Sistema:** Expõe o sistema atual de gerenciamento do Bazar do Sagrado Coração de Jesus e a proposta deste trabalho;

- **Capítulo 3 - Ferramentas Utilizadas:** Explica as ferramentas utilizadas no desenvolvimento do trabalho;
- **Capítulo 4 - Modelo de Processo de Desenvolvimento de *Software*:** Apresenta, detalhadamente, o ciclo de vida adotado no desenvolvimento do Sistema SCVE-BSCJ;
- **Capítulo 5 - Modelagem do Sistema SCVE-BSCJ:** Descreve o estudo de caso realizado;
- **Capítulo 6 - Protótipo do Sistema:** Apresenta o protótipo do Sistema SCVE-BSCJ;
- **Capítulo 7 - Conclusões e Considerações Finais:** Conclusão do Trabalho.

2 CONTEXTUALIZAÇÃO DO SISTEMA

Para uma melhor compreensão da proposta deste trabalho, neste capítulo é apresentado o modelo atual de controle de vendas e estoque, utilizado pelo Bazar do Sagrado Coração de Jesus.

2.1 Descrição do modelo atual

O Bazar do Sagrado Coração de Jesus é um comércio sem fins lucrativos. Seu principal objetivo é a comercialização de peças confeccionadas por artesãos e integrantes de projetos sociais da cidade de Jesus de Nazaré, com o intuito de auxiliá-los na divulgação e venda de seus trabalhos. Os artesãos cadastrados em projetos sociais do município recebem integralmente o valor oriundo de suas vendas, enquanto que os demais artesãos reverterem 10% do valor de suas vendas para a manutenção do bazar.

Os funcionários do bazar são voluntários, ou seja, não recebem nenhuma remuneração por seus serviços e, geralmente, trabalham se revezando no horário da manhã e da tarde.

Há uma grande diversidade de produtos, com variados preços, porque cada artesão define o valor de suas mercadorias, podendo haver o mesmo tipo de produto oriundo do mesmo artesão, mas com preços diferentes. Para este controle são confeccionadas etiquetas manualmente, com identificação da mercadoria e do artesão e o preço do produto.

Quando o artesão deixa seus produtos é preenchido um recibo em uma planilha do Excel, com a discriminação das mercadorias, quantidades e valores das mesmas. O recibo é emitido em duas vias, que são assinadas pelo artesão e pelo funcionário receptor. O artesão recebe uma via e a outra fica no bazar para controle do mesmo.

Ao vender um produto, o funcionário preenche uma outra planilha para acerto financeiro do artesão. Esta planilha é separada por artesão, sendo uma para cada

artesão. Ao encerrar o mês, é feito o fechamento das planilhas de acerto financeiro, para se obter o valor devido a cada artesão e o valor de repasse ao bazar. Estas planilhas servem de recibo de pagamento de artesãos e também são emitidas em duas vias, as quais são assinadas e datadas pelos artesãos no momento do acerto.

O bazar possui uma ficha de cadastro de cada artesão com seus dados pessoais, bem como o registro no Departamento de Cultura do município, denominado PAT.SUTACO. Esta ficha é preenchida manualmente e arquivada em uma gaveta, em ordem alfabética.

Quando há necessidade de saber a quantidade exata de determinado produto, é realizada a contagem manual do mesmo e quando chega um cliente perguntando se há determinado produto no bazar, o funcionário precisa procurar para descobrir.

Não é mantido nenhum cadastro de funcionários do bazar. Somente há o número do telefone dos mesmos anotados em um caderno.

2.2 Proposta

O Sistema para controle de vendas e estoque do Bazar do Sagrado Coração de Jesus – SCVE-BSCJ – visa informatizar todas as rotinas do bazar que, atualmente, são realizadas de forma manual, conforme exposto a seguir.

Será implementado um cadastro de todos os funcionários, denominados usuários do sistema, dos artesãos e das mercadorias deixadas pelos mesmos, denominadas produtos.

Para acessar o sistema será necessário que o usuário se identifique por meio de seu *login* e senha. Sem a validação do usuário não será possível o acesso ao mesmo. Outra funcionalidade a ser permitida é a alteração de senha, desde que haja primeiro a validação do usuário.

A entrada de mercadorias será feita através de um formulário de entrada de produtos, onde o usuário preencherá todos os dados das mercadorias deixadas e, para cada mercadoria, um código será gerado automaticamente pelo sistema. Este código servirá para identificação da mesma. Outro requisito é a identificação do

usuário que realizou a entrada do produto, pois em caso de dúvida, haverá como esclarecê-la.

Ao cadastrar a entrada do produto, o estoque será automaticamente atualizado. Ao final, será possível imprimir o recibo de entrega de mercadorias e as etiquetas dos produtos.

As vendas serão cadastradas no sistema por meio de um formulário de vendas, dando baixa automaticamente no produto e buscando o artesão que o forneceu para cálculo de seu pagamento e do repasse ao bazar.

Por meio do menu principal, poderão ser emitidos relatórios financeiros diários ou mensais.

O estoque também poderá ser consultado a qualquer momento, assim como o valor devido aos artesãos e o valor de repasse ao bazar. Caso seja necessário, poderão ser emitidos relatórios de acompanhamento, como por exemplo, relatórios de valor de repasse ao bazar ou aos artesãos, por períodos determinados pelo usuário.

O sistema proverá, ainda, um controle de tempo de permanência de produtos no estoque. No final de cada mês será possível imprimir a relação de produtos a serem devolvidos; neste caso, o sistema calculará e imprimirá a relação dos produtos que estão, há mais de três meses, no bazar.

O Capítulo 5 apresenta o Documento de Requisitos e Modelagem propostos para o sistema SCVE-BSCJ.

3 FERRAMENTAS UTILIZADAS

Neste Capítulo serão apresentadas as ferramentas utilizadas para desenvolver o Sistema SCVE-BSCJ.

3.1 Delphi

Delphi não é uma linguagem de programação, mas um ambiente de desenvolvimento integrado (MANZANO, 2006).

“O ambiente de programação Delphi é baseado na linguagem de programação *Object Pascal*, oriunda da linguagem *Pascal*, a qual foi projetada pelo Professor Niklaus Wirth” (MANZANO, 2006, p. 19).

O Delphi é um *Integrated Development Environment* (IDE) ou Ambiente de Desenvolvimento Integrado (MANZANO, 2006), ou seja, um ambiente de suporte que permite a programação visual: as telas podem ser montadas facilmente, simplesmente com o clicar e arrastar de componentes aos formulários, sendo possível compilar o projeto na linguagem *Object Pascal* e gerar o protótipo desejado.

Segundo Silva; Paula (2007, p. 20), “o IDE do Delphi possui uma estrutura de fácil compreensão, possibilitando que até mesmo os programadores iniciantes desenvolvam seus projetos sem muitas dificuldades”.

Este IDE foi criado pela *Borland* como uma ferramenta de programação *Rapid Application Development* (RAD), isto é, Desenvolvimento Rápido de Aplicações.

É muito versátil, podendo ser aplicado para pequenas, médias ou grandes aplicações, suportando criação de *softwares* multi-plataforma (*Windows* e *Linux*), sistemas multicamadas, aplicações para a Internet e para o ambiente cliente-servidor e tem propósito geral, permitindo desenvolvimento de aplicações tanto científicas como comerciais. Por todos esses motivos, “muitas empresas em diversos segmentos do mercado utilizam o Delphi como ferramenta de soluções de desenvolvimento de sistemas corporativos” (SILVA; PAULA, 2007, p. 13).

De acordo com Silva; Paula (2007, p. 19), o Delphi 7 possui várias edições com características particulares, conforme segue:

Personal – É uma edição para programadores eventuais e não possui suporte nem para banco de dados nem para outras características avançadas do Delphi.

Professional Studio – Esta edição é para desenvolvedores profissionais e possui todas as ferramentas básicas para trabalhar com banco de dados, programação para *web* e algumas das ferramentas externas, como *ModelMaker* e *IntraWeb*.

Enterprise Studio – É uma edição para o desenvolvimento de aplicativos corporativos. Para isso, conta com ferramentas para XML avançadas, tecnologias para *Web*, e muitas outras ferramentas.

Architect Studio – Esta edição amplia o suporte da edição *Enterprise* ao *Bold*, um ambiente para construir aplicações com modelo UML e capacidade de mapear seus objetos tanto na base de dados quanto na interface com o usuário.

Além de possuir várias edições, o ambiente do Delphi pode ser personalizado de acordo com as necessidades do desenvolvedor.

A estrutura de uma aplicação Delphi é muito simples. Conforme Garcia (2005, p. 16), “uma aplicação desenvolvida em Delphi consiste em um arquivo de projeto, composto por um ou mais *forms* (formulários) e *units*”. Os *forms* são elementos visuais para a criação de uma interface entre a aplicação e o usuário e as *units* são arquivos de programa-fonte. Todo formulário, obrigatoriamente, tem uma *unit* correspondente, porém nem toda *unit* corresponde a um formulário. Existem aplicações que possuem apenas *units*, denominadas *console application*. Um exemplo da estrutura de uma aplicação Delphi é apresentado na Figura 3.1.

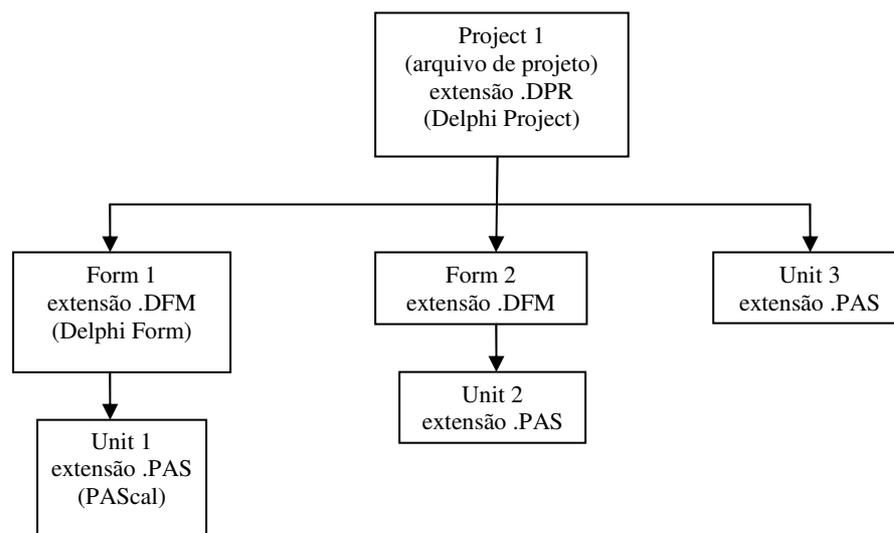
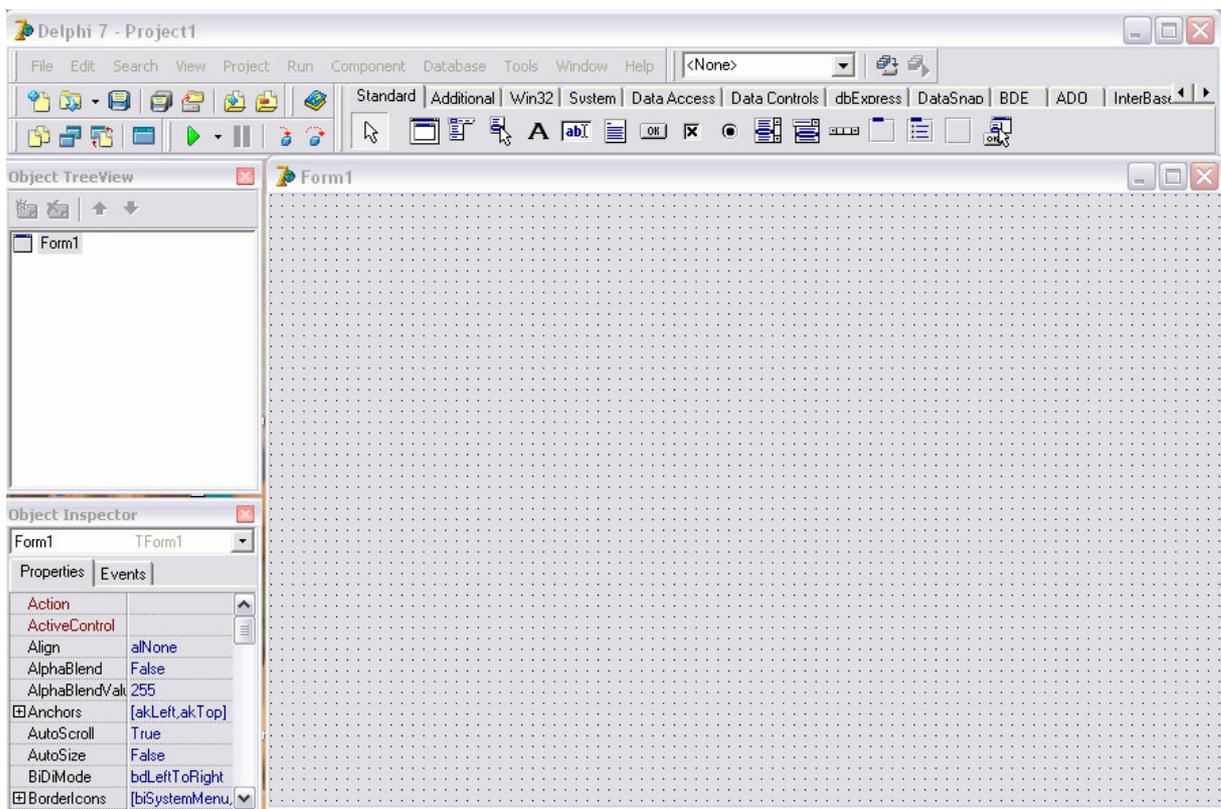


Figura 3.1- Aplicação estruturada em Delphi
GARCIA (2005, p. 19)

O Delphi possui dois níveis de programação: nível de *design* em que utiliza recursos de programação visual e aproveita componentes prontos e o nível de *component writer* em que o desenvolvedor elabora os componentes a serem utilizados pelos programadores que trabalham no nível *design* (GARCIA, 2005).

Em 2002, a *Borland* lançou a versão 7 do IDE Delphi, com novos recursos voltados para aplicações para Internet como a *Intraweb*, e *Rave Reports* para a criação de relatórios, entre outros (MANZANO, 2006). A Figura 3.2 apresenta a tela inicial do Delphi versão 7.

Conforme os componentes vão sendo selecionados, o Delphi escreve o código fonte. Os componentes, em geral, incluem classes e propriedades muito utilizadas, que se relacionam com outros objetos (GARCIA, 2005). Além disso, “o Delphi possui um amplo suporte para trabalho com banco de dados Paradox, dBase, Access, MySQL, Oracle, Firebird, entre outros” (SILVA; PAULA, 2007, p. 83).



**Figura 3.2 - Tela inicial do Delphi versão 7
BORLAND (2002)**

3.2 Access

“Nem todos os sistemas baseados em computador fazem uso de um banco de dados, mas, para todos que o fazem, essa modalidade de armazenamento de informações freqüentemente é de grande importância para a função global” (PRESSMAN, 1995, p. 197). Por esse motivo, assim que o domínio da informação é definido, é aplicada uma disciplina técnica denominada engenharia de banco de dados (*database engineering*), que abrange a análise, projeto e criação de banco de dados. O engenheiro de sistemas deve “definir as informações a serem contidas no banco de dados, os tipos de *queries* a serem submetidos a processamento, a maneira pela qual os dados serão acessados e a capacidade do banco de dados” (PRESSMAN, 1995, p. 197).

De acordo com Silva; Paula (2007, p. 83), “a utilização de banco de dados é essencial para o desenvolvimento de aplicativos comerciais”. As principais vantagens do banco de dados são redução ou eliminação de redundâncias, eliminação de inconsistências, rapidez e eficiência na recuperação e manipulação de dados, compartilhamento de dados, restrições de segurança, padronização e independência dos dados e manutenção de integridade (DATE, 2004).

Por ser um projeto pequeno, que exige rapidez de desenvolvimento, a implementação do Banco de Dados (DATE, 2004) será feita no *MSAccess - Microsoft Office Access 2003* (MICROSOFT, 2009), que permite o desenvolvimento rápido de aplicações que envolvem a modelagem e estrutura de dados. Apesar de ser um banco de dados proprietário, ou seja, pago, sua principal vantagem consiste no fato do desenvolvimento da estrutura de dados ser de forma intuitiva, não necessitando que o desenvolvedor tenha conhecimentos avançados em modelagem de dados e lógica de programação.

O *MSAccess* é um sistema relacional de administração de banco de dados da *Microsoft*, incluído no pacote do *Microsoft Office Professional*, que disponibiliza a linguagem de programação *Microsoft Visual Basic for Application*.

No *MSAccess*, os bancos de dados consistem em quatro objetos principais:

a) Tabelas: responsável por armazenar dados em linhas e colunas. Cada assunto principal deve constituir uma tabela individual, portanto os bancos de dados podem conter uma ou mais tabelas;

b) Consultas: recuperam e processam dados, podendo combinar dados de diferentes tabelas, atualizar dados e executar cálculos com base nesses dados;

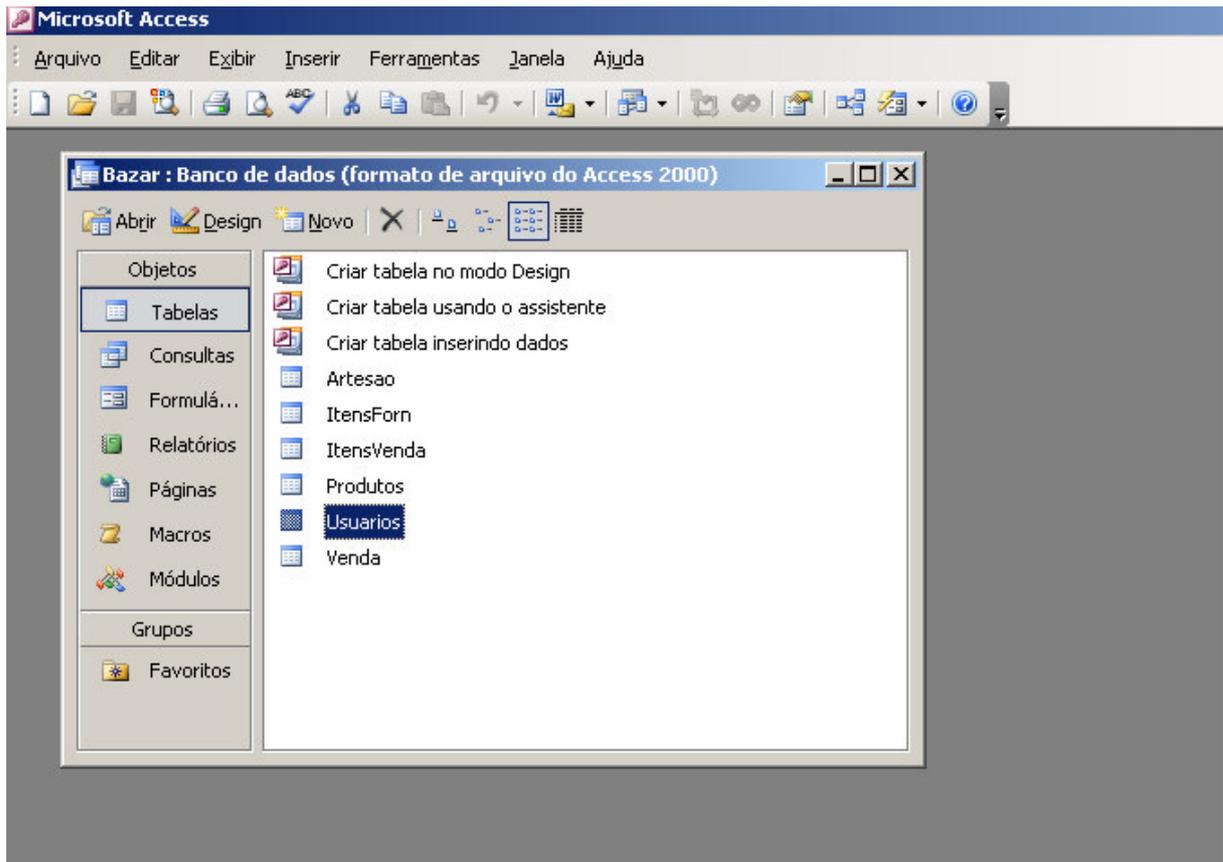
c) Formulários: controlam a entrada e as exibições de dados, fornecendo uma interface amigável como se fosse um formulário de papel, mas com utilização de *mouse* e teclado para entrada de dados. Conforme os dados são inseridos nos formulários, os mesmos são salvos na devida tabela;

d) Relatórios: fazem o resumo e a impressão de dados, transformando os dados de tabelas e consultas em documentos destinados à comunicação de idéias. Os formatos dos relatórios podem ser salvos para que tenham sempre a mesma aparência, mesmo que os dados sejam alterados.

Para manipular as informações armazenadas no *MSAccess* e também no IDE Delphi é utilizada a linguagem SQL (*Structured Query Language*) – ou, Linguagem de Consulta Estruturada, que se estabeleceu como linguagem padrão de banco de dados e utiliza vários comandos para manipular estas informações, como o comando *Create* para criação do banco de dados e das tabelas, o comando *Open* para abrir o banco de dados ou as tabelas antes do início das atividades com os mesmos, o comando *Insert* para inserir novos registros na tabela, o comando *Delete* para excluir registro de uma tabela, o comando *Update* para alterar valores das colunas de uma tabela e o comando *Select* que seleciona um grupo de registros de uma ou mais tabelas, sendo um dos principais recursos desta linguagem (SILVA; PAULA, 2007).

Os bancos de dados criados pelo *MSAccess* são relacionais, ou seja, os dados armazenados em várias tabelas separadas de acordo com o assunto ou a tarefa estão relacionados, podendo ser reunidos da maneira que for especificado (MICROSOFT, 2009).

A Figura 3.3 ilustra a tela inicial do *MSAccess* 2003.



**Figura 3.3 - Tela inicial do *Microsoft Office Access 2003*
MICROSOFT (2003)**

3.3 BDE Administrator

O BDE (*Borland Database Engine*) é um mecanismo fornecido pelo Delphi para acessar banco de dados locais por meio de *drivers* ODBC (CANTU, 2006). Foi construído com o objetivo de unificar a conectividade de banco de dados, através da abstração de toda a funcionalidade de um banco de dados dentro de um *engine*, facilitando, por exemplo, a atualização do acesso a banco de dados, pois não é necessária a atualização de um pacote de *software* inteiro. Além disso, economiza espaço em disco e garante que o código de acesso ao banco de dados seja escrito apenas uma vez (UNITRI, 2009).

Com o recurso do BDE não é necessário conhecimento prévio para desenvolvimento de aplicativos de bancos de dados em Delphi e nem os usuários

destes aplicativos precisam ter esse conhecimento. A instalação e a execução deste recurso são de forma automática no Delphi (SILVA; PAULA, 2007).

Os *drivers* padrões do BDE *Administrator* são Paradox, dBASE, FoxPro e ASCII *text*, além de incluir também um *driver* para o *Microsoft Access*. Outros *drivers* são instalados separadamente. Para abrir ou criar tabelas do *Microsoft Access* basta utilizar o *driver MSAccess* do BDE *Administrator* (BDE, 1998).

“Para facilitar o trabalho com banco de dados o BDE utiliza o conceito de *alias* (apelido)” (SILVA; PAULA, 2007, p. 103), que funciona como um apontador para o caminho especificado, conectando os componentes do Delphi com as tabelas físicas do banco de dados e permitindo, inclusive, “que as tabelas possam ser colocadas em outro diretório, sem que seja necessário alterar o seu caminho em todos os componentes da aplicação” (SILVA; PAULA, 2007, p. 103), bastando que o caminho seja alterado no BDE *Administrator*. A Figura 3.4 apresenta a tela inicial do BDE *Administrator*.



**Figura 3.4 - Tela inicial do BDE *Administrator*
BDE (1998)**

3.4 Case Studio

CASE significa *computer-aided software engineering*, ou seja, engenharia de *software* com o auxílio de computador (SOMMERVILLE, 2003). Ferramentas *CASE* são vários programas utilizados para apoiar as atividades de processo de *software*, desde a fase inicial (quando pode ser chamada de ferramenta *Upper-CASE*), como análise de requisitos, até a fase final (quando pode ser chamada de ferramenta *Lower-CASE*), como depuração e testes. São freqüentemente utilizados para proporcionar apoio aos métodos. Podem também incluir um gerador de códigos que, automaticamente, origina código-fonte a partir do modelo de sistema e alguma orientação de processo, que fornece conselhos ao engenheiro de *software* sobre o que fazer em seguida (SOMMERVILLE, 2003).

Pressman (1995) deixa claro que o verdadeiro poder do *CASE* só pode ser obtido quando as várias ferramentas são integradas. Entre os vários benefícios incluem-se transferência harmoniosa de informações de uma ferramenta para outra e de uma etapa da engenharia de *software* para a seguinte, aumento no controle do projeto, obtido por meio de um melhor planejamento, monitoração e comunicação e coordenação melhorada entre os membros de uma equipe que esteja trabalhando num grande projeto de *software* (PRESSMAN, 1995). Para essa integração, faz-se uso do repositório *CASE*, que é um banco de dados que armazena todas as informações de engenharia de *software*. Além das funções óbvias de um sistema de gerenciamento de banco de dados, o repositório também realiza a integridade de dados (valida entradas, garante consistência entre objetos relacionados e executa automaticamente modificações “em cascata”), compartilhamento de informações (entre múltiplos desenvolvedores e entre múltiplas ferramentas), integração dados-ferramenta, integração dados-dados, imposição metodológica (através de um paradigma específico para a engenharia de *software*) e padronização de documentos (criando definições para os objetos do banco de dados).

Existem diversas ferramentas *CASE* a disposição, cada uma abordando aspectos diferentes de desenvolvimento de *software*, por exemplo, diagramas propostos pela Análise Estruturada ou Orientada a Objetos, Requisitos, Modelo Entidade Relacionamento, dentre outros. Algumas são comercializadas e outras não.

Exemplos de ferramentas são *Case Studio*, *Case Design Studio*, *IBDataWorks*, *argoUML*, *Visual-Paradigm*, *Dia*, *Jude Community*, *ErWin*, dentre outros.

Neste trabalho será utilizada a ferramenta *Case Studio* (CASE, 2006) por possuir todos os diagramas necessários ao desenvolvimento de *software* proposto pela Análise Estruturada de Sistemas, ser uma ferramenta leve, rápida, de fácil utilização e muito didática.

Uma característica muito importante desta ferramenta é a engenharia reversa, com geração de *scripts* com muitas opções de configuração, entre elas *scripts* para tabelas, chaves primárias, índices, *triggers* e restrições referenciais, o que facilita muito quando não há documentação do sistema pronto.

O *Case Studio 2* trabalha somente com Análise Estruturada de Sistemas e disponibiliza os seguintes diagramas: Diagrama Entidade-Relacionamento (DER) e Diagrama de Fluxo de Dados (DFD), que são elaborados de forma visual e interativa. O projeto pode ser visualizado em módulos ou de maneira global e também podem ser visualizadas as entidades físicas ou lógicas, todos os atributos ou somente as chaves.

Case Studio é uma ferramenta CASE compatível com vários bancos de dados: *Access 2000*, *Access 97*, *Advantage 7*, *Clipper 5*, *DBIsam 3.23*, *DB2 UDB ver. 8.1*, *DB2 UDB ver. 7*, *Firebird 1.5*, *Informix 9*, *Informix*, *Ingres*, *InterBase 7*, *InterBase 6 SQL 3*, *InterBase 6 SQL 1*, *InterBase 5*, *InterBase 4*, *MaxDB*, *MS SQL 2000*, *MS SQL 7*, *MS SQL 6.5*, *MySQL 4.0*, *MySQL 3.23*, *Oracle 10g*, *Oracle 9i*, *Oracle 8*, *Paradox*, *Pervasive V8*, *PostgreSQL 7.4*, *PostgreSQL 7.3*, *PostgreSQL 7*, *Sybase Anywhere*, *Sybase ASE 12.5*, *Sybase ASE 12.5.1*.

Os principais recursos da ferramenta *Case Studio 2* são: implementação automática dos principais relacionamentos, bom controle de usuário e segurança, podendo ser feito controle, até mesmo, de qual operação determinado usuário pode fazer em determinada entidade e, controle de versionamento, com comparativo entre as diferentes versões.

Os documentos principais de análise são gerados em dois formatos: html e rtf.

A Figura 3.5 mostra um projeto visualizado em módulos no *Case Studio 2*.

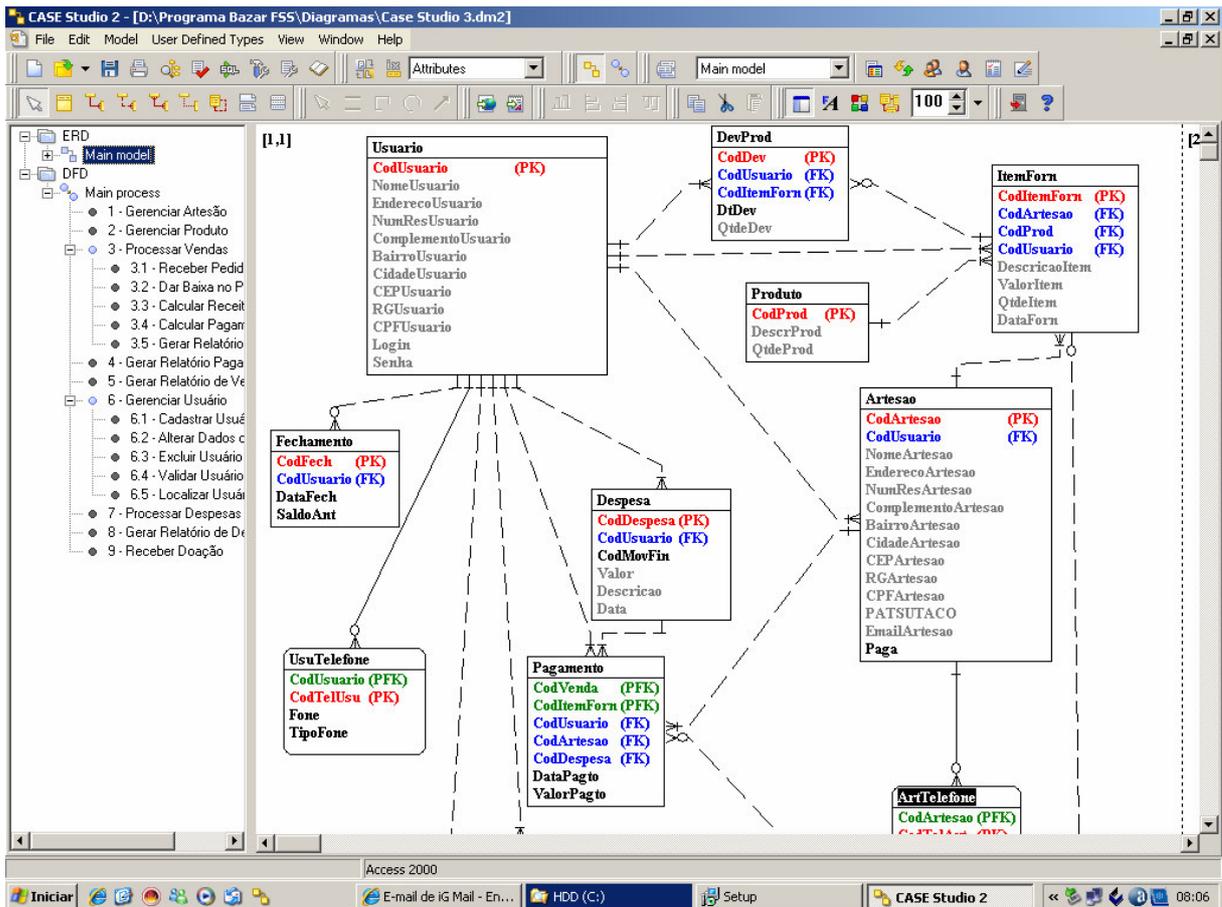


Figura 3.5- Case Studio 2 com projeto visualizado em módulos CASE (2006)

4 MODELO DE PROCESSO DE DESENVOLVIMENTO DE SOFTWARE UTILIZADO

O objetivo deste capítulo é definir Modelo de Processo de Desenvolvimento de *Software* e apresentar o modelo *Extreme Programming (XP)* adotado para o desenvolvimento do Sistema SCVE-BSCJ.

4.1 Definição de Modelo de Processo de Desenvolvimento de *Software*

Para que possa ser entendido o que é Modelo de Processo de Desenvolvimento de *Software* é necessário, primeiramente, ter uma visão bem definida do conceito de *Software*. Segundo Sommerville (2003), *software* não é apenas o programa, mas também toda a documentação associada, que descreve a estrutura do sistema de *software* e documentação do usuário, que explica como utilizar o sistema e os dados de configuração necessários para fazer com que esses programas operem corretamente.

Engenharia de *Software* é a disciplina que se ocupa de todos os aspectos da produção de *software*, desde os estágios iniciais de especificação do sistema até a manutenção desse sistema, depois que ele entrou em operação (SOMMERVILLE, 2003). A Engenharia de *Software* não se limita aos processos técnicos de desenvolvimento de *software*, mas envolve todo o gerenciamento de projetos de *software* e o desenvolvimento de ferramentas, métodos e teorias que dêem apoio à produção de *software*, com a finalidade de produzir *software* de alta qualidade, com uma boa relação custo-benefício.

De acordo com Sommerville (2003), Processo de Desenvolvimento de *Software* é um conjunto de atividades e resultados associados que geram um produto de *software*. É conhecido também como Ciclo de Vida de um *Software*.

Portanto, Modelo de Processo de Desenvolvimento de *Software* pode ser definido como uma sugestão de seqüência de atividades a ser seguida, como uma

receita, para a obtenção do sistema de *software*; fruto de estudos e experiências no desenvolvimento de *software*. Sua principal importância é auxiliar e conduzir o desenvolvedor nas etapas a serem seguidas para produzir um sistema de software de alta qualidade.

Há uma diversidade muito grande de Modelos de Processo de Desenvolvimento de *Software*. Na literatura são encontrados vários modelos como *Madaw* (NOGUEIRA; FERRETI, 2005), *Rational Unified Process* (RUP) (RATIONAL, 2009), Baseado em Usabilidade (MARTINEZ, 2003), Prototipação (PRESSMAN, 1995; SOMMERVILLE, 2003), *Extreme Programming* (XP) (BECK, 1999), dentre outros. A escolha de um ou de outro depende do domínio da aplicação, ou ainda, a empresa também pode optar por desenvolver um modelo próprio, de acordo com sua realidade.

Para o desenvolvimento do Sistema SCVE-BSCJ foi adotado o modelo de processo de desenvolvimento XP por ser um processo leve, centrado no desenvolvimento iterativo (entre desenvolvedor e cliente) e com a entrega constante de pequenas versões (*releases*) do *software*, atendendo perfeitamente às perspectivas do cliente (BECK, 1999). Na seção 4.2 é apresentado esse modelo em detalhes.

4.2 *Extreme Programming* (XP)

Introduzido por Kent Beck e Ward Cunningham por volta de 1996, é ideal para desenvolvimento rápido de projetos com requisitos vagos e freqüentes mudanças de escopo (BECK, 1999; CANTU, 2006).

Faz parte de uma família de processos de desenvolvimento de *software* denominada Metodologia *Agile*, que visa desenvolver *softwares* de qualidade, no menor tempo possível, atendendo as necessidades do cliente e respondendo com rapidez às mudanças nas especificações dos projetos (KUHN; PAMPLONA, 2009).

Prega a idéia de *releases* curtos, ou seja, o cliente recebe, assim que possível, pequenas versões para análise.

As versões devem ser incrementadas com a melhoria constante do código, denominada re-trabalho, mesmo que o código esteja funcionando perfeitamente.

Estas mudanças no código devem ser encaradas com naturalidade, visto que o XP assume que os requisitos do sistema mudam constantemente, sem que isso seja culpa do cliente.

Tem por base a presença do cliente junto aos desenvolvedores. Ele acaba por se tornar um membro da equipe de desenvolvimento, reavaliando a versão recebida e realimentando a equipe com suas principais necessidades e prioridades e recebendo da equipe informações como riscos, estimativas e alternativas de *design*. Com isso, é criado um elo de parceria e confiança mútua, dispensando muitos documentos formais e dando liberdade de negociação de atrasos ou outras necessidades, quando necessário.

O XP é norteado por quatro dimensões (BECK, 1999):

- **Simplicidade:** a equipe deve modelar e documentar apenas quando extremamente necessário e deve implementar da forma mais simples possível as necessidades do cliente para que ele possa aprender durante o projeto e consiga dar o *feedback* necessário. O desenvolvedor deve implementar apenas o necessário para atender o pedido do cliente, agilizando o processo e satisfazendo o cliente.
- **Comunicação:** os desenvolvedores e o cliente devem estar em constante comunicação.
- **Coragem:** é preciso ter coragem para admitir problemas, pedir auxílio quando necessário, alterar algo já pronto, dizer ao cliente que haverá atraso no prazo, enfim, fazer o que é correto, independente das reações (BONA, 2002).
- **Feedback:** quanto mais cedo se descobrir o problema, mais cedo será corrigido.

Ao invés de ser organizado de forma rigorosa, em processos burocráticos, o desenvolvimento é baseado em 12 práticas simples, a saber (BONA, 2002; *EXTREME*, 2009):

- **Histórias do Usuário (*user stories*):** as funcionalidades do sistema são descritas em histórias pelo próprio cliente com suas palavras e da forma mais simples possível. Estas histórias substituem os longos documentos

de requisitos nos métodos tradicionais e cada história deve ser descrita em aproximadamente três sentenças.

- **Plano de Entregas (*Release Planning*):** após a definição das histórias é necessário estimar o tempo de implementação das mesmas para que o cliente priorize o que deve ser implementado. É realizada uma reunião para o planejamento de entregas (cronograma de cada história). A idéia é que um projeto possa ser quantificado em quatro variáveis: escopo, recursos, tempo e qualidade, sendo as regras do negócio (escopo, prioridade, composição das versões e datas das versões) estimadas pelo cliente e as considerações técnicas (tempo, riscos técnicos e processo) estimadas pelos técnicos. A Figura 4.1 ilustra um modelo de Cartão de História.

Cartão de História e Tarefa			
Data ____/____/____.	Tipo de Atividade: Nova: ____ Dificuldade: ____ Valor: ____		
Número da História: ____	Prioridade: ____	Usuário: ____	Técnico: ____
Referência Anterior: ____	Risco: ____	Estimativa do Técnico: ____	
Descrição da Tarefa:			
Nome:			
Acompanhamento da tarefa:			
Data	Estado	Para Realizar	Comentário

Figura 4.1 – Modelo de Cartão de História no XP
BONA (2002, p. 42)

- **Pequenas Versões (*Small Releases*):** nas reuniões de planejamento são definidas as funcionalidades do sistema que serão implementadas a cada iteração de desenvolvimento para que o cliente possa se beneficiar do sistema. A entrega de cada *release* não deve ultrapassar o prazo de dois meses. Quanto mais rápido se introduzir uma funcionalidade no sistema, maior será o tempo para consertá-la, caso seja necessário.
- **Iterações (*Iterations*):** é a divisão dos *releases* em espaços menores para reduzir o tempo para o *feedback* com o cliente, pois dois meses é um

tempo longo demais para este *feedback*. Cada iteração contém um conjunto de histórias a serem implementadas, podendo durar de uma a três semanas, tendo ao final a avaliação do cliente. A equipe recebe orientações através de cartões de tarefas, cujo modelo é ilustrado na Figura 4.2.

Cartão de Tarefa

Data ____/____/____.

Número da História: _____ Autor do Software _____ Estimativa da Tarefa _____

Descrição da tarefa:

Notas do Autor do Software:

Acompanhamento da tarefa:

Data	Realizado	Para Realizar	Comentário

**Figura 4.2 – Modelo de Cartão de Tarefa no XP
BONA (2002, p. 43)**

- **Plano de Iteração (*Iteration Planning*):** no início de cada iteração é feita uma reunião para que o cliente possa definir quais histórias serão implementadas, priorizando as que possuem maior valor para o mesmo.
- **Reuniões Rápidas (*Stand-Up Meetings*):** cada dia de trabalho da equipe é iniciada com uma rápida reunião (aproximadamente 20 minutos) para comunicar problemas, soluções e decidir as histórias que serão implementadas no dia e, em conjunto, definir os responsáveis por cada uma delas. Estas reuniões devem ser feitas, preferencialmente, com todos os integrantes em pé para evitar conversas paralelas e fazer os integrantes irem direto ao assunto, agilizando e simplificando a reunião.
- **Conserte o XP (*Fix XP*):** quando o processo falhar, o mesmo deve ser corrigido. As regras do XP devem ser seguidas, mas não se deve hesitar em alterar o que não funcionar.
- **Soluções Rápidas (*Spike Solutions*):** para resolver problemas difíceis devem ser criadas soluções rápidas. Programas simples devem ser

criados para explorar soluções em potencial, reduzindo o risco de um problema.

- **Reestruturação (*Refactor*):** mesmo que o código esteja funcionando perfeitamente, deve-se reestruturá-lo sempre, removendo redundâncias, eliminando funcionalidades não utilizadas e modificando arquiteturas obsoletas. Todo desenvolvedor deve promover esta reestruturação, deixando o código mais legível e simples sem, no entanto, alterar o comportamento do mesmo.
- **Programação em Pares (*Pair Programming*):** a implementação de qualquer código deve ser feita em dupla, denominada de programação em par. Dois desenvolvedores trabalham no mesmo problema, ao mesmo tempo e no mesmo computador. Um deles é o responsável pela codificação (condutor), geralmente o novato. O outro, mais experiente, acompanha o trabalho do parceiro (navegador), revisando o código digitado, ajudando o outro a desenvolver suas habilidades, percebendo erros de programação que poderiam levar horas para serem depurados e cobrando padrões de desenvolvimento da equipe. Além disso, há uma troca de experiências e idéias entre os dois, facilitando na busca de soluções para possíveis problemas. Os papéis e os pares são trocados freqüentemente, permitindo que toda equipe conheça e possa alterar o código. Esta rotatividade do código representa sua propriedade coletiva, encorajando toda a equipe a colaborar com novas idéias. Qualquer membro da equipe pode adicionar funcionalidades, corrigir erros ou reestruturar o código. Todos são responsáveis pelo código inteiro. As codificações devem seguir padrões pré-estabelecidos pela equipe para que todos possam entendê-las.
- **Semana de trabalho de 40 horas:** o XP prega o ritmo sustentável da equipe, proibindo que os desenvolvedores trabalhem até mais tarde, respeitando suas condições físicas e psicológicas e garantindo a concentração da equipe, para reduzir pequenas falhas na implementação.
- **Testes:** o XP utiliza dois tipos de testes: o Teste Unitário e o Teste de Aceitação.
 - **Teste Unitário (*Unit Test*):** Todo código é testado através de *scripts* de teste automatizado que são desenvolvidos pelos próprios

desenvolvedores antes da codificação e servem para a validação do código. Estes testes devem ser automatizados para que possam ser executados rapidamente e rodem constantemente, garantindo que o código tenha as funcionalidades esperadas. Somente os códigos testados podem ser integrados ao sistema.

Se um erro for encontrado, novos testes devem ser criados, garantindo que o erro não passe novamente pelo antigo teste unitário.

Os testes unitários são a fonte de coragem para que o desenvolvedor realize a fatoraçoão, removendo duplicações e tornando o código mais flexível e legível.

- **Testes de Aceitação (*Acceptance Tests*):** cada teste de aceitação é elaborado pelo cliente para verificar uma funcionalidade descrita numa história do usuário. Sempre que houver uma nova integração devem ser rodados. São escritos no momento da escrita da história, devendo haver pelo menos um teste de aceitação para cada história. A história só é declarada terminada quando passar por todos os testes de aceitação.

4.2.1 Mecanismo de Desenvolvimento no XP

A Figura 4.3 ilustra o mecanismo do Modelo XP, cujo ciclo de vida compreende as fases de exploração, planejamento, iteração, produção, manutenção e fim do projeto (BONA, 2002).

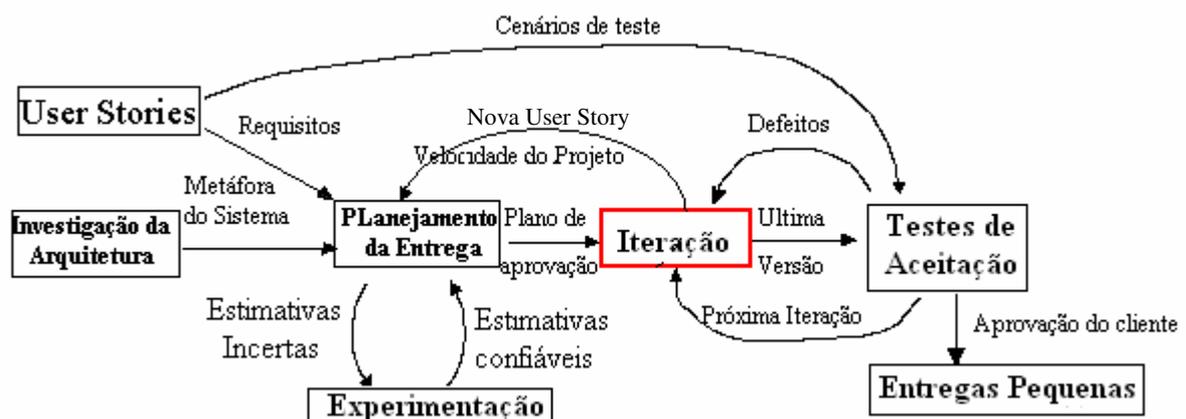


Figura 4.3 – Ciclo de vida nos projetos XP
SANTOS; DAHER (2008, p. 6)

- **Exploração**

Esta fase é iniciada com as regras do negócio e tem como objetivo o bom entendimento do que o sistema deve fazer para que possa ser estimado. O cliente escreve as histórias e o programador as estima, encerrando-se esta fase quando todas as histórias necessárias para a próxima fase tiverem sido estimadas. Enquanto os clientes vão escrevendo as histórias, os programadores vão experimentando diferentes tecnologias e configurações, explorando possibilidades para a arquitetura do sistema (BONA, 2002).

- **Planejamento**

A melhor maneira de executar esta fase é utilizando o Jogo do Planejamento. O cliente prioriza as histórias e é definida a menor data e o maior número de histórias que farão parte desta primeira versão, de acordo com estimativas entre clientes e programadores. Para auxiliar esta fase, alguns passos podem ser seguidos:

- O cliente seleciona as histórias por valor, enquanto que os programadores as qualificam por risco: alto, médio ou baixo;
- Os programadores declaram a velocidade, calculada empiricamente, baseada na experiência dos mesmos;
- Clientes escolhem o escopo: as histórias que farão parte da próxima versão.

As histórias do cliente são quebradas em pequenas tarefas e definidos quais programadores irão trabalhar em cada tarefa. As histórias que serão trabalhadas são decididas pela equipe no primeiro dia de cada iteração. As tarefas são selecionadas e pontuadas em dias pelos programadores.

- **Iteração**

Conforme Beck, 2000 apud Bona (2002), “os compromissos são divididos para serem executados em iterações que duram de uma a quatro semanas”. São produzidos testes funcionais para cada história executada naquela iteração.

O desenvolvimento é conduzido por uma seqüência de ciclos iterativos, concentrando o projeto, a codificação, os testes e as versões do produto. Ao final de cada iteração, o cliente completa todos os testes funcionais, sendo que no final da última iteração, o sistema estará pronto para a fase de produção.

- **Produção**

No final de uma versão, dá-se a produção do sistema. Podendo ser implementados novos testes para provar a estabilidade do sistema ou serem realizados ajustes no desempenho.

- **Manutenção**

Um projeto XP está constantemente em fase de manutenção, pois simultaneamente a produção de novas funcionalidades, deve-se manter o sistema existente rodando, substituir membros da equipe e incorporar novos membros.

Esta é a fase que pode-se tentar *refactorings* maiores, que causaram receio de serem tentados nas versões anteriores. Pode-se testar novas tecnologias ou migrar a tecnologia em uso para versões mais atualizadas. O cliente pode escrever novas histórias que melhorem o seu negócio.

- **Fim do projeto**

O momento de finalizar o projeto é quando o cliente está satisfeito com o sistema e não consegue mais escrever histórias.

Sugere-se então que sejam escritas algumas páginas (de 5 a 10) sobre a funcionalidade do sistema, para auxiliar futuras alterações no sistema.

Toda a equipe deve se reunir para uma reavaliação, aproveitando para analisar os pontos positivos e negativos do projeto.

4.2.2 Papéis

Desenvolvido para equipes pequenas e médias (BECK, 1999), alguns papéis são fundamentais no XP como programador, cliente, treinador e supervisor, sendo que outros papéis podem ser exercidos pela mesma pessoa, como por exemplo, gerente e supervisor.

O programador é responsável por estimar prazos, definir os cartões de tarefas a partir dos cartões de histórias, estimar os cartões de tarefas, implementar testes unitários, implementar o código de produção, trabalhar em par, fazer *refactoring* sempre que necessário, estar em contato direto com o cliente para *feedbacks*.

Outro papel essencial é o do cliente. Além de pagar pelo projeto, ele deve estar disposto a aprender, pois é o responsável por definir os requisitos do sistema, escrever os cartões de história, definir as prioridades das histórias, validar e definir os testes funcionais e esclarecer dúvidas sempre que solicitado.

O papel do testador em XP é aplicar os testes. Sua responsabilidade é definir com o cliente os testes funcionais do projeto, escrevê-los, executá-los e publicar os resultados dos mesmos para a equipe.

O supervisor é responsável por coletar as métricas do projeto uma ou duas vezes por semana, manter todos informados do que está acontecendo e tomar atitudes sempre que as coisas parecerem ir mal.

A função do treinador é garantir que o projeto permaneça extremo, ajudar com o que for necessário, manter a visão do projeto, não deixando que o time se desvie do processo, formular e comunicar uma tarefa que um programador queira trabalhar.

Por fim, ao gerente cabe transmitir coragem, confiança e saber cobrar o que é de responsabilidade de cada um. Seu trabalho consiste em gerenciar a equipe e seus problemas, agendar reuniões de planejamento, garantir que as mesmas fluam como planejado, documentar o que foi definido nas reuniões, manter o supervisor informado dos acontecimentos das reuniões e buscar recursos.

Tendo por base que “o XP defende que não se deve criar um grande volume de documentos ou diagramas que podem ficar desatualizados. Uma vez que, o objetivo é ganhar tempo para ir mais rápido” (BONA, 2002, p. 59), no próximo capítulo será apresentada a modelagem do sistema SCVE-BSCJ.

5 MODELAGEM DO SISTEMA SCVE-BSCJ

Este capítulo apresenta a modelagem do Sistema SCVE-BSCJ, bem como o conjunto de práticas empregadas, tendo como base o Modelo de Processo de Desenvolvimento XP.

5.1 Visão do Sistema

Em um contexto inicial (fase de Exploração), a estratégia proposta pelo XP consiste em elaborar, em poucas linhas, os objetivos do sistema, para que o mesmo possa ser estimado. Para isso, é fundamental o entendimento do sistema conforme a visão do cliente.

Neste contexto, foram levantadas, por meio de reuniões rápidas, todas as histórias do usuário, conforme pode ser observado na seção 5.1.3. Eventualmente, alguma história foi adicionada ou removida, conforme os requisitos foram sendo esclarecidos. Ressalta-se que um cartão de história é apenas um lembrete de uma conversa com cliente (ASTELS, 2002 apud BONA, 2002), não contendo todos os detalhes necessários à codificação do comportamento. Para isso, foram necessárias conversas diretas com o cliente, pessoalmente ou por meio de telefonemas.

Em paralelo as histórias dos clientes, foram exploradas as possibilidades para a arquitetura do sistema.

Na fase de planejamento foi especificado o projeto, as iterações e o dia-a-dia, com o objetivo de estimar o menor tempo possível e o maior número de histórias para a primeira versão. A estimativa foi elaborada com base nos cartões de histórias e em soluções simples, lembrando que o projeto poderia ser replanejado, caso alguma alteração significativa fosse identificada pelo cliente ou pelos desenvolvedores.

5.1.1 Composição e tarefas da equipe

A equipe foi composta pelo cliente que definiu os requisitos, fixou as prioridades e guiou o projeto, pela dupla de programadores que também ajudaram o cliente a definir os testes de aceitação e os requisitos e pelo gerente que provia recursos e mantinha a comunicação externa, além de coordenar as atividades. Os papéis não foram exclusivamente de propriedade de um só indivíduo, sendo que todos os membros da equipe colaboraram de todas as formas que puderam, de acordo com suas habilidades.

5.1.2 Escopo do Produto

O Sistema SCVE-BSCJ tem por objetivo controlar as vendas e o estoque do Bazar do Sagrado Coração de Jesus, mantendo um cadastro de todos os usuários do sistema, dos artesãos que fornecem os produtos ao bazar e dos produtos fornecidos pelos mesmos.

A entrada de produtos, bem como a venda dos mesmos, atualiza automaticamente tanto o estoque de produtos quanto os valores devidos ao artesão e ao bazar, mantendo a integridade dos dados.

Para auxiliar na tomada de decisões, relatórios financeiros, de estoque e de cadastros atualizados podem ser obtidos a qualquer momento por meio do menu principal.

Todos os usuários do sistema possuem os mesmos acessos, não havendo nenhum tipo de restrição ou privilégio.

5.1.3 Histórias do usuário

Nesta seção são apresentadas as histórias do usuário. Para facilitar a visualização, elas foram organizadas no formato de tabela, conforme pode ser observado na Tabela 5.1.

Tabela 5.1 – Cartões de história resumidos

Nº História	Descrição	Prioridade
001	Cadastrar os dados dos usuários do sistema através de formulários.	Alta
002	Cadastrar os dados dos artesãos (fornecedores) do bazar através de formulários.	Alta
003	Cadastrar os produtos que são vendidos no bazar através de formulários.	Alta
004	Validar os usuários que terão acesso ao sistema, impedindo que pessoas estranhas tenham acesso ao mesmo.	Alta
005	Realizar entrada de produtos fornecidos pelos artesãos, cujos preços podem variar de produto para produto bem como o mesmo artesão ter dois produtos iguais com preços diferentes.	Alta
006	Realizar devolução de produtos aos artesãos.	Média
007	Realizar venda de produtos.	Alta
008	Gerar cálculo de repasse do valor das vendas ao artesão.	Média
009	Disponibilizar formulário para entradas/saídas de dinheiro do bazar.	Média
010	Disponibilizar formulário para pagamento dos artesãos.	Média
011	Imprimir relatórios de controle de estoque.	Média
012	Imprimir relatórios de produtos com permanência no bazar maior que três meses.	Média
013	Imprimir relatórios da situação financeira do bazar por período.	Média
014	Imprimir relatórios de repasse ao bazar por período.	Baixa
015	Imprimir relatórios de cadastro de usuários, artesãos e produtos.	Baixa
016	Imprimir relatórios de movimentação financeira.	Média
017	Imprimir relatórios de vendas.	Média
018	Imprimir etiquetas de produtos.	Baixa

5.1.4 Estimativa, priorização e planejamento

Após o relato das histórias, o cliente as classificou por prioridade, sendo definido: alta (máxima urgência), média (necessárias, mas poderiam aguardar por algum tempo) e baixa (interessantes após a conclusão de outras histórias), como também pode ser observado na Tabela 5.1.

Com base nestas informações, a equipe de programação estimou as histórias em semanas, sendo que cada história não poderia ultrapassar três semanas, pois “histórias menores tendem a ter risco menor” (WAKE, 2002 apud BONA, 2002, p.52), além de permitir a implementação de um conjunto de histórias (as de maior prioridade) a cada iteração. Caso a estimativa da história ultrapassasse o prazo, ela deveria retornar ao cliente para que fosse dividida em histórias menores.

Ainda, foram planejadas as iterações, total de 4, como pode ser verificado na Tabela 5.2, além da estimativa e priorização. O planejamento das iterações foi de suma importância, pois cada iteração possuía as funcionalidades de maior prioridade para o cliente, agilizando a entrega do sistema ao cliente e permitindo sua avaliação e *feedback*, pois o mesmo pôde perceber detalhes não previstos inicialmente.

Tabela 5.2 – Cartões de história com prioridades, estimativa e interações

Nº História	Descrição	Prioridade	Estimativa	Iteração
001	Cadastrar os dados dos usuários do sistema através de formulários.	Alta	½ semana	1ª
002	Cadastrar os dados dos artesãos (fornecedores) do bazar através de formulários.	Alta	½ semana	1ª
003	Cadastrar os produtos que são vendidos no bazar através de formulários.	Alta	½ semana	1ª
004	Validar os usuários que terão acesso ao sistema, impedindo que pessoas estranhas tenham acesso ao mesmo.	Alta	1 semana	1ª
005	Realizar entrada de produtos fornecidos pelos artesãos, cujos preços podem variar de produto para produto bem como o mesmo artesão ter dois produtos iguais com preços diferentes.	Alta	1 ½ semanas	2ª
006	Realizar devolução de produtos aos artesãos.	Média	1 semana	3ª
007	Realizar venda de produtos.	Alta	2 semanas	2ª
008	Gerar cálculo de repasse do valor das vendas ao artesão.	Média	1 semana	3ª
009	Disponibilizar formulário para entradas/saídas de dinheiro do bazar.	Média	1 semana	3ª
010	Disponibilizar formulário para pagamento dos artesãos.	Média	1 semana	4ª
011	Imprimir relatórios de controle de estoque.	Média	½ semana	4ª
012	Imprimir relatórios de produtos com permanência no bazar maior que três meses.	Média	1 semana	4ª
013	Imprimir relatórios da situação financeira do bazar por período.	Média	½ semana	4ª
014	Imprimir relatórios de repasse ao bazar por período.	Baixa	½ semana	4ª
015	Imprimir relatórios de cadastro de usuários, artesãos e produtos.	Baixa	1 semana	1ª
016	Imprimir relatórios de movimentação financeira.	Média	1 semana	3ª
017	Imprimir relatórios de vendas.	Média	½ semana	2ª
018	Imprimir etiquetas de produtos.	Baixa	½ semana	4ª

Observado que os cartões de história formavam a funcionalidade básica do sistema, optou-se por desenvolver a primeira versão do sistema, que foram seguidas por outras versões para aperfeiçoamento e complemento de recursos.

5.1.5 Funções do Produto

A solução proposta para o SCVE-BSCJ consiste em uma série de módulos que trabalham de forma integrada para fornecer as seguintes funcionalidades:

- Gerenciar Usuário;
- Gerenciar Artesão;
- Gerenciar Produto;
- Gerenciar Movimentação de Produto;
- Gerenciar Movimentação Financeira;
- Gerenciar Relatório.

As funcionalidades afins foram agrupadas em módulos denominados genericamente por “gerenciar”. Por exemplo, funcionalidades como o cadastro, exclusão, alteração e consultas relacionadas ao usuário serão agrupadas no módulo Gerenciar Usuário.

Com base nas histórias dos usuários, para cada módulo, foram definidos os requisitos funcionais e não funcionais, os quais seguem a regra:

- **Requisitos Funcionais** possuem o identificador [RFabc]; onde a, b,c são dígitos que variam entre 0 e 9.
- **Requisitos Não-Funcionais** possuem o identificador [RNFacb]; onde a, b, c são dígitos que variam entre 0 e 9, RNF significa Requisito Não Funcional.

No que se refere à prioridade dos requisitos foram mantidas as denominações sugeridas pelos usuários.

A Tabela 5.3 apresenta os requisitos funcionais e a Tabela 5.4 os requisitos não funcionais. Ambas foram organizadas pelos módulos definidos.

Tabela 5.3 – Requisitos Funcionais

Gerenciar Usuário			
Requisito	Descrição	Prioridade	Iteração
RF001 – Cadastrar Usuário	O sistema deve permitir a inserção de um usuário. Os itens de informação são código para identificar o usuário, nome, endereço, bairro, cidade, CEP, telefones, celular, RG, CPF, <i>login</i> e senha.	Alta	1ª
RF002 – Atualizar dados do Usuário	O sistema deve permitir a alteração dos dados do usuário, com exceção do código que identifica o usuário.	Alta	1ª
RF003 – Excluir Usuário	O sistema deve prover mecanismos para permitir a exclusão de um determinado usuário.	Alta	1ª
RF004 – Validar Usuário	O sistema deve permitir a validação do usuário para acesso às funcionalidades do sistema.	Alta	1ª
RF005 – Localizar Usuário	O sistema deve permitir a procura rápida dos dados cadastrais de um determinado usuário através de seu nome.	Alta	1ª
Gerenciar Artesão			
Requisito	Descrição	Prioridade	Iteração
RF006 – Cadastrar Artesão	O sistema deve permitir a inserção de um artesão. Os itens de informação são código para identificar o artesão, nome, endereço, bairro, cidade, CEP, telefones, celular, RG, CPF, PAT.SUTACO, e-mail e se ele repassa ou não 10% de contribuição ao bazar, além do código do usuário que o cadastrou.	Alta	1ª
RF007 – Atualizar Dados do Artesão	O sistema deve permitir a alteração dos dados do artesão, com exceção do código que identifica o artesão.	Alta	1ª
RF008 – Excluir Artesão	O sistema deve prover mecanismos para permitir a exclusão de um determinado artesão.	Alta	1ª
RF009 – Localizar Artesão	O sistema deve permitir a procura rápida dos dados cadastrais de um determinado artesão através de seu nome.	Alta	1ª

Tabela 5.3 – Requisitos Funcionais

Gerenciar Produto			
Requisito	Descrição	Prioridade	Iteração
RF010 – Cadastrar Produto	O sistema deve permitir a inserção de um produto. Os itens de informação são código para identificar o produto, descrição e quantidade.	Alta	1ª
RF011 – Atualizar Dados do Produto	O sistema deve permitir a alteração dos dados do produto, com exceção do código que identifica o produto.	Alta	1ª
RF012 – Excluir Produto	O sistema deve prover mecanismos para permitir a exclusão de um determinado produto.	Alta	1ª
RF013 – Localizar Produto	O sistema deve permitir a procura rápida dos dados cadastrais de um determinado produto através de seu nome.	Alta	1ª
Gerenciar Movimentação de Produto			
Requisito	Descrição	Prioridade	Iteração
RF014 – Inserir Item Fornecido	O sistema deve permitir a entrada de produtos fornecidos pelos artesãos. Os itens de informação são código para identificar o item fornecido, data do fornecimento, código do produto, código do artesão que forneceu o produto, código do usuário que recebeu o produto, descrição do item fornecido, valor unitário e quantidade fornecida. Automaticamente, ao ser inserido um produto, deve ser atualizado o estoque.	Alta	2ª
RF015 – Atualizar Dados do Fornecimento do Produto	O sistema deve permitir a alteração dos dados dos produtos fornecidos, com exceção do código que identifica o item fornecido. Caso seja alterada a quantidade do produto fornecido, bem como o código do produto, o sistema deve fazer as alterações necessárias no estoque.	Alta	2ª
RF016 – Excluir Item Fornecido	O sistema deve prover mecanismos para permitir a exclusão de um determinado produto, atualizando ao mesmo tempo o estoque.	Alta	2ª
RF017 – Localizar Item Fornecido	O sistema deve permitir a procura rápida dos dados cadastrais de um determinado produto fornecido através de seu código.	Alta	2ª

Tabela 5.3 – Requisitos Funcionais

Gerenciar Movimentação de Produto			
Requisito	Descrição	Prioridade	Iteração
RF018 – Devolver Item Fornecido	O sistema deve permitir a devolução de um item fornecido, provendo atualização automática do estoque e da quantidade de item fornecido.	Média	3ª
RF019 – Imprimir Etiquetas de Produtos	O sistema deve permitir a impressão de etiquetas de produtos, contendo o código do item fornecido, sua descrição, o nome do artesão que o forneceu e o respectivo valor unitário.	Baixa	4ª
Gerenciar Movimentação Financeira			
Requisito	Descrição	Prioridade	Iteração
RF020 – Efetuar Vendas	O sistema deve permitir a venda de um produto. Os itens de informação são código para identificar a venda, código do usuário que efetuou a venda, data, código do item fornecido vendido e quantidade. Ao ser efetuada uma venda, o estoque deve ser atualizado, bem como o valor das receitas e o valor devido ao artesão decorrente da venda.	Alta	2ª
RF021 – Inserir Despesas	O sistema deve permitir inserir despesas financeiras. Os itens de informação são código da despesa, código do usuário responsável, data, descrição e valor.	Média	3ª
RF022 – Inserir Receitas	O sistema deve permitir inserir receitas. Os itens de informação são código da receita, código do usuário responsável, data, descrição e valor.	Média	3ª
RF023 – Fechar Caixa Diário	O sistema deve prover mecanismos para que seja feito o fechamento do caixa diariamente, obtendo o saldo para verificação do caixa.	Média	3ª
RF024 – Pagar Artesão	O sistema deve calcular o valor devido ao artesão por período determinado pelo usuário, emitindo um recibo de pagamento e atualizando o caixa.	Média	3ª

Tabela 5.3 – Requisitos Funcionais - Conclusão

Gerenciar Relatório			
Requisito	Descrição	Prioridade	Iteração
RF025 – Emitir Relatório de Estoque	O sistema deve permitir a emissão de relatório de estoque. Os itens de informação são código do produto, descrição e quantidade.	Média	4ª
RF026 – Emitir Relatório de Itens Fornecidos a Devolver	O sistema deve permitir a emissão de relatório de itens fornecidos há mais de três meses.	Média	4ª
RF027 – Emitir Relatório de Caixa	O sistema deve permitir a emissão de relatório do caixa por dia determinado pelo usuário.	Média	3ª
RF028 – Emitir Relatório de Repasse ao Bazar	O sistema deve permitir a emissão de relatório de repasse ao bazar por período determinado pelo usuário.	Baixa	4ª
RF029 – Emitir Relatórios de Cadastro de Artesãos, Produtos e Usuários	O sistema deve permitir a emissão de relatórios contendo os dados cadastrais dos artesãos, produtos e usuários.	Alta	1ª
RF030 – Emitir Relatórios de Movimentação Financeira	O sistema deve permitir a emissão de relatórios contendo todas as receitas e despesas de determinado período selecionado pelo usuário.	Média	3ª
RF031 – Emitir Relatórios de Vendas	O sistema deve permitir a emissão de relatórios contendo código, data e usuário responsável pela venda.	Média	2ª

Tabela 5.4 – Requisitos não Funcionais

Requisito	Descrição	Prioridade
RNF001 – BDE <i>Administrator</i>	O sistema necessita que se tenha o BDE <i>Administrator</i> instalado na máquina.	Alta
RNF002 – Sistema Operacional <i>Windows</i>	É necessário ter o sistema operacional <i>Windows</i> XP ou superior na máquina onde o sistema será instalado.	Alta
RNF003 – Banco de Dados <i>MSAccess</i>	O <i>MSAccess</i> 2000 ou superior deve estar instalado na máquina.	Alta
RNF004 – Programa <i>WinRar</i>	É necessário ter o programa <i>WinRar</i> instalado na máquina.	Alta
RNF005 - <i>Hardware</i>	Configuração mínima: Computador Pentium 4 2,79 GHz ou semelhante 1 Gb de memória RAM 150 Mb de espaço livre em disco Unidade de CD-ROM ou USB	Alta

Por se tratar de um processo mais complexo, optou-se, também, por apresentar a decomposição do processo denominado Processar Vendas, conforme Figura 5.2.

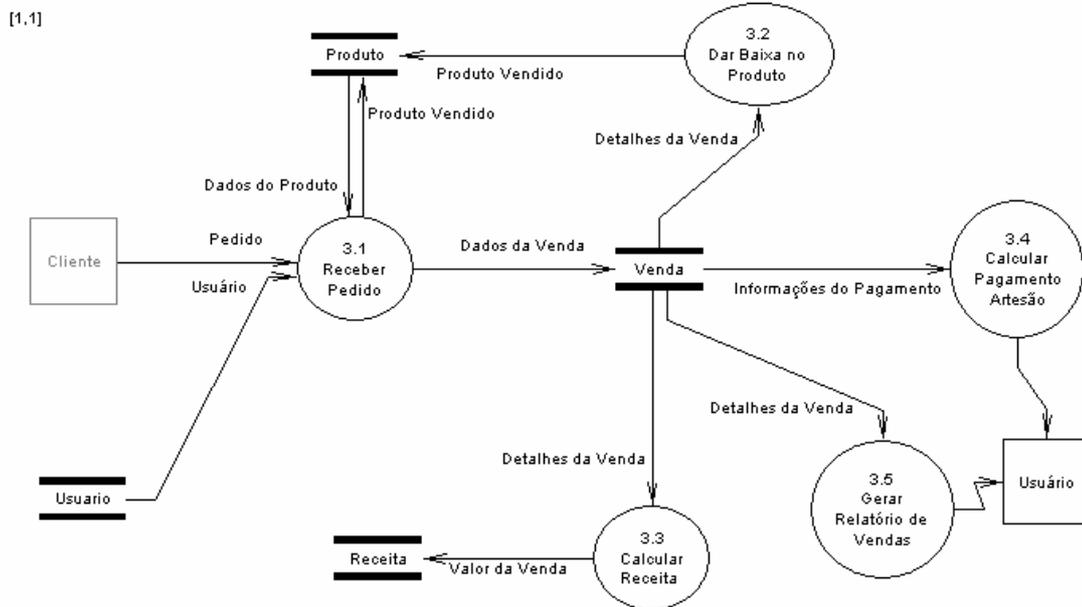


Figura 5.2 – Decomposição do Processo: Processar Vendas

Para demonstrar em detalhes as funcionalidades contidas nos módulos denominados Gerenciar, apresenta-se a Figura 5.3 que ilustra o Módulo Gerenciar Usuários.

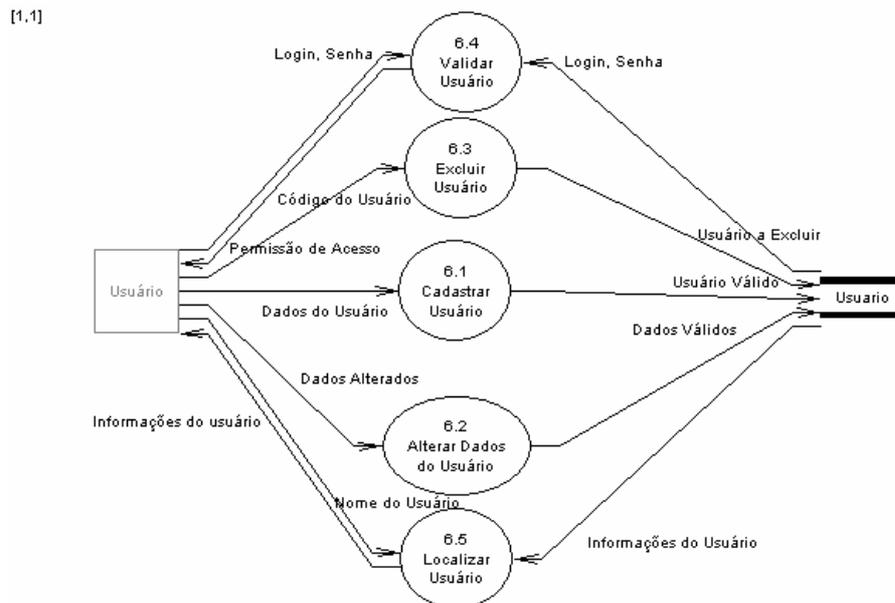


Figura 5.3 – Decomposição do Processo: Gerenciar Usuários

O Diagrama Entidade-Relacionamento (DER), demonstrado na Figura 5.4, foi adotado para expressar graficamente toda a estrutura lógica do banco de dados (MENESES, 2009) do Sistema SCVE-BSCJ. Por motivo de melhor estruturação deste trabalho, foram ocultados os atributos no DER apresentado na Figura 5.4, os quais são apresentados na Figura 5.5 juntamente com as definições de chaves primárias. A Figura 5.5 ilustra o Esquema Relacional, obtido com a ferramenta Case Studio, voltado para a análise estruturada de sistemas.

Enquanto o DFD enfatiza as funções do sistema, o DER descreve cada depósito de dados do DFD e o relacionamento entre eles.

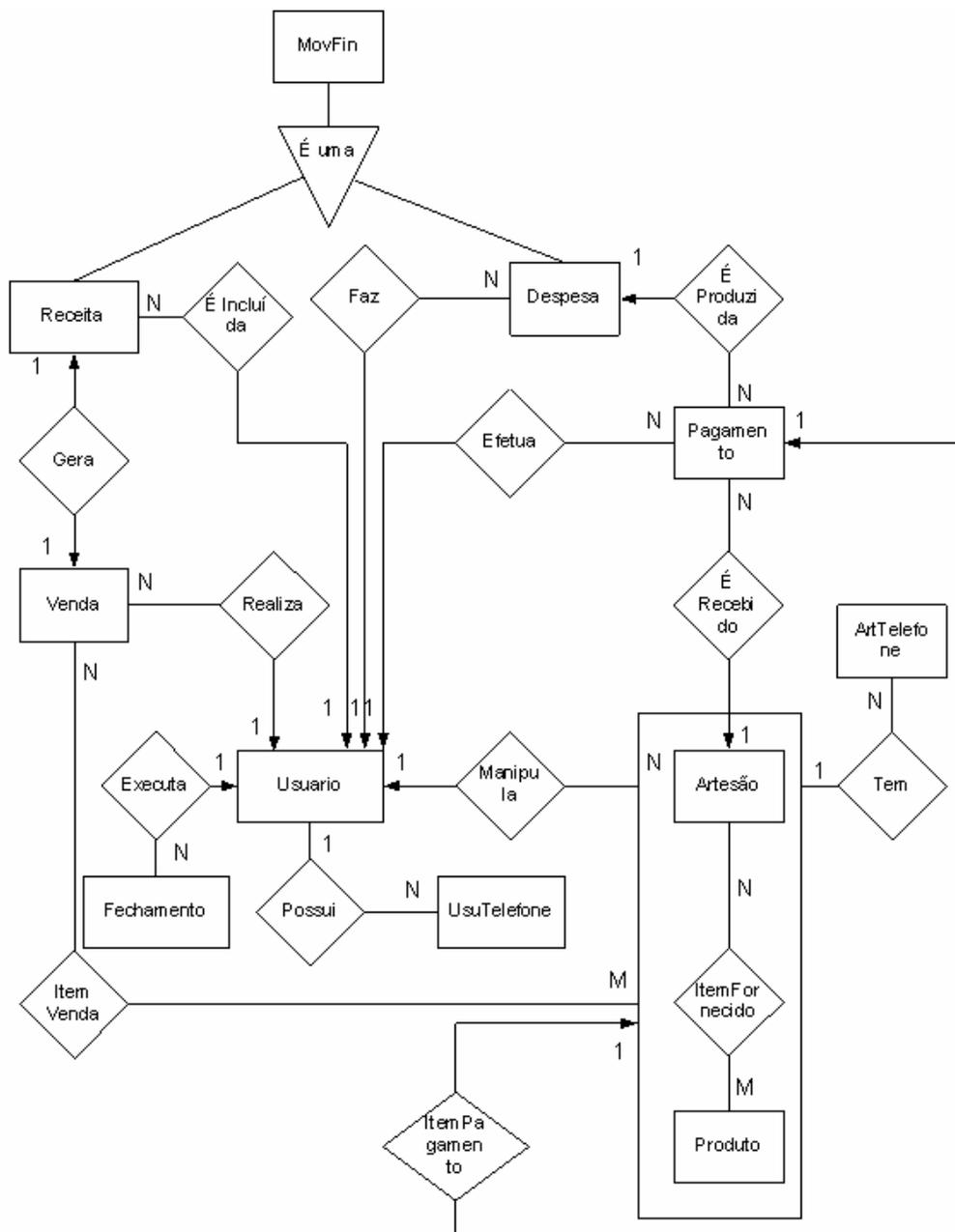


Figura 5.4 – Diagrama Entidade-Relacionamento do Sistema SCVE-BSCJ

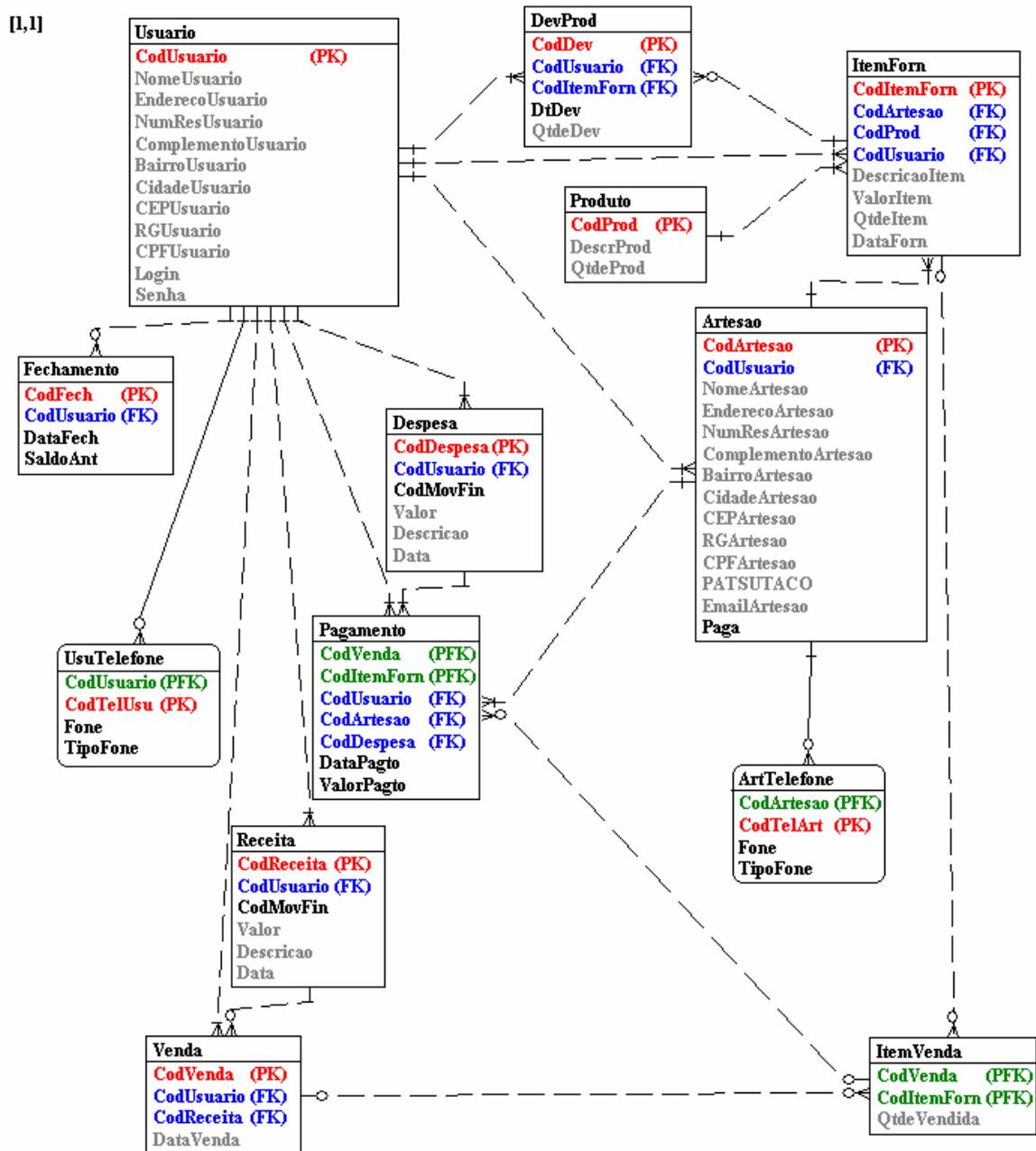


Figura 5.5 – Esquema Relacional do Sistema SCVE-BSCJ

5.1.6 Testes de Aceitação

Da mesma forma que os programadores realizaram os testes no sistema - teste estrutural e funcional (BONA, 2002), também foram realizados os testes de aceitação pelo cliente. Percebidas algumas inconsistências, estas foram prontamente corrigidas pelos programadores. Os testes de aceitação foram realizados por módulos disponibilizados ao cliente.

5.1.7 Fim do projeto

Após a fase de produção, quando foi desenvolvido o Sistema SCVE-BSCJ, visto que o cliente deu-se por satisfeito, não havendo mais nenhuma história a acrescentar, encerrou-se o projeto com a produção da documentação, contendo os diagramas ilustrados nas Figuras 5.1 à 5.4, os cartões de histórias do usuário, cujos resumos são apresentados na Tabela 5.1 e os requisitos funcionais apresentados na Tabela 5.3 e os requisitos não funcionais apresentados na Tabela 5.4.

No próximo capítulo é apresentado o protótipo do Sistema SCVE-BSCJ obtido à partir dos diagramas ilustrados neste capítulo.

6 PROTÓTIPO DO SISTEMA

Este capítulo tem por objetivo apresentar o protótipo do Sistema SCVE-BSCJ, explicando cada uma de suas funcionalidades.

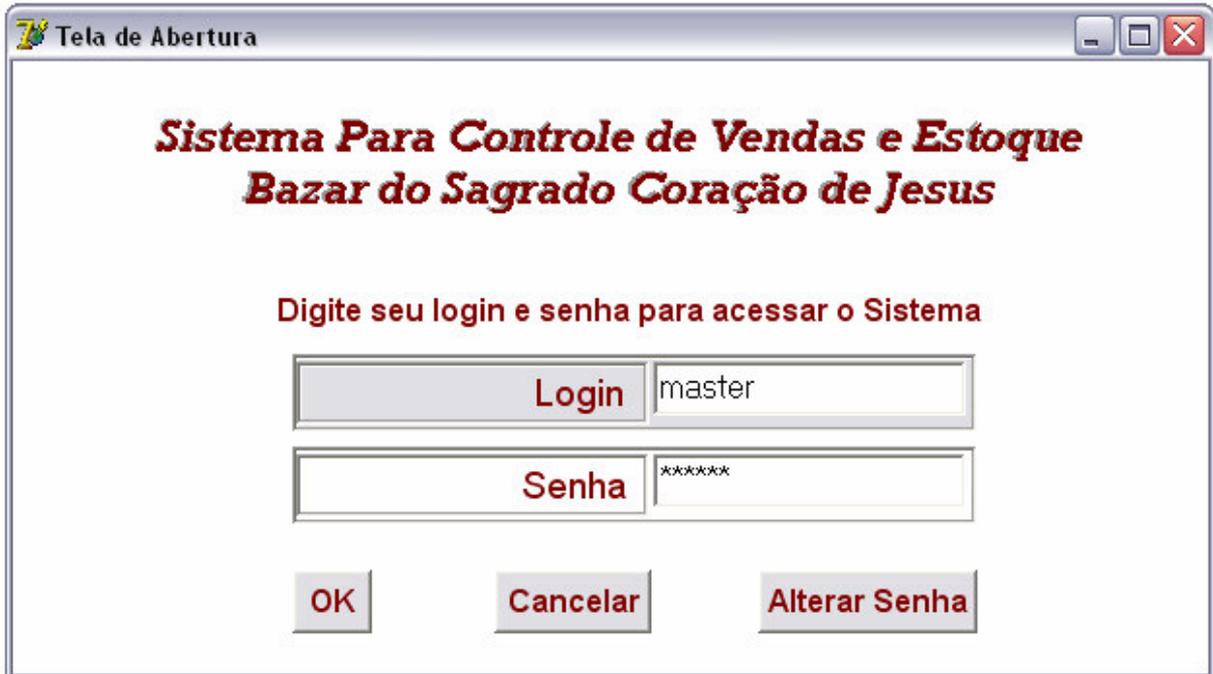
Foram implementados os seguintes módulos:

- Gerenciar Usuários;
- Gerenciar Artesãos;
- Gerenciar Produtos;
- Validação de Usuários;
- Gerenciar Movimentação de Produtos;
- Gerenciar Movimentação Financeira;
- Gerenciar Relatórios.

O protótipo é descrito neste capítulo obedecendo à seqüência cronológica de sua estrutura arquitetural.

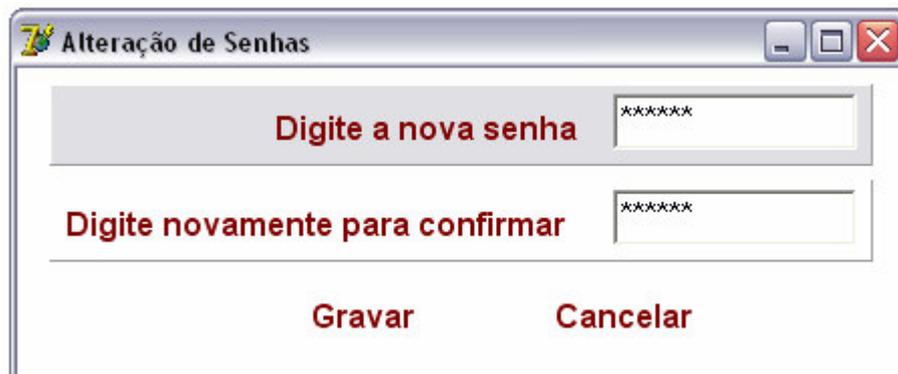
6.1 Apresentação do Protótipo do Sistema

Ao acessar o Sistema SCVE-BSCJ, é exibida a tela de abertura ilustrada na Figura 6.1. É necessário o *login* e a senha do usuário para que, após autenticados, possam liberar o acesso às outras funcionalidades. O botão 'OK' apresenta o formulário principal, conforme mostra a Figura 6.3, e o botão 'Alterar Senha' permite ao usuário alterar sua própria senha, por meio do formulário apresentado na Figura 6.2.



The screenshot shows a window titled "Tela de Abertura" with a logo on the left and standard window controls on the right. The main text is "Sistema Para Controle de Vendas e Estoque Bazar do Sagrado Coração de Jesus" in a bold, red, serif font. Below this, it says "Digite seu login e senha para acessar o Sistema". There are two input fields: the first is labeled "Login" and contains the text "master"; the second is labeled "Senha" and contains seven asterisks. At the bottom, there are three buttons: "OK", "Cancelar", and "Alterar Senha".

Figura 6.1 – Tela de Abertura do SCVE-BSCJ



The screenshot shows a window titled "Alteração de Senhas" with standard window controls on the right. It contains two input fields, both with seven asterisks. The first field is labeled "Digite a nova senha" and the second is labeled "Digite novamente para confirmar". At the bottom, there are two buttons: "Gravar" and "Cancelar".

Figura 6.2 – Alterar Senha

O acesso ao Sistema SCVE-BSCJ é idêntico para todos os usuários, não sendo, portanto, necessário um gerenciamento de acesso. A Figura 6.3, mostra a tela de Menu Principal, onde o usuário fará a seleção das opções desejadas.



Figura 6.3 – Menu Principal

As figuras a seguir serão mostradas na sequência em que aparecem no Menu Principal. O módulo Cadastro apresenta os formulários de Cadastro de Artesãos (Figura 6.4), Cadastro de Produtos (Figura 6.5) e Cadastro de Usuários (Figura 6.6), bem como as respectivas relações (Figura 6.7, Figura 6.8 e Figura 6.9).

Pode-se notar pelas figuras ilustradas, que os formulários de Cadastro de Artesãos (Figura 6.4), Cadastro de Produtos (Figura 6.5) e Cadastro de Usuários (Figura 6.6) contém os botões necessários para se movimentar pelas tabelas Artesãos, Produtos e Usuários (botões Anterior, Próximo, Primeiro e Último) e para acessar todas as funcionalidades destas tabelas (botões Inserir, Alterar, Excluir, Gravar, Imprimir e Procurar).

Cadastro de Artesãos

Código
2

Nome
LETICIA TERESA

Endereço
RUA MARECHAL DEODORO

Número 510 **Complemento** **Bairro** VILA PRADO

Cidade PINDAMONHANGABA **CEP** 12.450-000

Fone 1 (12)3879-6225 **Fone 2** (56)9872-6286 **Fone 3** (12)6827-2256

Celular (12)9829-3827 **RG** 23.232.323 **CPF** 232.323.232/32

PAT.SUTACO 23232 **Email** leticiateresa@ig.com.br

Cadastrado por: MASTER **Paga 10% ?** Sim Não

 Inserir  Alterar  Excluir  Anterior  Próximo  Primeiro  Último

Gravar  Cancelar  Imprimir  Procurar  Fechar

Figura 6.4 – Cadastro de Artesãos

Cadastro de Produtos

Código
1

Descrição
PANO DE PRATO

Quantidade
15

 Inserir  Alterar  Excluir  Anterior  Próximo  Primeiro  Último

Gravar  Cancelar  Imprimir  Procurar  Fechar

Figura 6.5 – Cadastro de Produtos

Cadastro de Usuarios

Código Usuário
1

Nome
MASTER

Endereço
PRINCIPAL

Número **Complemento** **Bairro**
sn CENTRO

Cidade **CEP**
PINDAMONHANGABA 12.400-000

Fone 1 **Fone 2** **Fone 3**
() - () - () -

Celular **RG** **CPF**
() - 11.111.111-X 111.111.111/11

Login **Senha**
master *****

 Inserir
  Alterar
  Excluir
  Anterior
  Próximo
  Primeiro
  Último

 Gravar
  Cancelar
  Imprimir
  Procurar
  Fechar

Figura 6.6 – Cadastro de Usuários

Bazar do Sagrado Coração de Jesus

Listagem de Artesãos

Cód	Nome	PAT.SUTACO	Fone	Fone	Celular
1	FERNANDA MONTERO	2200	(12)3642-0000	(12)3643-0000	(12)9191-9191
2	JOANA APARECIDA SOUZA	8956	(12)3644-0000		
3	SÍLVIA HELENA SILVA	5632	(12)3637-3737		

Figura 6.7 – Lista de Artesãos

Bazar do Sagrado Coração de Jesus		
<i>Listagem de Produtos</i>		
Cód	Descrição	Qtde.
3	CAIXA DE PRESENTE DECORADA	50
2	PANO DE PRATO	60
1	TRUFAS	150

Figura 6.8 – Lista de Produtos

Bazar do Sagrado Coração de Jesus				
<i>Listagem de Usuários</i>				
Código	Nome	Fone	Fone	Celular
2	MARCIA MONTERO	(12)3644-5959		(12)9191-9191
1	MASTER	(12)3644-5616		
3	SUZETE RAMOS	(12)3637-5000		

Figura 6.9 – Lista de Usuários

No módulo Movimentação, o usuário pode optar pela Movimentação Financeira, onde é possível realizar movimentações financeiras como Despesas (Figura 6.10) e Receitas (Figura 6.11), Fechamento do Caixa Diário (Figura 6.12), Pagamento ao Artesão (Figura 6.13) e Vendas do Bazar (Figura 6.15), que abre o formulário Itens de Venda (Figura 6.16), ou optar pela Movimentação de Produtos, onde é possível realizar as transações com produtos, seja Entrada de Produtos (Figura 6.17), que imprime inclusive o Recibo de Entrega de Produtos (Figura 6.18) ou Devolução de Produtos (Figura 6.19), que imprime o Recibo de Devolução de Produtos (Figura 6.20).

Ao fazer o Fechamento do Caixa Diário, o saldo obtido é automaticamente lançado na Tabela Fechamento como saldo anterior no dia seguinte. Assim como, ao efetivar uma Venda, além de atualizar a Tabela Receita, também é dado baixa na quantidade do produto em estoque e é calculado os valores de repasse ao Bazar e ao Artesão, para posterior pagamento.

No Formulário de Pagamento ao Artesão (Figura 6.13), além de selecionar o artesão e o período desejado, é possível visualizar o relatório (Figura 6.14), por meio do botão Relatório, ou efetivar o pagamento, pelo botão Pagamento.

Ao movimentar produtos, seja entrada ou devolução, a quantidade de produtos em estoque é automaticamente atualizada, assim como a tabela que controla os itens fornecidos, mantendo a integridade dos dados tanto do estoque quanto dos itens fornecidos pelos artesãos.



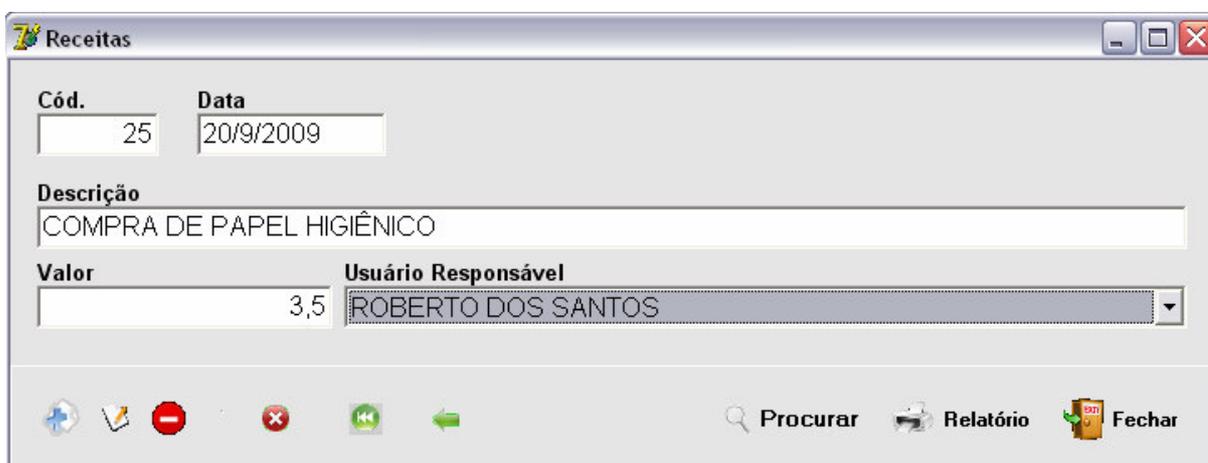
Cód.	Data
2	19/9/2009

Descrição
COMPRA DE 2 CANETAS

Valor	Usuário Responsável
5	PAMELA CRISTINA SOUZA

Procurar Relatório Fechar

Figura 6.10 – Formulário de Despesas do Bazar



Cód.	Data
25	20/9/2009

Descrição
COMPRA DE PAPEL HIGIÊNICO

Valor	Usuário Responsável
3,5	ROBERTO DOS SANTOS

Procurar Relatório Fechar

Figura 6.11 – Formulário de Receitas do Bazar

Fechamento de Caixa

Selecione a data para fechamento

18/9 /2009

Confirma data

Data selecionada:
18/09/2009

Selecione data do dia útil seguinte

19/9 /2009

Usuário responsável

PAMELA CRISTINA SOUZA

Fechamento

Figura 6.12 – Formulário de Fechamento do Caixa Diário

Formulário de Seleção de Artesão e Período

Selecione o Artesão: FERNANDA MONTEIRO

Data Inicial: 1/7 /2009

Data Final: 31/7 /2009

Relatório **Pagamento**

Figura 6.13 – Formulário de Pagamento ao Artesão

Bazar do Sagrado Coração de Jesus

Recibo de Repasse ao Artesão

ARTESÃO: CARLOS ANTONIO DE MORAES
Código: 4
Período: 05/05/2009 à 21/09/2009

Data	Produto	Vr.Unit.	Quant.	SubTotal	Rep. Bazar	Total
20/9/2009	PANO DEPRATO	R\$ 5,50	2	R\$ 11,00	R\$ 1,10	R\$ 9,90
20/9/2009	PANO DEPRATO	R\$ 5,50	1	R\$ 5,50	R\$ 0,55	R\$ 4,95
20/9/2009	PANO DEPRATO	R\$ 5,50	1	R\$ 5,50	R\$ 0,55	R\$ 4,95
20/9/2009	PANO DEPRATO	R\$ 5,50	3	R\$ 16,50	R\$ 1,65	R\$ 14,85

Somas:

SubTotal: R\$ 38,50

Repasse Bazar: R\$ 3,85

Total: R\$ 34,65

Figura 6.14 – Recibo de Pagamento ao Artesão

Venda de Produtos

Cód. Venda: **Data Venda:**

Responsável pela Venda:

Total da Venda:

Gravar Venda

Itens de Venda

Procurar

Relação

Fechar

Figura 6.15 – Formulário de Vendas

Itens de Venda

Cód. Venda: Data da Venda: Total da Venda:

Cód. Item Fornecido: Descrição Produto:

Valor Unitário: Qtde. Vendida: Total:

Usuário Responsável:

Repasse ao Artesao: Repasse ao Bazar:

Cód. Venda	Cód. Item Forn	Descrição do Produto	Qtde. Vendida	Total
3	2	COLAR	1	R\$ 5,00
3	1	PANO DE PRATO	2	R\$ 15,00

Inserir Gravar Gravar Alteração Relação Editar Excluir Fechar

Figura 6.16 – Formulário de Itens de Venda

Entrada de Produtos

Cód. Item Fornecido: Data Fornecimento:

Descrição do Produto:

Nome Artesão:

Descrição do Item:

Valor Unitário: Quant. Fornecida: Recebido por:

CodItemForn	CodArtesao	CodProd	DescricaoItem	QtdeItem	ValorItem
1	2	1	PINTADO E BORDADO	13	
2	1	6	PLASTICO	1	
4	3	5	COLORIDA	25	
5	4	1	PINTADO	10	

Procurar Imprimir Recibo Imprimir Etiquetas Fechar

Figura 6.17 – Formulário de Entrada de Produtos

Bazar do Sagrado Coração de Jesus

Recibo de Produtos

Artesão: JOANA APARECIDA SOUZA		Cód: 2
Endereço: TRAVESSA MARQUES DO HERVAL		Bairro: CENTRO
Fone: (12)3644-0000		Celular:
RG: 58.555.963-3	CPF: 111.125.165/42	PAT. SUTACO: 8956
Período: 01/08/2009 à 31/08/2009		

Quantidade Produto	Descrição	ValorUnitário
35	CAIXA DE PRESENTE DECORADA CAIXA DE PAPELÃO	R\$ 3,50

Figura 6.18 – Recibo de Entrega de Produtos

Devolução de Produtos

Código	Cod. Art.	Artesão
<input type="text" value="4"/>	<input type="text" value="2"/>	<input type="text" value="LETICIA TERESA"/>
Data Devolução	Cod. Item Forn	Cod.Prod.
<input type="text" value="19/9/2009"/>	<input type="text" value="1"/>	<input type="text" value="1"/>
Descrição Produto		
<input type="text" value="PANO DE PRATO"/>		
Qtde. Devolvida	Usuário Resp.	
<input type="text" value="2"/>	<input type="text" value="PAMELA CRISTINA SOUZA"/>	

Figura 6.19 – Formulário de Devolução de Produtos

Bazar do Sagrado Coração de Jesus
Recibo de Devolução de Produtos

Artesão: JOANA APARECIDA SOUZA		Cód: 2
Endereço: TRAVESSA MARQUES DO HERVAL		Bairro: CENTRO
Fone: (12)3644-0000	Celular:	
RG: 56.555.983-3	CPF: 111.125.165/42	PAT. SUTACO: 8956

Período: 01/08/2009 à 31/08/2009

Cód.Dev.	Dt. Dev.	Cód.Item	Quant.	Produto
1	24/8/2009	2	5	CAIXA DE PRESENTE DECORADA

Figura 6.20 – Recibo de Devolução de Produtos

O módulo Relatórios permite visualizar e imprimir relatórios de Estoque, seja para controle de permanência dos produtos em estoque (Figura 6.21) ou para simples conferência (Figura 6.22), relatórios Financeiros, como Relatório de Caixa Diário (Figura 6.23) e Resumo Mensal de Fluxo de Caixa (Figura 6.24), que mostra um Resumo do Movimento do Caixa no mês determinado pelo usuário e, por fim, Relatório de Repasse ao Bazar (Figura 6.25) pelo período determinado pelo usuário.

Bazar do Sagrado Coração de Jesus
Produtos em Estoque há mais de três meses

Cód.Art.	Nome Artesão	Fone	Celular	Cód.It.Forn.	Descrição	Data Forn.	Qtde.
4	CARLOS ANTONIO DEMORAES	(12)3654-6625		7	PANO DEPRATO	20/5/2009	10
2	LETICA TERESA	(12)3879-6225	(12)9829-3827	8	PULSERA	4/3/2009	20
2	LETICA TERESA	(12)3879-6225	(12)9829-3827	9	COLAR	4/6/2009	25

Figura 6.21 – Relatório de Produtos com tempo de permanência vencido

Bazar do Sagrado Coração de Jesus		
<i>Produtos em Estoque</i>		
Código:	Descrição:	Quantidade:
3	CAIXA DE PRESENTE DECORADA	83
2	PANO DE PRATO	110
1	TRUFAS	298
Total Produtos em Estoque:		491

Figura 6.22 – Relatório de Produtos em Estoque

Bazar do Sagrado Coração de Jesus				
<i>Caixa Diário</i>				
Data: 27/09/2009				
Cód.	Data	Tipo	Descrição	Valor
28	27/9/2009	R	VENDA DE PRODUTOS	R\$ 100,00
4	27/9/2009	D	COMPRA DE APONTADOR	R\$ 3,50
7	27/9/2009	D	REP. AO ART. LETICIA TERESA - 20/09	R\$ 52,50
Saldo Anterior: R\$		1.824,00		
Total Entrada: R\$		100,00		
Total Saída: R\$		56,00		
Saldo: R\$		1.868,00		

Figura 6.23 – Relatório de Caixa Diário

Bazar do Sagrado Coração de Jesus	
<i>Resumo Mensal de Caixa</i>	
Mês: 01/09/2009 à 30/09/2009	
Saldo Anterior: R\$	0,00
Rep. Artesão Ant: R\$	0,00
Total Entrada: R\$	2.062,50
Total Saída: R\$	65,00
Saldo: R\$	1.997,50
Saldo Atual: R\$	1.997,50
Rep. Artesão: R\$	73,10
Saldo Real: R\$	1.924,40

Figura 6.24 – Resumo Mensal de Fluxo de Caixa

Bazar do Sagrado Coração de Jesus						
<i>Recibo de Repasse ao Bazar</i>						
Período: 05/05/2009 à 31/08/2009						
Data	Produto	Vr.Unit.	Quant.	SubTotal	Rep.Artesão	Total
24/8/2009	CAIXA DE PRESENTE DECORADA	R\$ 3,50	2	R\$ 7,00	R\$ 7,00	R\$ 0,00
24/8/2009	TRUFAS	R\$ 2,50	2	R\$ 5,00	R\$ 5,00	R\$ 0,00
<u>Somas:</u>				SubTotal: R\$	12,00	
				Repasse Artesão: R\$	12,00	
				Total: R\$	0,00	

Figura 6.25 – Relatório de Repasse ao Bazar

No último módulo, denominado Ajuda, pode-se realizar a rotina de *Backup* do banco de dados selecionando-se o diretório de destino no Formulário de *Backup* (Figura 6.26), acessar o Manual de Orientação ao Usuário, cuja capa pode ser visualizada na Figura 6.27 e podem ser obtidas informações sobre o sistema, como

versão, desenvolvedor e base de dados, conforme pode ser observado na Figura 6.28.

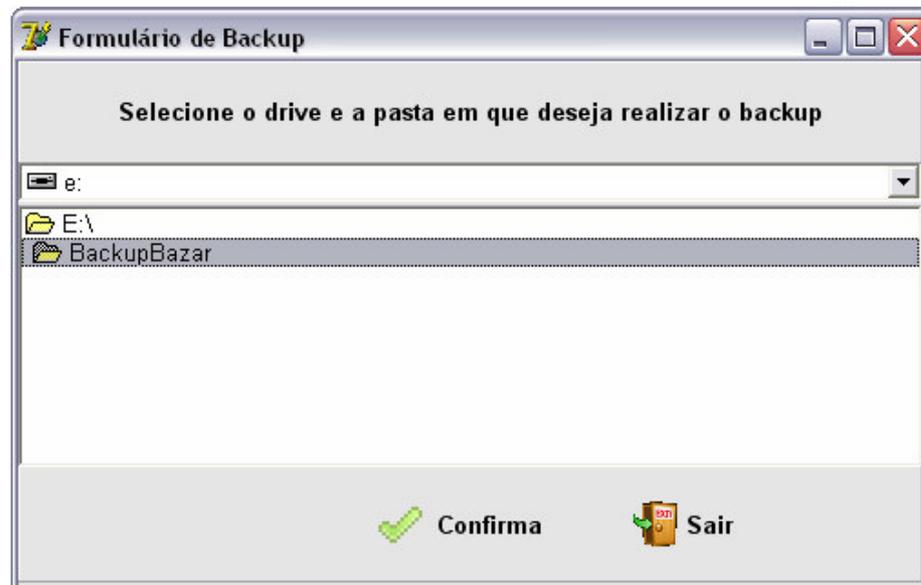


Figura 6.26 – Formulário de *Backup*

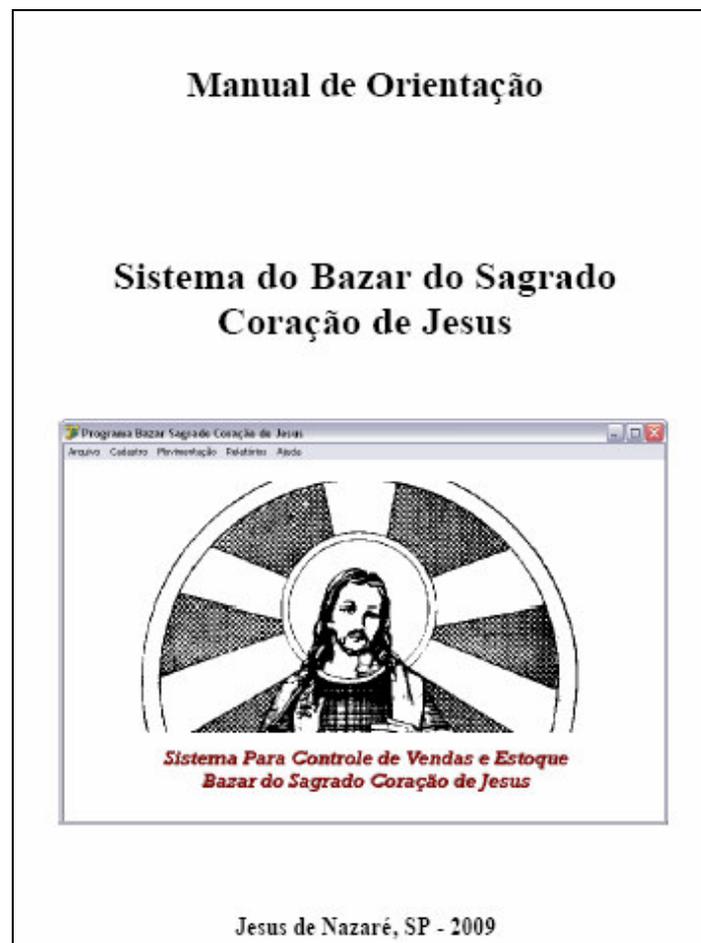


Figura 6.27 – Capa do Manual de Orientação ao Usuário



Figura 6.28 – Formulário de Informações do Sistema

7 CONCLUSÕES E CONSIDERAÇÕES FINAIS

Os *softwares* são utilizados para gerar soluções e agregar facilidades para todas as empresas, sejam elas de pequeno, médio ou grande porte, com ou sem fins lucrativos, criando comodidades para seus usuários nas mais diferentes áreas: educação, lazer, comercial, dentre outras.

O sistema SCVE-BSCJ foi criado para atender a pequena empresa, sem fins lucrativos, denominada, neste trabalho, Bazar do Sagrado Coração de Jesus, auxiliando-a no conhecimento e controle de seus estoques e de seus movimentos financeiros.

O desenvolvimento deste sistema permitiu a participação na construção de um *software* em todas as suas fases, desde o levantamento de requisitos até a fase de testes, garantindo a aplicação prática do que foi aprendido, de forma teórica, no período acadêmico.

Um fator importante foi a possibilidade de trabalhar em equipe, habilidade muito exigida no mercado de trabalho para qualquer área, mas principalmente na área de informática, onde é necessário relacionar-se com clientes e usuários, de forma a extrair deles os requisitos necessários para se obter um *software* de qualidade.

O Bazar do Sagrado Coração de Jesus, antes do sistema desenvolvido, utilizava planilhas de Excel e fichas de papel para realizar, precariamente, seus controles de estoque e cadastros, o que gerava um grande acúmulo de papéis a serem arquivados, além de dados distorcidos da realidade e inexistência de um controle da movimentação financeira do bazar.

Com o sistema desenvolvido, todos os controles realizados em planilhas e fichas de papel foram transformados para serem realizados por computador, facilitando tanto a organização quanto à agilidade e precisão de dados para realizar cadastros, gerar relatórios, verificar situação de estoque e financeira e outras opções oferecidas pelo sistema. Em resumo, o sistema traz facilidades tanto para o bazar quanto para os clientes.

Como trabalho futuro, propõe-se a reestruturação deste *software*, para que possa ser utilizado em rede e a adição de novas funcionalidades, como controle de contas a pagar e a receber e controle de estoque mínimo e máximo, para que possa ser utilizado por empresas comerciais de fins lucrativos. Sugere-se, também, a implementação de funcionalidades, como impressão de etiquetas com código de barras e relatórios de fluxo de caixa, que não foram implementadas no protótipo apresentado.

REFERÊNCIAS BIBLIOGRÁFICAS

BDE Administrator. Version 5.01. [S.l]: Inprise Corporation, 1998. Parte do produto Borland Delphi Enterprise. 1 CD-ROM.

BECK, Kent. **Extreme Programming Explained: Embrace Change**. Boston: Addison Wesley, 1999. Prévia do livro disponível em: <http://books.google.com.br/books?id=G8EL4H4vf7UC&printsec=frontcover&source=gbs_v2_summary_r&cad=0#v=onepage&q=&f=false>. Acesso em: 10 ago. 2009 às 16:40h.

BONA, Cristina. **Avaliação de processos de software**: Um estudo de caso em XP e Iconix. Dissertação (Mestrado em Engenharia de Produção), Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, SC, 2002. Disponível em: <<http://trac.itapirunet.com.br/browser/MONO-PONTES/Monografia-Pontes/matRef/Monografias/10816.pdf?rev=740&format=raw>>. Acesso em: 12 ago. 2009 às 16:36h.

BORLAND Delphi Enterprise. Version 7.0 [S.l]: Borland Software Corporation, 2002. 1 CD-ROM.

CANTU, Marco. **Dominando o Delphi 2005: a Bíblia**. São Paulo: Pearson Prentice Hall, 2006.

CASE Studio 2, versão 2.25.0. [S.l]: Quest Software, Inc., 2006. Disponível em: <www.casestudio.com>. Acesso em: 29 jan. 2009 às 13:46h.

DATE, Christopher J. **Introdução a Sistemas de Bancos de Dados**. 6. ed. Rio de Janeiro: Elsevier Editora Ltda., 2004.

EXTREME. **Extreme Programming**. Don Wells. Disponível em: <<http://www.extremeprogramming.org/index.html>>. Acesso em: 21 jul. 2009 às 07:20h.

GARCIA, Carlos A. **Universidade delphi**. São Paulo: Digerati Books, 2005.

KUHN, Giovane Roslindo; PAMPLONA, Vitor Fernando. **Apresentando XP**: Encante seus clientes com *Extreme Programming*. Disponível em: <<http://javafree.uol.com.br/artigo/871447/Apresentando-XP-Encante-seus-clientes-com-Extreme-Programming.html>>. Acesso em: 15 mar. 2009 às 15:36h.

MANZANO, José Augusto N. G.; MENDES, Sandro Santa Vicca. **Estudo Dirigido de Delphi 7**. 3. ed. São Paulo: Érica Ltda., 2006.

MARTINEZ, Maria Laura. **Um método de Web Design baseado em usabilidade.** In: 16º SIMPÓSIO NACIONAL DE GEOMETRIA DESCRITIVA E DESENHO TÉCNICO - V INTERNATIONAL CONFERENCE ON GRAPHICS ENGINEERING FOR ARTS AND DESIGN, 2003, Santa Cruz do Sul, RS. Disponível em: <http://www.lsi.usp.br/~martinez/works/_artigos/martinez03a.pdf> Acesso em: 22 mar. 2009.

MENESES, Prof. Francisco Gerson A. de. **Banco de Dados – Unidade III – Diagrama Entidade-Relacionamento.** Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Parnaíba. Disponível em: <http://www.cefetparnaiba.edu.br/index.php?option=com_docman&task=doc_view&gid=213&tmpl=component&format=raw&Itemid=79>. Acesso em: 19 ago. 2009 às 22:13h.

MICROSOFT Office Access 2003. Version 11.0. [S.l]: Microsoft Corporation, 2003. Parte do produto *Microsoft Office Professional* Edição 2003. 1 CD-ROM.

MICROSOFT. Microsoft Office Online: Treinamento. Disponível em: <<http://office.microsoft.com/training/training.aspx?AssetID=RC061181381046>>. Acesso em: 24 mar. 2009 às 14:40h.

NOGUEIRA, Carlos Eduardo; FERRETI, Eliana. **Modelos de processo de desenvolvimento de software: abordagens voltadas à internet.** 2005. Monografia (Graduação em Computação Bacharelado) - Departamento de Informática, Universidade de Taubaté, 2005

PRESSMAN, Roger S. **Engenharia de Software.** São Paulo: Pearson Makron Books, 1995.

RATIONAL. IBM: Software. Disponível em: <<http://www-01.ibm.com/software/br/rational/rup.shtml>>. Acesso em: 29 abr. 2009 às 10:11h.

SANTOS, Venícios Gustavo; DAHER, Professor Nesley. **Utilização de storytelling como ferramenta de aquisição de requisitos em processo de desenvolvimento de software apoiados em modelos ágeis: o uso apoiado no Extreme Programming.** 2008. Revista Científica do Departamento de Tecnologia do Uni-BH, Belo Horizonte, v.1, n.1, nov 2008. Disponível em: <<http://docs.google.com/gview?a=v&q=cache:PXRmCCxErQEJ:site1.unibh.br/imgMarketing/revistas/dtec/include/getdoc.php%3Fid%3D46%26article%3D2%26mode%3Dpdf+storytelling+extreme+programming&hl=pt-BR&gl=br>>. Acesso em: 12 ago. 2009 às 20:30h.

SILVA, Camila C.; PAULA, Everaldo A. **Delphi 7 Desvendando os Mistérios.** Santa Cruz do Rio Pardo: Viena, 2007.

SOMMERVILLE, Ian. **Engenharia de Software.** 6. ed. São Paulo: Pearson Addison Wesley, 2003.

UNESC. **Diagrama de Fluxo de Dados.** Curso de Ciência da Computação, Análise e Projeto de Sistemas I, Universidade do Extremo Sul Catarinense, SC. Disponível em: < http://www.ead.unesc.net/resources/files/615/diagrama_fluxo_dados.doc>. Acesso em: 17 ago. 2009 às 17:20h.

UNITRI. **Capítulo 1: O BDE Administrador e os SQL Links.** Uberlândia: Centro Universitário do Triângulo. Disponível em: <<http://www.computacao.unitri.edu.br/downloads/professores/DelphiBDModulo01.pdf>>. Acesso em: 14 mai. 2009 às 16:10h.

REFERÊNCIAS BIBLIOGRÁFICAS COMPLEMENTARES

PONTES, Artur Moltocar. **Influência do processo de desenvolvimento sobre prazo e custo de construção de *software***. 15º CONGRESSO DE INICIAÇÃO CIENTÍFICA. Disponível em: <<http://www.unimep.br/phpg/mostraacademica/anais/5mostra/1/202.pdf>>. Acesso em: 14 mai. 2009 às 16:10h.

TELES, Vinícius Manhães. **Um estudo de caso da adoção das práticas e valores do *Extreme Programming***. 2005. Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, 2005. Disponível em: <<http://improveit.com.br/xp/dissertacaoXP.pdf>>. Acesso em: 12 ago. 2009 às 20:19h.