

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

APLICAÇÃO COMERCIAL PARA CELULARES BASEADA
EM M-COMMERCE UTILIZANDO J2ME

DAIANA MARIA SEDREZ

BLUMENAU
2006

2006/1-07

DAIANA MARIA SEDREZ

**APLICAÇÃO COMERCIAL PARA CELULARES BASEADA
EM M-COMMERCE UTILIZANDO J2ME**

Trabalho de Conclusão de Curso submetido à
Universidade Regional de Blumenau para a
obtenção dos créditos na disciplina Trabalho
de Conclusão de Curso II do curso de Sistemas
de Informação — Bacharelado.

Prof. Marcel Hugo, M. Eng. – Orientador

**BLUMENAU
2006**

2006/1-07

APLICAÇÃO COMERCIAL PARA CELULARES BASEADA EM M-COMMERCE UTILIZANDO J2ME

Por

DAIANA MARIA SEDREZ

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente: _____
Prof. Marcel Hugo, M. Eng. – Orientador, FURB

Membro: _____
Prof. Francisco Adell Péricas, Mestre – FURB

Membro: _____
Prof. Everaldo Artur Grahl, Titulação – FURB

Blumenau, 11 de julho de 2006.

Dedico este trabalho aos meus pais, aos meus familiares e amigos que sempre me apoiaram, ao meu namorado Babbiton por me compreender e confiar em mim, os quais foram fundamentais para a realização do mesmo.

AGRADECIMENTOS

Agradeço a Deus pela vida e pela minha vida.

Aos meus pais Irineu e Edit, e às minhas irmãs Deisi e Daniela, que sempre me apoiaram, incentivaram a estudar e seguir em frente.

Aos meus sobrinhos Bruno e Letícia que representam felicidade na minha vida.

Ao meu namorado Babbiton, por estar na minha vida, por me fazer amar, por me compreender e me ensinar a compreender.

Ao Fabiano Rosa, por ter me auxiliado com seu conhecimento, o qual foi de grande importância para a conclusão do trabalho.

Ao meu orientador, Marcel Hugo, pela ajuda e dedicação para que a conclusão deste trabalho tivesse sucesso.

“Se podemos sonhar, também podemos tornar
nossos sonhos realidade.”

Walt Disney

RESUMO

Este trabalho apresenta o desenvolvimento de uma aplicação utilizando o conceito de M-Commerce, ou seja, o comércio móvel, possibilitando realizar a compra de um produto através de um celular. A utilização da plataforma Java, da tecnologia J2ME e de Web Services foram os componentes necessários para a implementação deste sistema. O sistema foi desenvolvido utilizando a ferramenta NetBeans, com o módulo NetBeans Mobility Pack na implementação da aplicação J2ME.

Palavras-chave: M-Commerce. Comércio móvel. J2ME. Celular.

ABSTRACT

This article has as objective the development of an application using M-Commerce concept, that is, the mobile commerce, making possible to carry through the purchase of a product through a cellular. Using Java platform, technology J2ME and Web Services it had been the necessary components for the implementation. The system was developed using the NetBeans tool, with the module NetBeans Mobility Pack in implementation of J2ME application.

Key-words: M-Commerce. Mobile commerce. J2ME. Cellular

LISTA DE ILUSTRAÇÕES

Figura 1 – Universo J2ME	25
Figura 2 – Estrutura de uma mensagem SOAP	30
Figura 3 – Comunicação entre módulos da aplicação M-Commerce.....	33
Quadro 1 – Requisitos funcionais.....	35
Quadro 2 – Requisitos não funcionais	35
Figura 4 – Diagrama de pacotes	37
Figura 5 – Diagrama de caso de uso módulo central.....	38
Figura 6 – Diagrama de caso de uso módulo MIDP comprador	39
Figura 7 – Diagrama de atividades do caso de uso “Identificar usuário”.....	40
Figura 8 – Diagrama de atividades do caso de uso “Cadastrar consumidor”	40
Figura 9 – Diagrama de atividades dos casos de uso “Identificar consumidor” e “Efetuar compra”	41
Figura 10 – Diagrama de classes	43
Figura 11 – Diagrama físico de dados	44
Figura 12 – Diagrama de seqüência do caso de uso “Cadastrar consumidor”	45
Figura 13 – Diagrama de seqüência do caso de uso “Selecionar produtos para venda”	46
Figura 14 – Diagrama de seqüência dos casos de uso “Identificar consumidor” e “Efetuar compra”	47
Figura 15 – Fluxo NetBeans Mobility Pack	51
Figura 16 – Tela de login.....	56
Figura 17 – Tela de cadastro de usuário	57
Figura 18 – Tela de cadastro de forma de pagamento.....	58
Figura 19 – Tela de cadastro de tipo de produto	59
Figura 20 – Tela de cadastro de produto	60
Figura 21 – Tela de cadastro de cidade	61
Figura 22 – Tela de cadastro de local.....	62
Figura 23 – Tela de cadastro de ponto de venda	63
Figura 24 – Tela de cadastro de produto no ponto de venda.....	64
Figura 25 – Tela inicial.....	65
Figura 26 – Tela de identificação do consumidor	66
Figura 27 – Tela de identificação do ponto de venda.....	66

Figura 28 – Tela de aviso para conferir ponto de venda	67
Figura 29 – Tela para conferir ponto de venda.....	67
Figura 30 – Tela para informar produto e quantidade	68
Figura 31 – Tela para escolher forma de pagamento.....	69
Figura 32 – Tela para confirmar a compra	69
Figura 33 – Tela de mensagem de finalizando a compra	70
Figura 34 – Tela de confirmação da compra	70
Figura 35 – Tela de e-mail recebido pelo ponto de venda.....	71
Quadro 3 – Funcionalidades de cada trabalho.....	73
Quadro 4 – UC01.01 Identificar usuário	79
Quadro 5 – UC01.02 Cadastrar consumidor	80
Quadro 6 – UC01.03 Cadastrar gerente / vendedor.....	81
Quadro 7 – UC01.04 Cadastrar formas de pagamento.....	82
Quadro 8 – UC01.05 Cadastrar tipos de produto	83
Quadro 9 – UC01.06 Cadastrar produtos para venda.....	84
Quadro 10 – UC01.07 Cadastrar cidade para o local de ponto de venda.....	85
Quadro 11 – UC01.08 Cadastrar local para ponto de venda	86
Quadro 12 – UC01.09 Cadastrar pontos de venda	87
Quadro 13 – UC01.10 Selecionar produtos para venda	88
Quadro 14 – UC02.01 Identificar consumidor	89
Quadro 15 – UC02.02 Efetuar compra.....	89

LISTA DE SIGLAS

Anatel – Agência Nacional de Telecomunicações

APIs – *Application Programming Interface*

B2B – *Business-to-Business*

B2C – *Business-to-Consumer*

B2E – *Business-to-Employee*

C2C – *Consumer-to-Consumer*

CDC – *Connected Device Configuration*

CLDC – *Connected Limited Device Configuration*

CTBC – Companhia de Telecomunicações do Brasil Central

E-Commerce – *Eletronic-Commerce*

E-M-Commerce – *Eletronic-Mobile-Commerce*

EA – *Enterprise Architect*

GPS – *Global Positioning System*

HTTP – *Hiper Text Transfer Protocol*

HTTPS – *Hyper Text Transfer Protocol Secure*

J2EE – *Java 2 Enterprise Edition*

J2ME – *Java 2 Micro Edition*

J2SE – *Java 2 Standard Edition*

JAR – *Java Archive*

JVM – *Java Virtual Machine*

JSP – *Java Server Pages*

KVM – *Kilobyte Virtual Machine*

M-Commerce – *Mobile-Commerce*

MID – Mobile Information Device

MIDP – Mobile Information Device Profile

MP3 – Mpeg Layer 3

MVC – Model View Controller

PDA – Personal Digital Assistant

RMS – Record Management System

RPC – Remote Procedure Calls

SMS – Short Message Service

SOAP – Simple Object Access Protocol

SSL – Secure Sockets Layer

T-Commerce – TV-Commerce

TLS – Transport Layer Security

UML – Unified Modeling Language

XML – Extensible Markup Language

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 JUSTIFICATIVA	16
1.2 OBJETIVOS DO TRABALHO	16
1.3 ESTRUTURA DO TRABALHO	17
2 FUNDAMENTAÇÃO TEÓRICA	18
2.1 COMÉRCIO ELETRÔNICO	18
2.2 M-COMMERCE	20
2.3 DISPOSITIVOS MÓVEIS	22
2.4 JAVA	24
2.4.1 J2ME	25
2.4.1.1 Configuração CLDC	26
2.4.1.2 Perfil MIDP	27
2.4.2 J2EE	28
2.5 WEB SERVICES	28
2.5.1 SOAP	29
2.6 TRABALHOS CORRELATOS	31
3 DESENVOLVIMENTO DO TRABALHO	33
3.1 REQUISITOS PRINCIPAIS DO SISTEMA	34
3.1.1 Requisitos funcionais	34
3.1.2 Requisitos não funcionais	35
3.2 ESPECIFICAÇÃO	36
3.2.1 Casos de uso do sistema.....	36
3.2.2 Diagramas de atividades do sistema	39
3.2.3 Diagramas de classes do sistema.....	41
3.2.4 Diagrama físico de dados	44
3.2.5 Realizações de casos de uso.....	44
3.2.6 Ferramenta utilizada para a especificação.....	48
3.3 IMPLEMENTAÇÃO	48
3.3.1 Técnicas e ferramentas utilizadas.....	49
3.3.1.1 NetBeans IDE	49
3.3.1.2 NetBeans Mobility Pack	49

3.3.1.3 Sun Java Wireless Toolkit	52
3.3.1.4 Biblioteca kSOAP2 e kXML	52
3.3.1.5 Tomcat	52
3.3.1.6 Hibernate.....	52
3.3.1.7 Apache Struts.....	53
3.3.1.8 Padrões de projeto implementados	54
3.3.1.8.1 Padrão MVC	54
3.3.1.8.2 Padrão <i>Facade</i> e <i>Singleton</i>	54
3.3.1.9 Segurança na comunicação com o Web Service	55
3.3.2 Operacionalidade da implementação	55
3.3.2.1 Módulo central.....	55
3.3.2.2 Módulo MIDP comprador	64
3.4 RESULTADOS E DISCUSSÃO	71
4 CONCLUSÕES.....	74
4.1 EXTENSÕES	74
REFERÊNCIAS BIBLIOGRÁFICAS	76
APÊNDICE A – Especificação dos casos de uso.....	79
APÊNDICE B – Utilização da biblioteca kSOAP2.....	90
APÊNDICE C – Arquivo de mapeamento do Hibernate.....	91
APÊNDICE D – Padrão <i>Facade</i> e <i>Singleton</i>.....	93
APÊNDICE E – Código gerado pelo módulo NetBeans Mobility Pack	98

1 INTRODUÇÃO

Mundialmente a telefonia celular tornou-se popular tanto no ambiente de trabalho quanto na vida privada. Segundo Alencar (2006, p. 1), a base de celulares no Brasil alcançou a marca de 89,4 milhões em março de 2006. Conforme a Agência Nacional de Telecomunicações (Anatel), o mês de março de 2006 foi o melhor desde o ano de 1999, registrando-se 1,3 milhões de novas habilitações, tendo os últimos 12 meses um crescimento de 30% comparado ao período anterior, pois o país ganhou 18 milhões de novos assinantes. De acordo com Teixeira (2005, p. 1), há dois bilhões de celulares no mundo, marca concedida em setembro de 2005. No Brasil o celular já faz parte do dia-a-dia dos brasileiros, onde 23 milhões de brasileiros pretendem comprar o primeiro celular e 10 milhões pretendem trocar seu aparelho por um novo (Rydlewski, 2004, p. 102).

Hoje o aparelho celular é bastante diferente de 10 anos atrás, conforme Menezes (2003, p. 2), pois além de celular é também máquina fotográfica, *Mpeg Layer 3 (MP3) Player*, *Personal Digital Assistant (PDA)*, rádio, internet, executor de aplicativos. É tendência natural da evolução do celular acrescentar a essas funcionalidades um cartão de crédito eletrônico, para que se faça presente o *Mobile-Commerce (M-Commerce)*.

De acordo com Moskorz (2002, p. 12), *M-Commerce* é a sigla para *Mobile-Commerce*. Numa tradução livre, poder-se-ia chamar de Comércio Móvel, onde na verdade, a própria sigla é uma derivada do *Eletronic-Commerce (E-Commerce)*, ou comércio eletrônico. De certa forma, o termo mais correto seria *Eletronic-Mobile-Commerce (E-M-Commerce)*. Então, uma melhor tradução seria “Comércio Eletrônico Móvel”. Segundo Observatório do Comércio (2000, p. 1), *M-Commerce* pode ser definido como a venda de produtos e serviços ao consumidor através de aparelhos móveis (Observatório do Comércio, 2000, p. 1).

Conforme Adam Zawell (2005, apud Reuters, 2005, p. 1) do Yankee Group, a previsão

de transações de comércio eletrônico móvel é de somar US\$ 1,2 bilhão na Europa em 2009, registrando alta em relação aos US\$ 243 milhões registrados em 2004. Enquanto isso, as transações na Ásia podem crescer para US\$ 1,7 bilhão daqui a quatro anos, partindo de um patamar de US\$ 370 milhões registrados no ano passado. “Se a tecnologia chegar a um ponto em que seja mais fácil usar o celular do que tirar dinheiro da carteira para pagar por um produto, então os consumidores vão adotá-la”.

Segundo Almeida (2004, p. 21), para a execução de aplicativos dentro dos celulares, é necessário algo para desenvolver os aplicativos. A tecnologia Java possui a edição *Java 2 Micro Edition* (J2ME) para pequenos dispositivos como *paggers*, telefones celulares, *set-top boxes* de TVs a cabo, PDA, sendo uma versão específica da máquina virtual criada para ser executada em um ambiente com recursos limitados de memória e processamento. Os desenvolvedores são livres para criar aplicações e executá-las em qualquer dispositivo de qualquer fabricante que possua uma máquina virtual, não sendo necessário se prender a um dos fabricantes ou a uma tecnologia. Também é livre a escolha do modelo de telefone celular que possa executar o aplicativo desejado.

Conforme Montenegro e Pereira (2005, p. 29), na indústria de dispositivos de consumo, a área de celulares é onde o J2ME tem recebido mais investimentos. Atualmente existem mais de 350 modelos de *handsets* Java de 31 fabricantes. No final do ano de 2004, existiam mais de 570 milhões de aparelhos com suporte à J2ME. Antes do J2ME, para executar uma aplicação em um celular, ele deveria ser escrito na linguagem nativa do próprio dispositivo. O processo de conversão do aplicativo de um dispositivo para outro era potencialmente complexo, com custos que poderiam inviabilizá-lo.

Portanto, neste trabalho foi realizada a construção de um aplicativo para celular para compra de produtos e serviços oferecidos para venda, utilizando o conceito de *M-Commerce* e os recursos disponibilizados pela tecnologia J2ME. Para a troca de informações entre os

aplicativos dos celulares que foram desenvolvidas em J2ME e a aplicação servidora que foi desenvolvida em *Java 2 Enterprise Edition* (J2EE), e vice-versa, foram utilizados *Web Services* em um servidor de aplicações, utilizando *Hiper Text Transfer Protocol* (HTTP) e *Simple Object Access Protocol* (SOAP).

1.1 JUSTIFICATIVA

O aspecto principal identificado para desenvolver este trabalho é a utilização de um conceito novo, chamado *M-Commerce*. Conforme Reuters (2005, p. 1), as companhias norte-americanas têm mantido silêncio sobre o comércio via telefone celular desde que a bolha das empresas de tecnologia estourou em 2000. Segundo Chris Bierbaum (2005, apud Reuters, 2005, p. 1), executivo de desenvolvimento de negócios da Sprint, terceira maior operadora dos EUA, “a discussão (sobre comércio móvel) tem mais validade agora. A tecnologia e os modelos de negócios estão evoluindo”.

O potencial de negócios do *M-Commerce* é imenso, pois os consumidores possuem cada vez menos tempo para dispensar às suas compras, preferindo soluções que oferecem comodidade, praticidade e horários adequados aos seus estilos de vida, podendo comprar um bem ou serviço a partir de qualquer lugar.

Dessa forma o trabalho estará contribuindo para mais uma fonte de pesquisa sobre *M-Commerce*, juntamente com uma aplicação utilizando o conceito.

1.2 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é desenvolver uma aplicação que permita a compra de produtos pelo celular, utilizando a tecnologia J2ME e o conceito de *M-Commerce*.

Os objetivos específicos do trabalho são:

- a) oferecer uma aplicação servidora que disponibilizará uma lista de produtos / serviços que serão vendidos (módulo central);
- b) executar a aplicação no aparelho celular do consumidor (módulo *Mobile Information Device Profile* (MIDP) comprador) que irá interagir com a aplicação servidora (módulo central);
- c) efetuar o processo de compra e venda através da aplicação servidora (módulo central) e da aplicação no aparelho celular do ponto de venda (módulo MIDP vendedor).

1.3 ESTRUTURA DO TRABALHO

Este trabalho está dividido em quatro capítulos. O primeiro capítulo é composto pela introdução do trabalho, os objetivos a serem alcançados com o desenvolvimento do trabalho e sua estrutura. O segundo capítulo apresenta a fundamentação teórica do trabalho, contextualizando o comércio eletrônico, o conceito de *M-Commerce*, a utilização dos dispositivos móveis, a tecnologia Java, os recursos do J2EE e J2ME e a utilização de *Web Services*. O terceiro capítulo aborda o desenvolvimento do trabalho, apresentando os requisitos do sistema, a especificação, como os diagramas de casos de uso, diagramas de atividade, diagramas de classe, diagrama físico de dados, diagramas de realização do caso de uso, ferramenta de modelagem utilizada na especificação, técnicas e ferramentas utilizadas na implementação do sistema, a operacionalidade da implementação passo a passo e resultados e problemas encontrados durante o desenvolvimento do trabalho. Por último, o capítulo quatro descreve as considerações finais sobre o trabalho, incluindo as extensões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica traz o conceito dos itens abordados no trabalho e está dividida nos tópicos seguintes:

- a) comércio eletrônico;
- b) *M-Commerce*;
- c) dispositivos móveis;
- d) Java;
- e) *Web Services*;
- f) trabalhos correlatos.

2.1 COMÉRCIO ELETRÔNICO

Segundo Franco Junior (2005, p. 19), a compra de produtos e serviços pela Internet está causando uma revolução bastante grande no mundo dos negócios e na vida dos consumidores. Segundo Alexander, Soares e Viana (2002, p. 15), através do comércio eletrônico é possível fazer várias negociações e transações, com muita rapidez, comodidade e segurança.

Conforme Alexander, Soares e Viana (2002, p. 15), o conceito de *E-Commerce*, como é denominado o comércio eletrônico, significa um conjunto de atividades comerciais que acontecem *online*, ou seja, uma transação comercial. Alguns tipos de *E-Commerce* são *TV-Commerce (T-Commerce)* e *M-Commerce*.

De acordo com Rodrigo (2005, p. 1), o *E-Commerce* faz parte do *E-Business*, pois além da parte de compras e vendas, o *E-Business* também compreende o marketing, a logística, a infra-estrutura e o atendimento ao cliente. Segundo Franco Junior (2005, p. 21), o

E-Business é o negócio eletrônico, ou seja, uma negociação feita pela Internet. A IBM define *E-Business* conforme abaixo.

[...] uma forma segura, flexível e integrada de fornecer um valor diferenciado na gestão administrativa pela combinação de sistemas e processos para a administração e funcionamento de operações centrais, de forma simples e eficiente alavancada pela aplicação de tecnologia da Internet. (FRANCO JUNIOR, 2005, p. 21, grifo do autor).

Segundo Rodrigo (2005, p. 1), o *E-Business* pode ser definido como sistemas de informação que auxiliam os processos de negócio. Além do *E-Commerce* o *E-Business* possui diversos sistemas como o *Business-to-Business* (B2B), *Business-to-Consumer* (B2C), *Business-to-Employee* (B2E), *Consumer-to-Consumer* (C2C).

Todos os tipos possuem algumas finalidades em comum como conquistar a fidelidade de clientes, por exemplo. Abaixo segue a definição de alguns tipos de *E-Business* e *E-Commerce*.

- a) B2B: de acordo com Rodrigo (2005, p. 1), B2B são as transações de comércio entre empresas utilizando a Internet. Conforme Alexander, Soares e Viana (2002, p. 16), é um comércio bastante consolidado entre as empresas, o qual é de fundamental importância para a sobrevivência delas;
- b) B2C: segundo Alexander, Soares e Viana (2002, p. 16), é um comércio entre empresa e consumidor que também utiliza a Internet como meio, envolvendo todos os tipos de venda de produtos, serviços ou informações;
- c) B2E: conforme Alexander, Soares e Viana (2002, p. 17), B2E é classificada como a relação de negócios entre a empresa e seus colaboradores. Esta tecnologia está relacionada aos portais (*intranet*) que as empresas possuem para atender seus colaboradores, possibilitando à empresa uma comunicação direta com eles;
- d) C2C: de acordo com Alexander, Soares e Viana (2002, p. 17), este tipo de comercialização é classificada pela comercialização realizada entre os consumidores. Este comércio oferece aos consumidores serem compradores e/ou

vendedores, podendo oferecer seus produtos através de classificados, leilões eletrônicos;

- e) *T-Commerce*: conforme Alexander, Soares e Viana (2002, p. 18), este é o comércio eletrônico pela televisão, sendo oferecido aos usuários serviços interativos, permitindo a compra de produtos via TV utilizando somente o controle remoto.

2.2 M-COMMERCE

Segundo Dornan (2001, p. 145), o maior interesse comercial por dados móveis e pela *Web* sem fio está direcionado à promessa de um grande e novo mercado de produtos e serviços. As operadoras sem fio esperam ser capazes de fazer com que seus negócios cresçam até mais rapidamente do que o setor como um todo. Entretanto, a receita obtida não será suficiente para as empresas permanecerem no negócio. Acredita-se que a maior parte da receita seja obtida com o comércio eletrônico móvel ou *M-Commerce*. A questão é por que as pessoas escolheriam fazer compra pelo telefone móvel? Muitas pessoas portam seus telefones o tempo todo, mesmo quando estão distantes de uma loja, de um micro conectado à Internet ou de qualquer outro recurso que possa ser utilizado para lhes vender algo. O *M-Commerce* fornece aos comerciantes a capacidade de entrar em contato com os clientes a qualquer hora e de acompanhar seus movimentos no ciberespaço e no mundo real.

De acordo com Prado (2005, p. 1), o negócio de telefonia móvel no futuro será um “pote de ouro” pois os celulares terão muitas utilidades e muitos serviços. Um exemplo desta tendência são as visões da empresa NTT DoCoMo do Japão, que anunciou que em 2006 terá um cartão de crédito embutido nos seus celulares. No entanto, a primeira versão do seu celular, chamado “telefone carteira”, foi lançada em julho de 2004, onde 22 mil lojistas do

Japão possuem instalados dispositivos especiais nas suas caixas registradoras, os quais recebem o sinal do “telefone carteira”. Além disso, ainda em 2006, os japoneses poderão pagar o seu metrô acenando seus telefones móveis em sensores especiais.

Conforme Moskorz (2002, p. 90), o *M-Commerce*, se for considerado como um produto, é muito diferenciado, tanto pelas tecnologias utilizadas, quanto pelos novos valores adicionados ao conceito. É uma nova tecnologia, composta de software e hardware, que vem para melhorar a qualidade de vida das pessoas. Os benefícios desta tecnologia não são imediatamente claros para as pessoas e isto gera uma incerteza com relação a sua utilização e conseqüentemente aos seus resultados. Mas também é uma oportunidade para a melhoria de qualidade de vida e esta pode ser a motivação para uma pessoa aprender sobre a inovação, no caso o *M-Commerce*.

Quando se fala de *M-Commerce* e de aparelhos sem fio, não se pode limitar a celulares. Vários outros aparelhos poderão fazer conexão com o mundo a partir de qualquer lugar, no entanto as tecnologias envolvidas no *M-Commerce* vão muito além dos aparelhos. Basicamente, segundo Siau, Lim e Shen (2001, apud Moskorz, 2002, p. 38), existem três tecnologias-chave envolvendo o *M-Commerce*:

- f) tecnologias de comunicação: para que o potencial do *M-Commerce* se realize, é necessária uma taxa consideravelmente alta em relação à velocidade de transmissão;
- g) tecnologia de troca de informação: a utilização de tecnologias de telecomunicações tem como principal característica a troca de informações. Atualmente existem dois grandes padrões de linguagem para a troca de informações por dispositivos móveis. (i) Padrões de troca de informações baseados em HTML: Uma grande vantagem é que HTML já é bastante utilizado como padrão para a Internet. A maioria dos navegadores suportam este padrão e existem diversas ferramentas para

- ele; (ii) Padrões baseados em *Extensible Markup Language* (XML): O XML “etiqueta” os dados, pondo conteúdo num contexto. Através do uso extensivo do XML, pretende-se fazer com que diversos softwares, escritos em diversas linguagens diferentes se comuniquem entre si;
- h) tecnologia de identificação da localização e do usuário: A localização do usuário é de bastante importância na comunicação móvel, de modo a lhe fornecer apenas serviços e informações relevantes. A tecnologia *Global Positioning System* (GPS), que usa satélites para calcular a posição geográfica com grande precisão, promete desempenhar um importante papel na comunicação sem fio. Já sobre a identificação do usuário, uma das tecnologias-chave para isto está sendo o *Smartcard*. São cartões do tamanho de cartões de crédito magnéticos, mas com um chip e um pequeno processador, que podem ter a funcionalidade de cartão de débito/crédito e também servir como assinatura digital ou carteira eletrônica.

De acordo com Barbosa (2006, p. 1), um estudo da consultoria In-Stat indica que na próxima década cerca de 25 milhões de telefones nos EUA farão pagamentos, ou seja, poderão ser utilizados como carteiras. Em lugar do *M-Commerce*, o celular será utilizado para finalizar transações de compra. Isso ocorreria por meio de mecanismos seguros de autenticação e pelo fato do celular ser um aparelho que acompanha o usuário pela maior parte do tempo.

2.3 DISPOSITIVOS MÓVEIS

Conforme Schmitt Junior (2004, p. 17), pode ser considerado como dispositivo móvel, um dispositivo com capacidade de processamento em um ambiente de rede sem fio. Os dispositivos móveis possuem algumas limitações em relação à taxa de comunicação e

processamento, por se tratar de um ambiente de rede sem fio e por terem um tamanho pequeno. Alguns dos dispositivos móveis que são encontrados atualmente são:

- a) telefones celulares;
- b) *paggers*;
- c) PDAs;
- d) dispositivos embarcados;
- e) computadores de bordo automotivo;
- f) *handheld*.

Conforme Nokia (2005, apud Pamplona, 2005, p. 18), hoje em dia os mais conhecidos e utilizados dispositivos são os celulares, que evoluíram tanto que não podem mais ser considerados como simples telefones.

Segundo Teixeira (2005, p. 1), o celular é a tecnologia que mais rapidamente cresce na história da humanidade. Pode-se destacar o número de dois bilhões de celulares como um número fabuloso, sendo talvez difícil de compreender. A população mundial, por exemplo, é de 6.469.622.414 pessoas, segundo U.S. Census Bureau (2005, apud Teixeira 2005). Este número pode ser comparado com outras estatísticas tecnológicas, onde dois bilhões é cerca do dobro do número de usuários da Internet; dois bilhões é o triplo do número de computadores pessoais funcionando no mundo. Existem mais celulares do que televisão, mais celulares do que cartão de crédito, mais celulares do que automóveis e muito mais celulares do que aparelhos de telefonia fixa.

Conforme Teixeira (2005, p. 1), o celular é o único aparelho digital portado por quase qualquer pessoa economicamente viável no planeta Terra. Cada vez menos pessoas estão usando relógios de pulso, dependendo do celular como referência de hora certa. A “febre” do celular chegou a ponto de que a maioria dos usuários mantém seu aparelho ao alcance da mão às 24 horas do dia. Cerca de 60% dos usuários de celular levam seu aparelho para perto de

onde dormem, seja para receber chamadas de voz, ou então para utilizá-lo como despertador, ou para ler mensagens *Short Message Service* (SMS), caso receba.

Segundo Teixeira (2005, p. 1), quanto ao futuro é esperado três bilhões de aparelhos celulares para o ano de 2010, portanto é possível imaginar as variadas funcionalidades que esses aparelhos terão. É provável um grande aumento das taxas de penetração das redes de alta velocidade e também um grande aumento de processamento dos aparelhos de telefonia móvel.

2.4 JAVA

Segundo Montenegro e Pereira (2005, p. 28), a tecnologia Java completa 10 anos e o seu sucesso é inquestionável. As milhares de aplicações Java existentes englobam as variadas áreas do conhecimento e rodam nas mais diversas plataformas e sistemas operacionais, desde equipamentos bastante limitados (como celulares, PDAs, computadores de bordo) até poderosos *clusters* de servidores, atendendo a centenas ou milhares de clientes simultâneos.

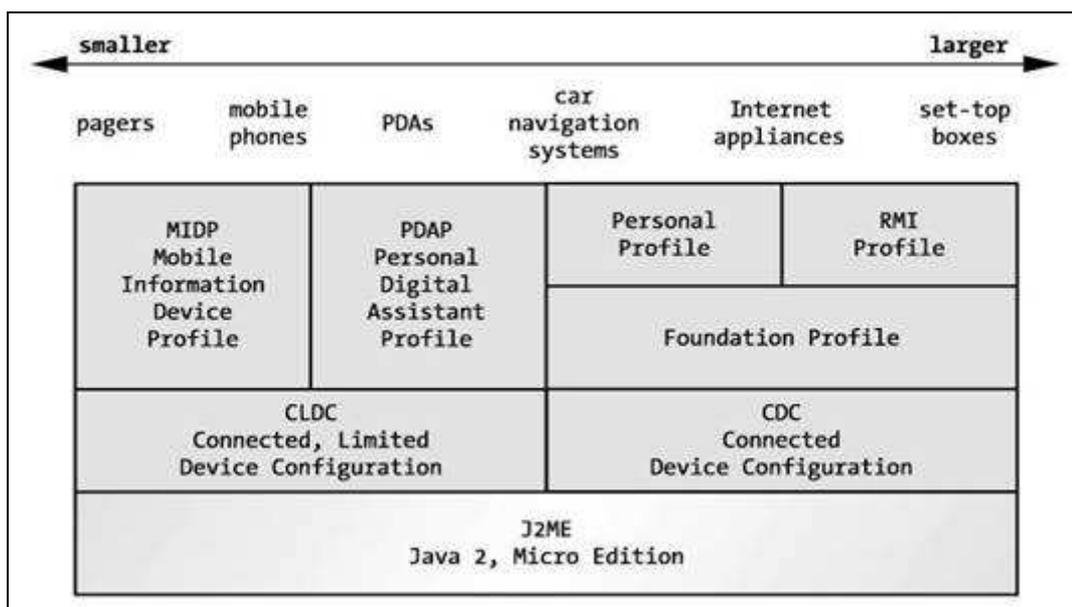
A plataforma Java foi dividida em três grandes edições:

- a) *Java 2 Standard Edition* (J2SE): o núcleo da plataforma, com a máquina virtual e as *Application Programming Interface* (APIs) básicas;
- b) *Java 2 Enterprise Edition* (J2EE): complementando a J2SE e fornecendo recursos para o desenvolvimento de aplicações *Web* e corporativos;
- c) *Java 2 Micro Edition* (J2ME): definindo um ambiente Java para dispositivos móveis.

2.4.1 J2ME

Segundo Muchow (2004, p. 2), J2ME é destinado aos dispositivos consumidores com poder limitado. Muitos desses dispositivos (por exemplo, celular ou *pager*) não tem opção de *download* e software de instalação, além daquele que foi configurado durante o processo de fabricação. Com o conceito de J2ME, os dispositivos móveis não precisam mais ter natureza “estática”. Exatamente como um navegador *Web* fazendo *download* de *applets* Java, uma implementação de J2ME em um dispositivo permite a opção de navegar, fazer *download* e instalar aplicativos Java e conteúdo.

De acordo com Almeida (2004, p. 21), atualmente a plataforma J2ME consiste de um conjunto de *configurations* (configurações), *profiles* (perfis) e APIs opcionais. Essa divisão permite ao desenvolvedor conhecer informações específicas sobre as diferentes famílias de dispositivos e as APIs disponíveis em cada uma delas. Na Figura 1 é possível visualizar os componentes da J2ME.



Fonte: Knudsen (2003, p. 2)

Figura 1 – Universo J2ME

Conforme Muchow (2004, p. 3), uma *configuration* define uma plataforma Java para uma ampla variedade de dispositivos. Ela está intimamente vinculada a uma máquina virtual Java (*Java Virtual Machine (JVM)*). Na verdade, uma *configuration* define os recursos da linguagem Java e as bibliotecas Java básicas da JVM para essa configuração em particular. Entre as configurações disponíveis destacam-se a *Connected Device Configuration (CDC)* que possui um conjunto de APIs que suportam equipamentos fixos de porte médio, tal como televisores e a *Connected Limited Device Configuration (CLDC)* que possui um conjunto de APIs destinadas a aparelhos cujo poder de processamento, *display* e memória são limitados.

Segundo Almeida (2004, p. 22), *profiles* são conjuntos de APIs que suplementam as *configurations*, fornecendo funcionalidades para um determinado tipo de dispositivo ou mercado vertical. São mais específicos que as *configurations*, apesar de serem baseados nelas. Adicionam APIs para interface com o usuário, armazenamento persistente e outras funções auxiliares. O *profile* utilizado em celulares é a MIDP; o utilizado em dispositivos de rede sem interface gráfica é o *Foundation Profile*; já o *profile* utilizado em dispositivos com suporte gráfico é o *Personal Profile*.

2.4.1.1 Configuração CLDC

Conforme Almeida (2004, p. 21), a CLDC consiste em uma máquina virtual chamada de *Kilobyte Virtual Machine (KVM)* e um conjunto de classes mais apropriado para dispositivos pequenos e com limitações de desempenho e memória. Esta configuração é destinada para dispositivos *wireless* menores, possuindo geralmente entre 160 *kilobytes (kb)* e 512 kb disponíveis para Java (mínimo de 128 kb), uma conexão à rede limitada, intermitente e provavelmente lenta. A tela é de tamanho pequeno, e a fonte de energia é reduzida, fornecida por baterias. Este ambiente foi desenvolvido para dispositivos como telefones celulares,

paggers, PDAs.

2.4.1.2 Perfil MIDP

Segundo Almeida (2004, p. 22), o perfil MIDP traz as classes de interface com o usuário e acesso à rede, dando nome aos aplicativos desenvolvidos para celulares. De maneira semelhante às *applets* e aos *servlets*, um aplicativo MIDP é chamado de *midlet*, por herdar de uma classe denominada *MIDlet*. Com essa herança, a classe recebe a capacidade de se comunicar com o gerenciador de aplicativos que os dispositivos MIDP possuem e de interagir com a interface do aparelho.

Um aplicativo MIDP é um conjunto de classes que implementam seu funcionamento, reunidas em um pacote chamado *MIDlet Suite*. O aplicativo é desenvolvido em um computador *desktop*, compilado, pré-verificado, empacotado no arquivo *Java Archive* (JAR) e então instalado em um dispositivo para execução ou em um emulador.

Segundo Muchow (2004, p. 20), os requisitos de hardware de um dispositivo que pretende implementar o MIDP, são:

- a) a tela deve suportar pelo menos 96 x 54 *pixels*;
- b) deve haver pelo menos um tipo de entrada de usuário disponível: teclado de uma mão (teclado de telefone), teclado de duas mãos (teclado de computadores), ou uma tela de toque;
- c) 128 kb de memória não-volátil para executar os componentes *Mobile Information Device* (MID), dispositivo de informação móvel;
- d) 8 kb de memória não-volátil para os aplicativos armazenarem dados persistentes, como configurações e dados do aplicativo;
- e) 32 kb de memória volátil para executar o Java;

f) conectividade de rede sem fio.

2.4.2 J2EE

Conforme Montenegro e Pereira (2005, p. 29), o J2EE possui recursos que o desenvolvedor não precisará implementar. Por exemplo, para o desenvolvimento de aplicações corporativas e/ou para Internet, normalmente é necessário o uso de recursos avançados como objetos distribuídos, mapeamento objeto-relacional para persistência de dados, controle de transação, segurança, manipulação de XML, *Web Services*, entre outros.

Com J2EE, toda a infra-estrutura para gerenciar os aspectos complexos ficam a cargo de um servidor de aplicativos ou de um servidor *Web*. Qualquer componente da J2EE precisa rodar dentro de um servidor. Portanto, existe uma inversão de controle, na qual o desenvolvedor deve usar uma API padronizada e implementar métodos que serão invocados pelo servidor.

Por exemplo, o desenvolvedor não precisa se preocupar com o número de objetos que devem ser instanciados para que a aplicação possa ter escalabilidade. Quem decide quando um objeto deve ser instanciado é o servidor. Isso permite que o desenvolvedor se preocupe somente com o domínio do problema, ou seja, a implementação das regras de negócio da aplicação. Usando as APIs de forma correta, possibilita-se o desenvolvimento de aplicações escaláveis, fáceis de integrar com outros sistemas e compostas por componentes reutilizáveis e portáteis.

2.5 WEB SERVICES

De acordo com Pamplona (2004, p. 1), um requisito básico de qualquer empresa é

prover serviços. Cada empresa oferece serviços para a comunicação entre ela e outras pessoas, sejam pessoas físicas ou jurídicas, internas ou externas a empresa.

Alguns desses serviços podem ser automatizados. Por exemplo, não é necessário existir um representante de vendas se o seu cliente já tem, em mãos, o preço e todos os outros dados relevantes para constituir um pedido de compra. Este pedido pode e, em muitos casos, já é feito, via interfaces computacionais. O cliente entra no *site*, monta o pedido como desejar e confirma a compra. Isto é um serviço *Web*, ou seja, um serviço que está publicado na *Web* para que qualquer pessoa possa fazer uso.

Web Services foram criados para construir aplicações deste tipo, aplicações que são serviços na Internet. Porém, não faz parte do conceito de *Web Service* a criação de interfaces gráficas para os usuários. Assim é a tecnologia ideal para comunicação entre sistemas, pois a comunicação entre os serviços é padronizada possibilitando a independência de plataforma e de linguagem de programação.

Conforme Cunha (2002, p. 1), *Web Services* são descritos e definidos usando XML. Um dos motivos que tornam *Web Services* atrativos é o fato deste modelo ser baseado em tecnologias *standards*, em particular XML e HTTP. O protocolo SOAP é um dos protocolos utilizados para a troca de mensagens entre aplicações e *Web Services*, que é construída com base em XML e HTTP.

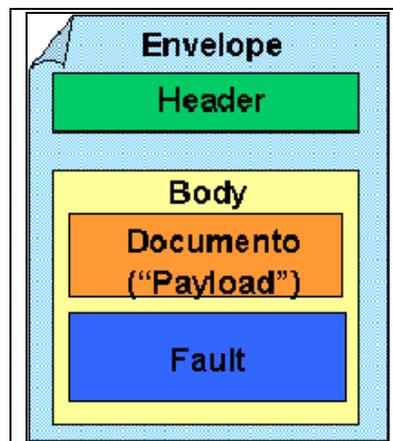
2.5.1 SOAP

De acordo com Cunha (2002, p. 1), SOAP é um protocolo projetado para invocar aplicações remotas através de *Remote Procedure Calls* (RPC), Chamadas Remotas de Procedimentos, ou troca de mensagens, em um ambiente independente de plataforma e de linguagem de programação. Portanto é um padrão aceito para a utilização com *Web Services*.

Desta forma, pretende-se garantir a interoperabilidade e intercomunicação entre diferentes sistemas, através da utilização de uma linguagem (XML) e mecanismo de transporte (HTTP) padrões.

Uma mensagem SOAP (Figura 2) consiste basicamente dos seguintes elementos:

- a) *envelope*: toda mensagem SOAP deve contê-lo. É o elemento raiz do documento XML. O *Envelope* pode conter declarações e *namespaces* e também atributos adicionais como o que define o estilo de codificação, *encoding style*. Um *encoding style* define como os dados são representados no documento XML;
- b) *header*: é um cabeçalho opcional. Ele carrega informações adicionais, como por exemplo, se a mensagem deve ser processada por um determinado nó intermediário. Pode-se dizer que ao trafegar pela rede, a mensagem normalmente passa por diversos pontos intermediários, até alcançar o destino final. Quando utilizado, o *Header* deve ser o primeiro elemento do *Envelope*;
- c) *body*: este elemento é obrigatório e contém o *payload*, ou a informação a ser transportada para o seu destino final. O elemento *Body* pode conter um elemento opcional *Fault*, usado para carregar mensagens de *status* e erros retornadas pelos nós ao processarem a mensagem.



Fonte: Cunha (2002, p. 2)

Figura 2 – Estrutura de uma mensagem SOAP

2.6 TRABALHOS CORRELATOS

Nesta seção são apresentados alguns trabalhos correlatos ao presente trabalho.

Rosa (2005), desenvolveu um protótipo de uma aplicação para dispositivos móveis onde, também como neste trabalho, foi utilizada a tecnologia J2ME, possuindo como base o diário de classe da FURB, que foi implementado para utilizá-lo no telefone celular. O módulo do telefone celular também se comunicou com um *Web Service* em um servidor de aplicações, via *Hyper Text Transfer Protocol Secure* (HTTPS) e SOAP.

Em Schmitt Junior (2004), é relatado o desenvolvimento de um protótipo de *front end* de Controle de Acesso que, também como neste trabalho, utilizou a tecnologia J2ME, que possui como objetivo principal a automatização do controle de segurança patrimonial. Para o armazenamento das informações utilizadas pelo sistema foi utilizado o recurso *Record Management System* (RMS), um dos recursos especiais da J2ME. Para a troca de dados com um servidor, foram utilizadas classes que automatizaram a leitura de arquivos XML, outro recurso avançado da J2ME.

Pio (2001) apresenta a GoWap, empresa de soluções *wireless*, que desenvolveu o sistema *Gopayment*, que, também como nesta proposta, permite que o usuário, por meio do celular, acione uma máquina de refrigerantes. O pedido é registrado e enviado para a operadora do cliente, que no final do mês, manda a fatura junto com a conta telefônica. Segundo Fabrício Biasi (2001, apud Pio, 2001), este é um exemplo do que o comércio sem fio é capaz. “Hoje só podemos comprar refrigerantes, amanhã poderemos comprar o que quisermos. É só uma questão de desenvolver a tecnologia que já existe”.

A Companhia de Telecomunicações do Brasil Central (CTBC) desenvolveu o serviço SHOPcel, que, também como nesta proposta, trata de *M-Commerce*, onde permite que o usuário ligue para um número exposto em cada máquina de refrigerante, que reconhece o

número do telefone da CTBC, e em seguida o cliente escolhe o refrigerante e este é liberado automaticamente. Este procedimento leva em média dois minutos e o débito será feito somente na próxima conta do telefone pós-pago que originou a ligação. Já os usuários de celulares pré-pagos precisam fazer uma autorização prévia de débito diretamente na operadora ou através do 0800. O cliente paga somente o valor do produto e a ligação local que é efetuada para a máquina, de acordo com o plano de tarifas do seu celular (Call to Call, 2002).

3 DESENVOLVIMENTO DO TRABALHO

A aplicação comercial baseada em *M-Commerce*, resultante deste trabalho é um software que permite a compra de produtos ou serviços através do aparelho celular. Especificamente neste trabalho optou-se por trabalhar com compra de refrigerante em máquinas de venda de refrigerante. O software é composto por três módulos:

- a) módulo MIDP comprador que é executado em um aparelho celular, utilizado pelo consumidor, o qual utiliza a tecnologia J2ME, sendo que este módulo se comunica com um *Web Service* em um servidor de aplicações, via HTTPS e SOAP;
- b) módulo central que é executado na *Web*, o qual utiliza a tecnologia J2EE, onde são mantidas todas as informações do consumidor, ponto de venda, produtos e compra e venda;
- c) *Web Service* para possibilitar a comunicação do módulo MIDP comprador com o módulo central.

A Figura 3 demonstra a comunicação entre os módulos apresentados.

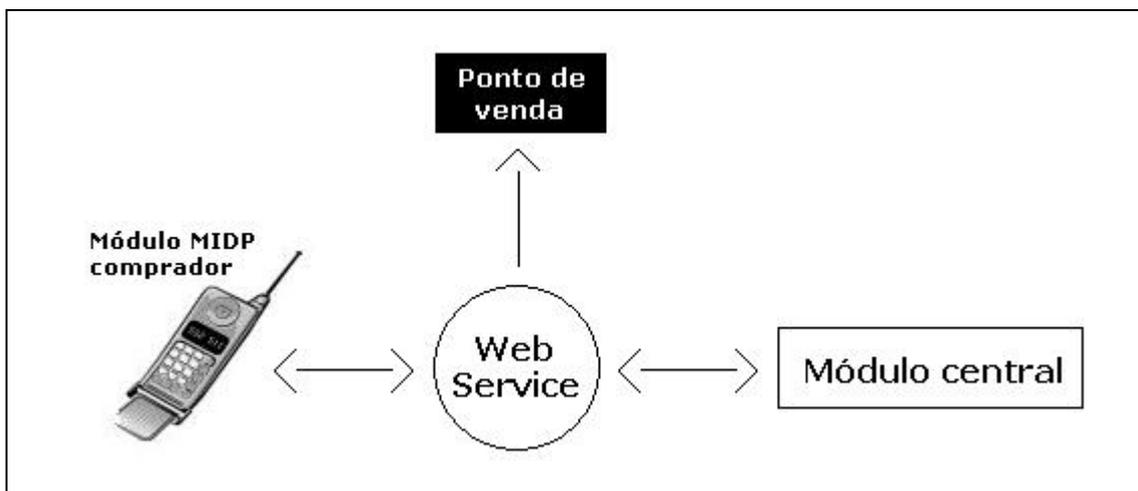


Figura 3 – Comunicação entre módulos da aplicação M-Commerce

O módulo MIDP vendedor não foi desenvolvido neste trabalho conforme o proposto, pois o escopo do trabalho não era o gerenciamento de recebimento de uma informação de compra e disponibilizar produtos. Em torno disso foi optado enviar ao dispositivo do ponto de venda um e-mail com os produtos, como sinal de que uma compra foi realizada.

Para o desenvolvimento do trabalho foi realizado o levantamento e a análise de requisitos que contemplam as características do sistema. Os requisitos foram definidos com base no conceito de *M-Commerce*. Foi necessária uma especificação para expressar através de diagramas como os requisitos foram tratados. A análise dos requisitos e suas especificações são tratadas a seguir.

3.1 REQUISITOS PRINCIPAIS DO SISTEMA

Os requisitos descrevem o que o sistema deve fazer. Eles estão classificados em requisitos funcionais e requisitos não funcionais. Requisitos funcionais são funcionalidades que o sistema deve possuir e o comportamento do sistema em determinadas situações, podendo também explicitar o que o sistema não deve fazer. Requisitos não funcionais são restrições sobre os serviços ou as funções oferecidas pelo sistema, como usabilidade, hardware, segurança.

3.1.1 Requisitos funcionais

O Quadro 1 apresenta os principais requisitos funcionais do sistema.

REQUISITOS FUNCIONAIS
RF01: O módulo central deve solicitar ao usuário a sua identificação através de login e senha.
RF02: O módulo central deve permitir ao gerente o cadastramento dos consumidores.
RF03: O módulo central deve permitir ao gerente o cadastramento dos gerentes e vendedores.
RF04: O módulo central deve permitir ao gerente o cadastramento das formas de pagamento.
RF05: O módulo central deve permitir ao gerente o cadastramento dos tipos de produto.
RF06: O módulo central deve permitir ao gerente o cadastramento dos produtos disponíveis para venda.
RF07: O módulo central deve permitir ao gerente o cadastramento das cidades para os locais dos pontos de venda.
RF08: O módulo central deve permitir ao gerente o cadastramento dos locais para os pontos de venda.
RF09: O módulo central deve permitir ao gerente o cadastramento dos pontos de venda.
RF10: O módulo central deve permitir ao vendedor a seleção dos produtos disponíveis para venda em cada ponto de venda.
RF11: O módulo MIDP comprador deve solicitar ao consumidor a sua identificação através de login e senha.
RF12: O módulo MIDP comprador deve solicitar ao consumidor a identificação do ponto de venda.
RF13: O módulo MIDP comprador deve disponibilizar ao consumidor uma lista de itens disponíveis para compra, de acordo com o ponto de venda identificado.
RF14: O módulo MIDP comprador deve permitir ao consumidor a escolha de itens disponíveis na lista para compra, e permitir informar a quantidade desejada para compra.
RF15: O módulo MIDP comprador deve permitir ao consumidor a confirmação dos itens escolhidos para compra.
RF16: O módulo MIDP comprador deve permitir ao consumidor a escolha da forma de pagamento.
RF17: O módulo MIDP comprador deve permitir ao consumidor confirmar a transação de compra.
RF18: O módulo central deve receber a solicitação de compra do módulo MIDP comprador e cadastrar a solicitação de compra.
RF19: O módulo central deve enviar um e-mail ao celular do ponto de venda, como confirmação para o ponto de venda disponibilizar o produto ao consumidor.

Quadro 1 – Requisitos funcionais

3.1.2 Requisitos não funcionais

O Quadro 2 apresenta os principais requisitos não funcionais do sistema.

REQUISITOS NÃO FUNCIONAIS
RNF01: A comunicação entre o módulo central e o módulo MIDP comprador deve ser através de <i>Web Service</i> com SOAP.
RNF02: Interface com usuários do sistema através do dispositivo celular e <i>browser Web</i> .
RNF03: Meio de comunicação seguro entre cliente e o servidor utilizando o protocolo HTTPs.
RNF04: Utilizado o banco de dados MySQL.
RNF05: Utilizado o <i>J2ME Wireless Toolkit</i> , ambiente de desenvolvimento com suporte a J2ME MIDP.
RNF06: Portabilidade em celulares usando a configuração CLDC.
RNF07: Compatibilidade com celulares compatíveis com perfil MIDP.

Quadro 2 – Requisitos não funcionais

3.2 ESPECIFICAÇÃO

A *Unified Modeling Language* (UML) é uma linguagem visual para modelar sistemas orientados a objetos, ou seja, é uma linguagem constituída de elementos gráficos utilizados na modelagem que permitem representar os conceitos do paradigma da orientação a objetos. Através dos elementos gráficos definidos nesta linguagem pode-se construir diagramas que representam diversas perspectivas de um sistema.

A seguir serão apresentados os casos de uso do sistema, diagrama de atividades dos casos de uso, diagramas de classe, diagrama físico de dados e os diagramas de realização de casos de uso. Também será apresentada a ferramenta de modelagem utilizada para construir a especificação.

3.2.1 Casos de uso do sistema

Os casos de uso representam as funcionalidades externamente observáveis do sistema e dos elementos externos ao sistema que interagem com ele.

Na Figura 4 é possível visualizar o diagrama de pacotes.

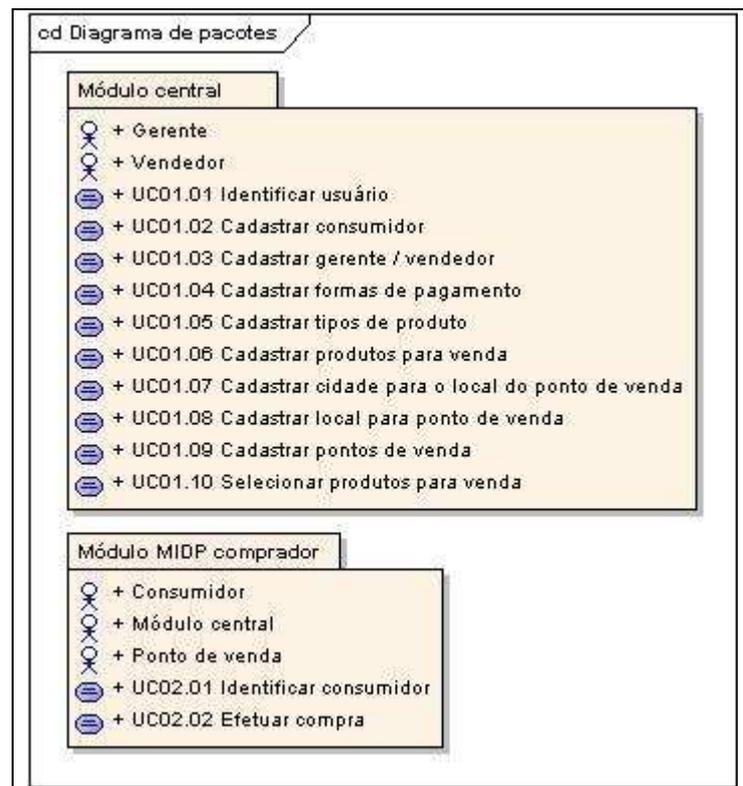


Figura 4 – Diagrama de pacotes

Para o módulo central foram definidos dez casos de uso:

- a) identificar usuário;
- b) cadastrar consumidor;
- c) cadastrar gerente / vendedor;
- d) cadastrar formas de pagamento;
- e) cadastrar tipos de produto;
- f) cadastrar produtos para venda;
- g) cadastrar cidade para o local do ponto de venda;
- h) cadastrar local para ponto de venda;
- i) cadastrar pontos de venda;
- j) selecionar produtos para venda.

Os casos de uso definidos estão representados no diagrama de caso de uso, conforme

Figura 5.

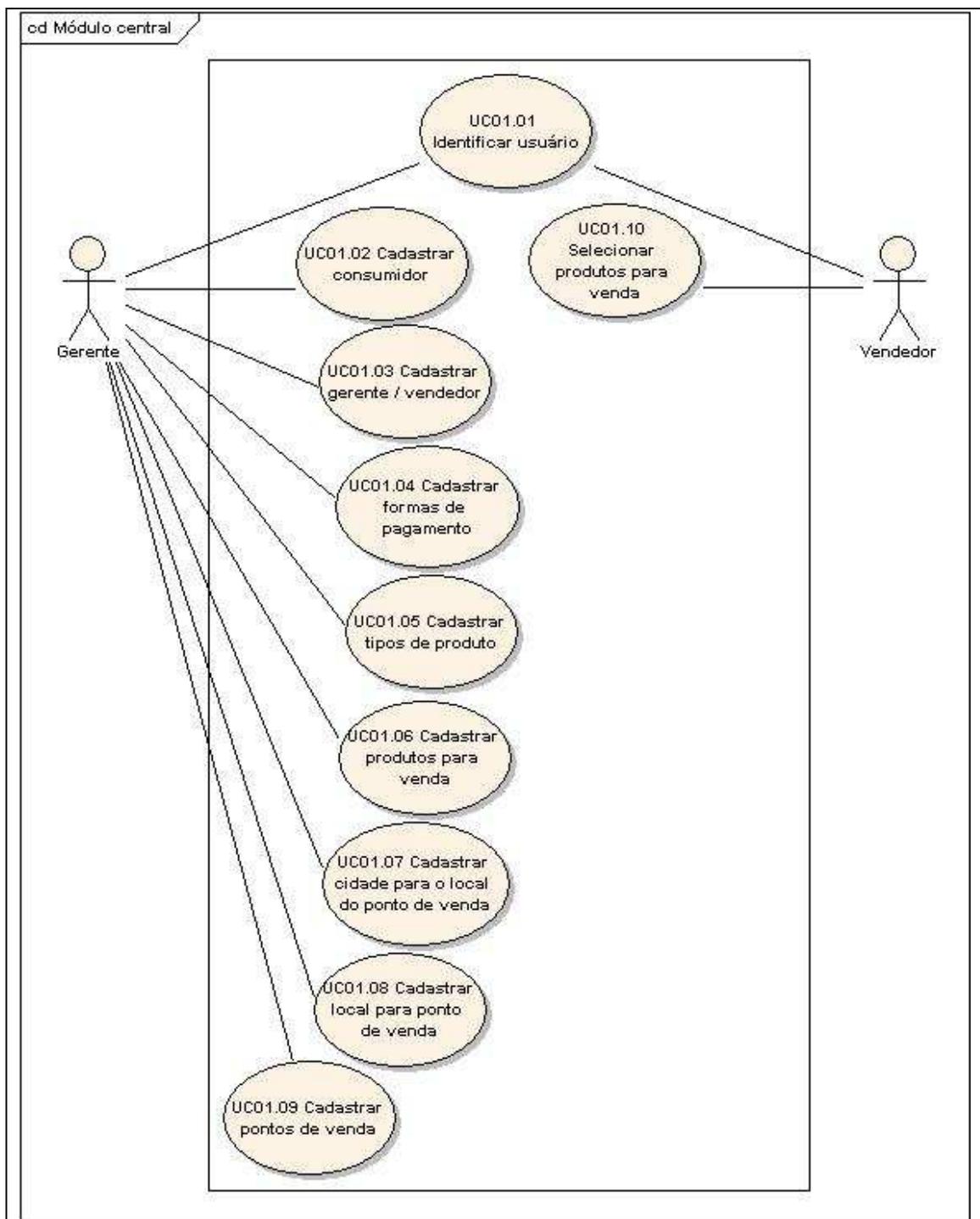


Figura 5 – Diagrama de caso de uso módulo central

Para o módulo MIDP comprador foram definidos dois casos de uso:

- identificar consumidor;

b) efetuar compra.

A seguir, os casos de uso definidos acima estão representados no diagrama de caso de uso, conforme Figura 6.

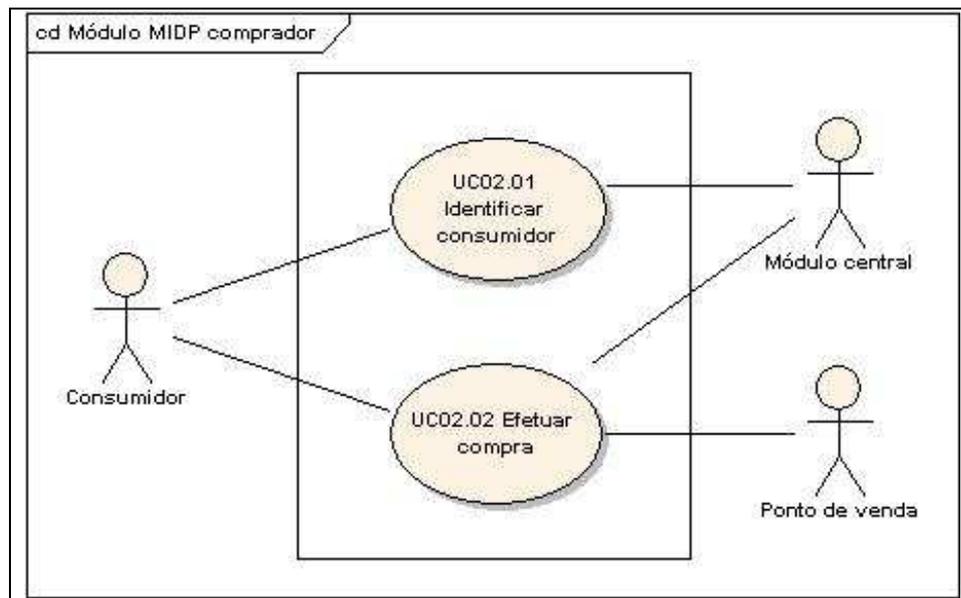


Figura 6 – Diagrama de caso de uso módulo MDP comprador

A descrição mais detalhada dos casos de uso pode ser consultada no Apêndice A.

3.2.2 Diagramas de atividades do sistema

Um diagrama de atividades demonstra o fluxo de uma atividade para outra em um sistema. Para os principais casos de uso foi criado um diagrama de atividades, onde casos de uso de cadastro possuem a mesma lógica do diagrama de atividade do caso de uso “Cadastrar consumidor”.

- a) Identificar usuário (Figura 7);
- b) Cadastrar consumidor (Figura 8);
- c) Identificar consumidor (Figura 9);
- d) Efetuar compra (Figura 9);

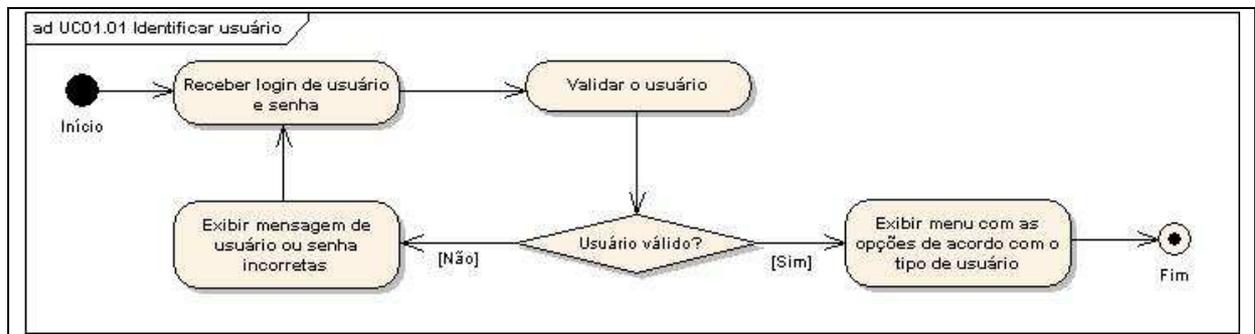


Figura 7 – Diagrama de atividades do caso de uso “Identificar usuário”

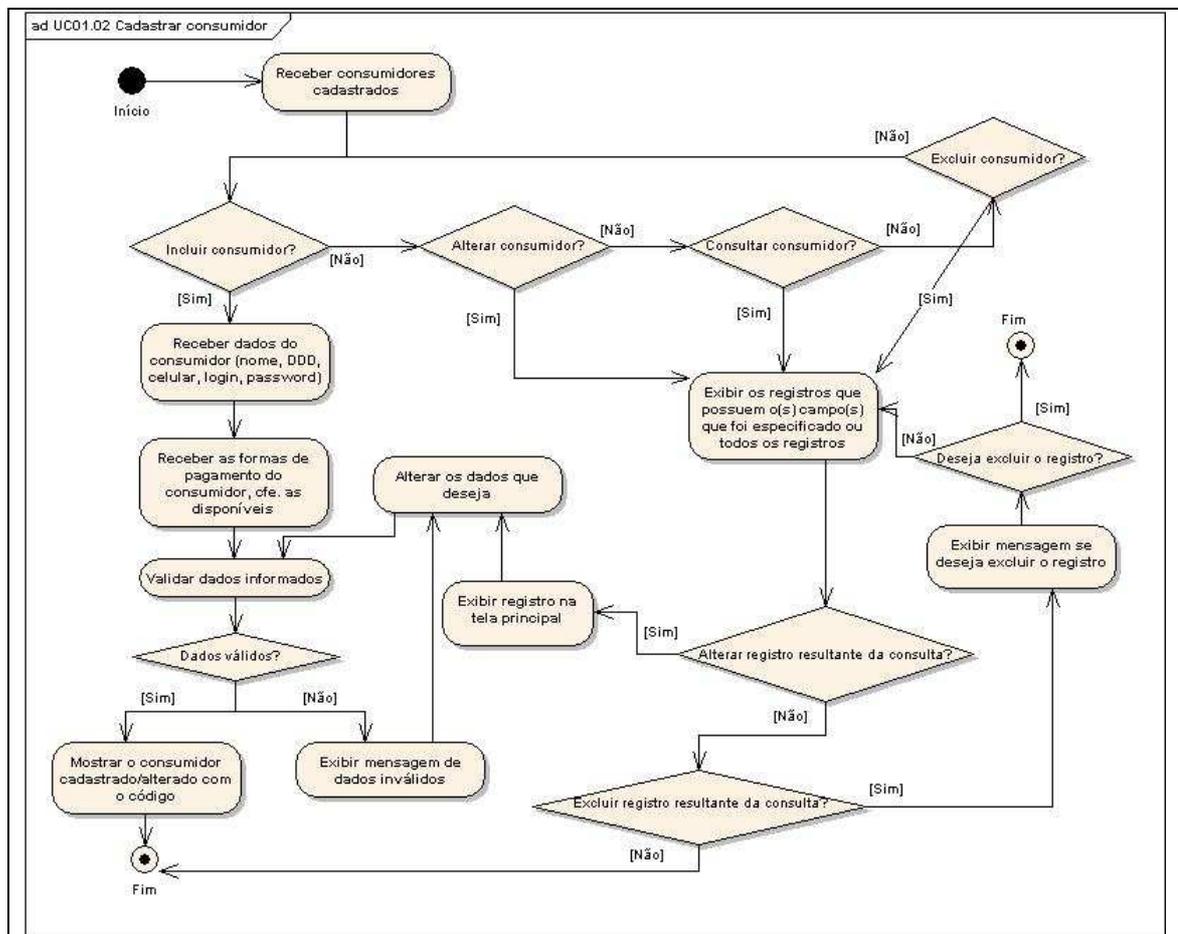


Figura 8 – Diagrama de atividades do caso de uso “Cadastrar consumidor”

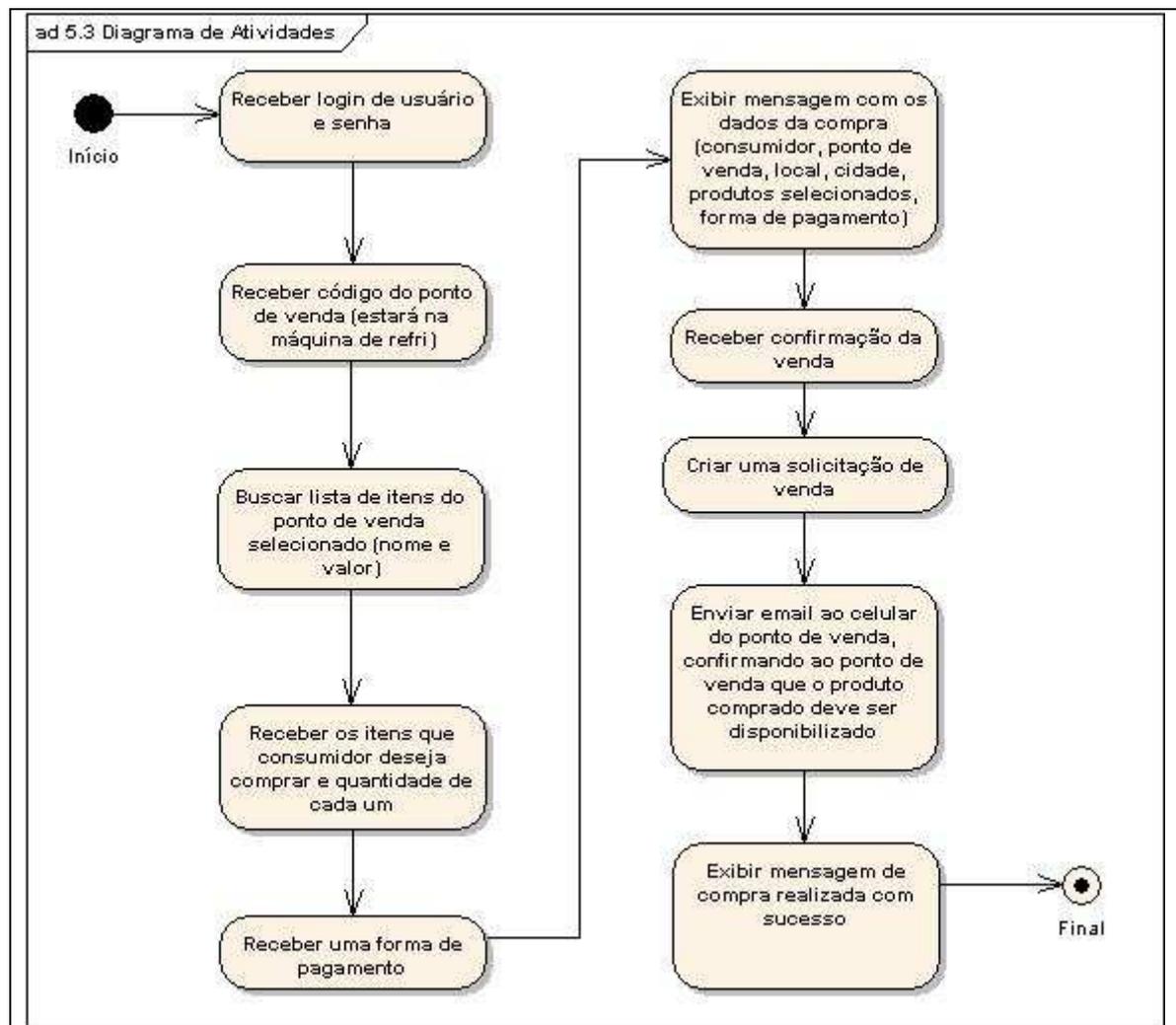


Figura 9 – Diagrama de atividades dos casos de uso “Identificar consumidor” e “Efetuar compra”

3.2.3 Diagramas de classes do sistema

Um diagrama de classes tem por objetivo apresentar o conjunto de classes e os seus relacionamentos em um sistema. Ela detalha também os atributos e métodos que possui. A Figura 10 demonstra o diagrama de classes do sistema, voltado ao negócio.

A classe “PsProdutoPontoVenda” representa o produto (classe “PsProduto”) no ponto de venda (classe “PsPontoVenda”), este está descrito por quantidade e valor do produto em determinado ponto de venda.

A classe “PsProduto” representa o produto que será comprado por um consumidor, este descrito por um código e descrição. Um produto está associado a um tipo de produto, representada pela classe “PsTipoProduto”, descrita com um código e uma descrição.

A classe “PsPontoVenda” representa o ponto de venda que o consumidor irá comprar um produto, este descrito por um código, telefone da máquina, nome e email. Um ponto de venda está associado a um local, representado pela classe “PsLocal”, descrito com um código e um nome. Um local pode estar em uma cidade, representada pela classe “PsCidade”, descrita com um código e um nome.

A classe “PsSolicitacaoCompra” representa a solicitação de compra no momento que o consumidor (classe “PsConsumidor”) confirma uma compra, esta solicitação está descrita por um código, por um campo confirmado representando se a solicitação foi confirmada e a data da compra. Uma solicitação possui o ponto de venda (classe “PsPontoVenda”) que foi feita a solicitação, a forma de pagamento (classe “PsFormaPagamento”) utilizada, o consumidor (classe PsConsumidor) que fez a compra e os itens que foram comprados (classe “PsItemCompra”).

A classe “PsItemCompra” representa o item de uma compra, este está descrito com um código, uma quantidade caso o consumidor queira comprar mais de um item de cada produto, e o valor de cada item. Um item está associado a um produto.

A classe “PsFormaPagamento” representa a forma de pagamento que o consumidor irá pagar a sua compra, esta está descrita por um código e um tipo. Uma forma de pagamento está associada a um ou mais consumidores.

A classe “PsConsumidor” representa o consumidor do sistema, este é um tipo de usuário sendo herança da classe “PsTipoUsuario”, este é descrito com um ddd e um telefone celular. O consumidor possui permissão para efetuar compra no sistema.

A classe “PsUsuario” representa o usuário do sistema, este é descrito com um código,

3.2.4 Diagrama físico de dados

A Figura 11 representa o modelo das classes do modelo de negócios convertido para tabelas no banco de dados.

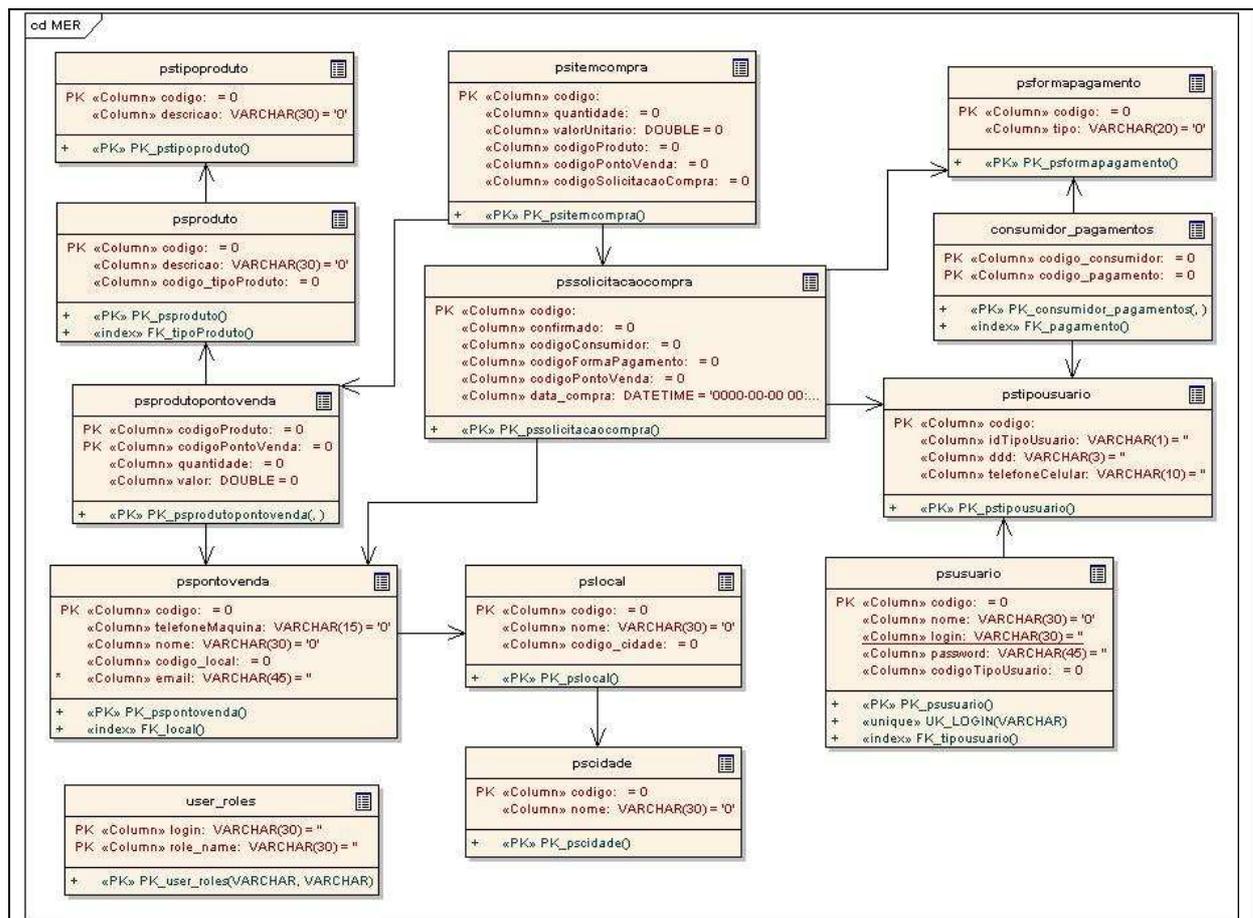


Figura 11 – Diagrama físico de dados

3.2.5 Realizações de casos de uso

O diagrama de seqüência, também conhecido por realização de caso de uso, demonstra a interação das instâncias de classe, demonstrando a troca de mensagens.

Foi construído o diagrama de seqüência dos principais casos de uso, demonstrando a

interação do usuário com a tela do aplicativo e esta com a classe fachada (padrão de projeto *Facade*) do sistema:

- Cadastrar consumidor (Figura 12);
- Selecionar produtos para venda (Figura 13);
- Identificar consumidor (Figura 14);
- Efetuar compra (Figura 14).

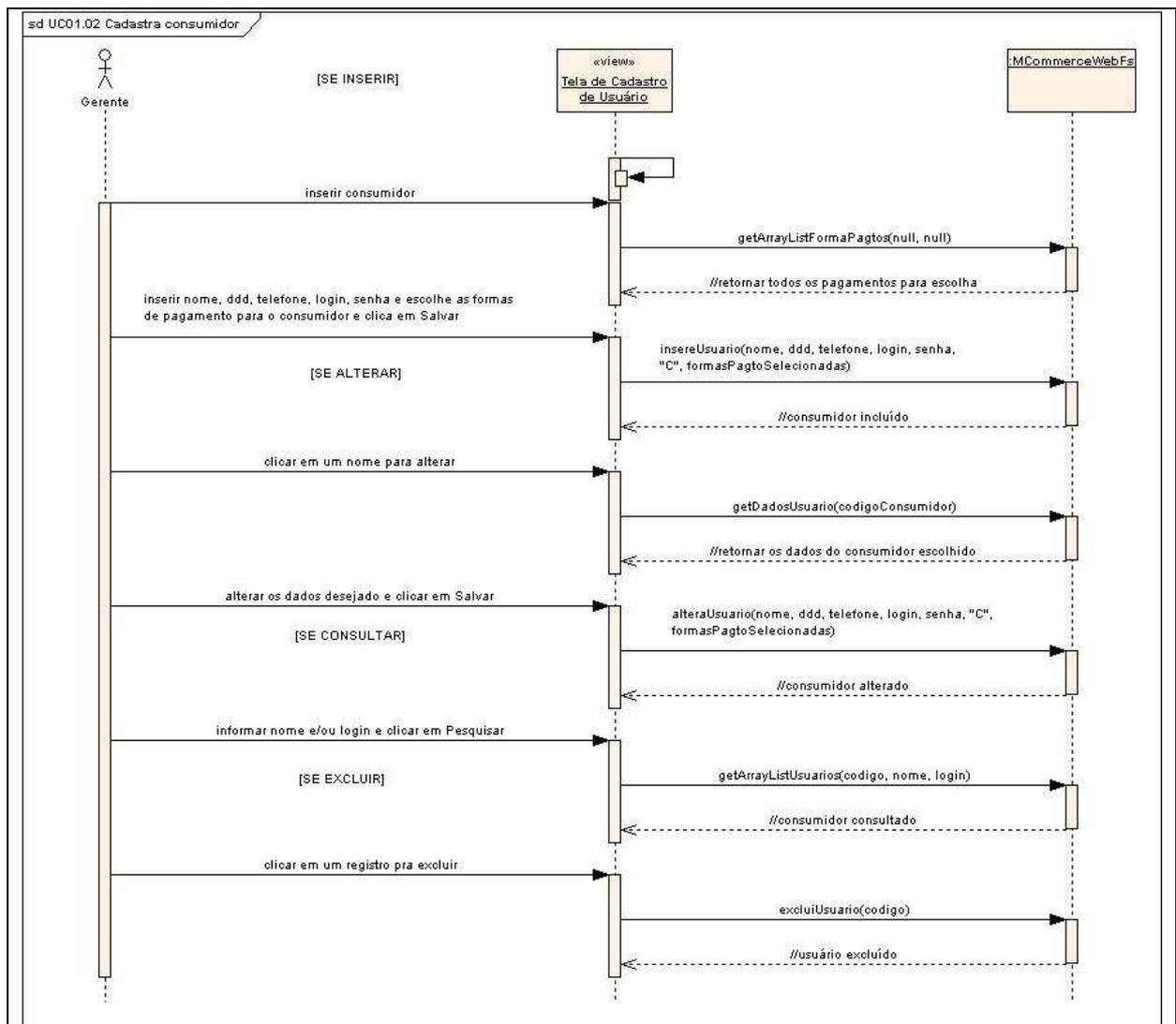


Figura 12 – Diagrama de seqüência do caso de uso “Cadastrar consumidor”

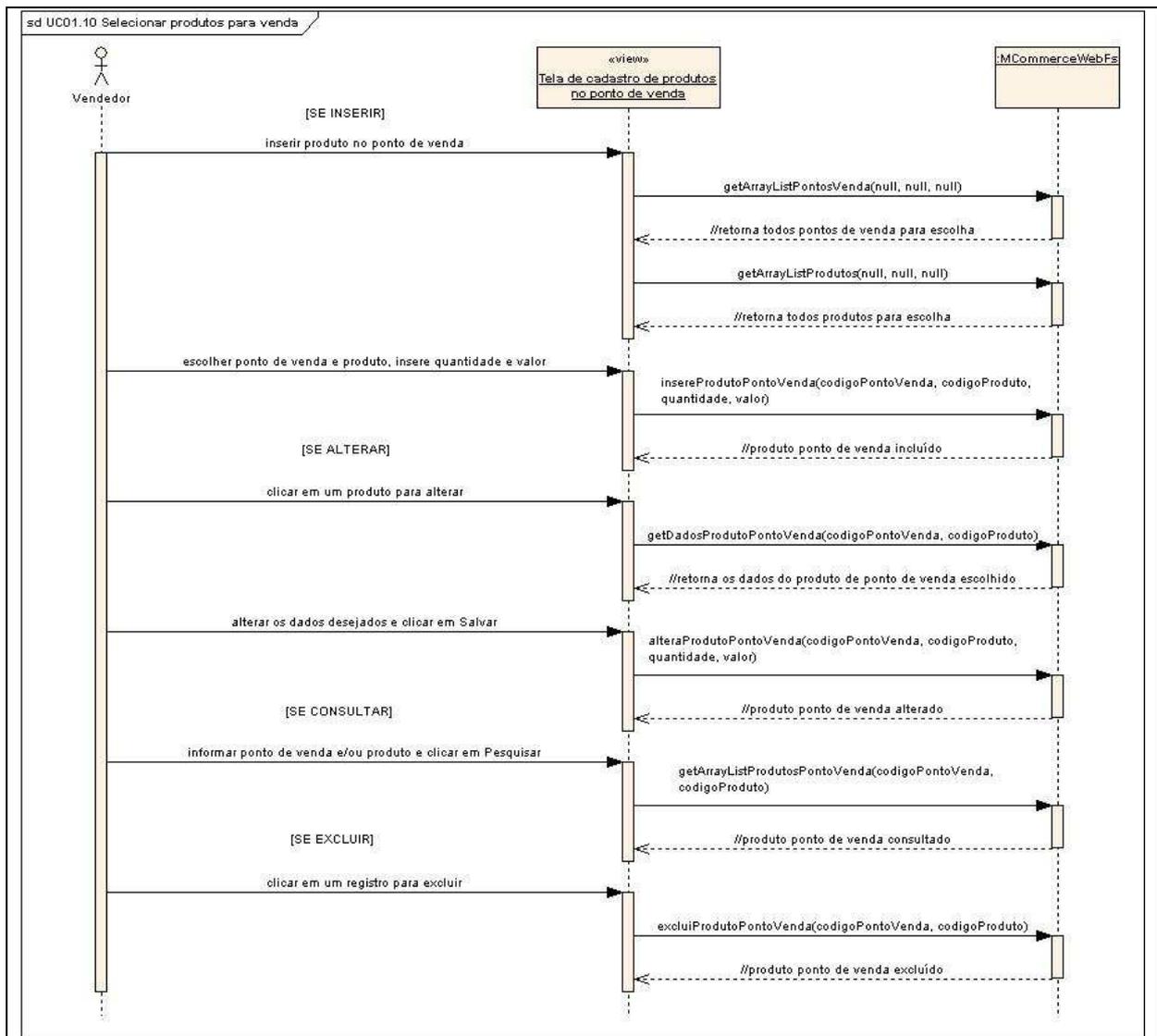


Figura 13 – Diagrama de seqüência do caso de uso “Selecionar produtos para venda”

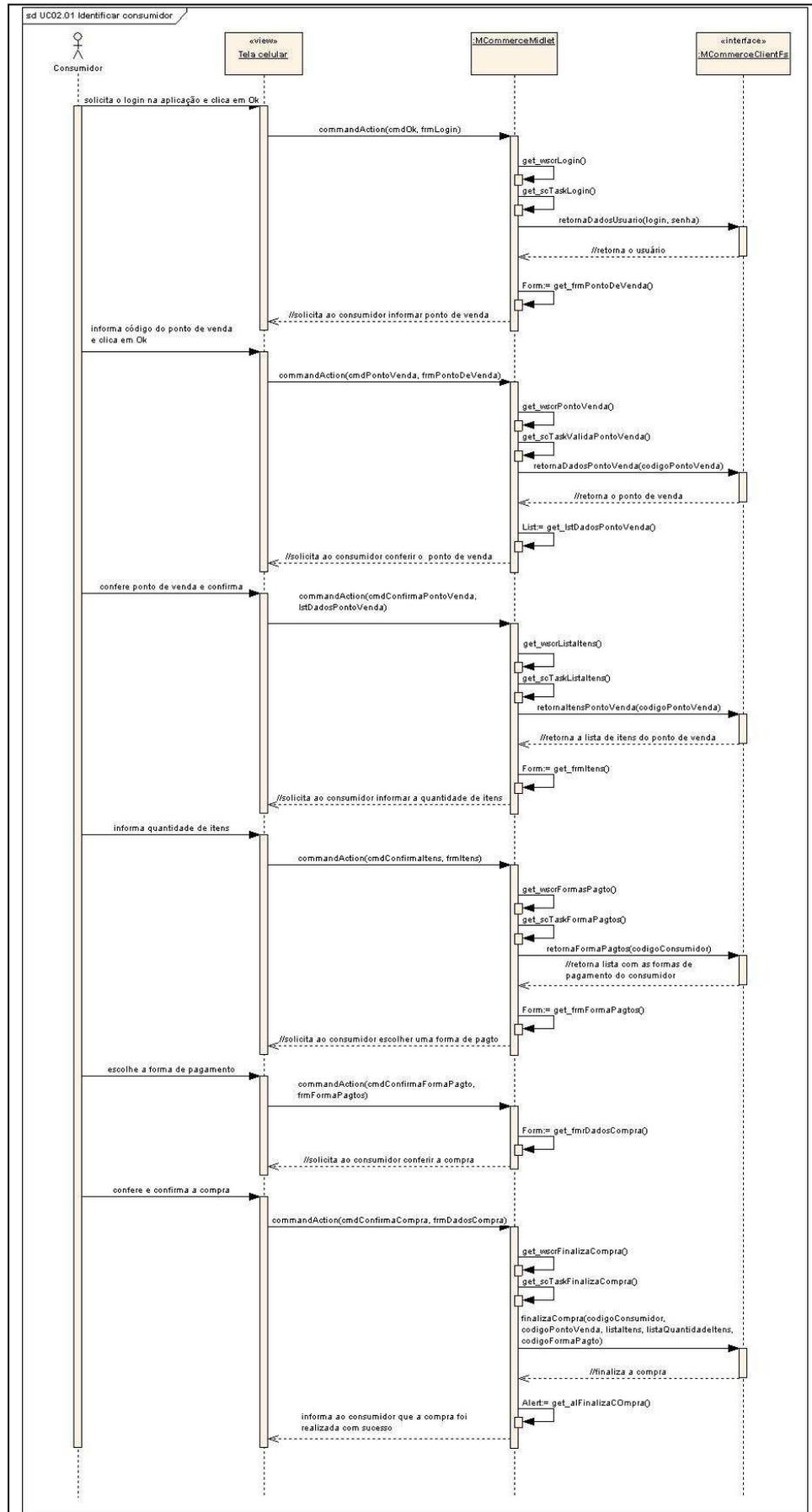


Figura 14 – Diagrama de seqüência dos casos de uso “Identificar consumidor” e “Efetuar compra”

3.2.6 Ferramenta utilizada para a especificação

Segundo Sparx Systems (2004, apud Schmitt Junior, 2004, p. 41), o Enterprise Architect (EA) é uma ferramenta CASE para modelagem, construção e manutenção de sistemas, baseada na linguagem de modelagem unificada (UML).

Com a utilização do EA, foi possível realizar a modelagem do projeto nas seguintes visões:

- a) visão de caso de uso;
- b) visão lógica;
- c) visão dinâmica;
- d) visão física.

A visão de caso de uso utilizou os diagramas de casos de uso para a modelagem, os quais foram definidos a partir dos requisitos funcionais levantados. A visão lógica utilizou os diagramas de atividades com base nas classes modeladas. Já a visão dinâmica, demonstrou através dos diagramas de seqüência a troca de mensagens entre as classes, dos principais casos de uso. A visão física foi realizada através da engenharia reversa, criando o diagrama através das tabelas do banco MySQL.

3.3 IMPLEMENTAÇÃO

A seguir são apresentadas as técnicas, ferramentas utilizadas e operacionalidade da implementação.

3.3.1 Técnicas e ferramentas utilizadas

Para o desenvolvimento do sistema foram utilizadas ferramentas e técnicas que auxiliaram o trabalho, mantendo as informações organizadas e consistentes e tornando possível a implementação do sistema. A seguir são apresentados alguns detalhes de cada uma.

3.3.1.1 NetBeans IDE

A NetBeans IDE é um ambiente de desenvolvimento que permite a escrita, compilação, depuração e instalação de programas, o qual suporta qualquer linguagem de programação. Este ambiente proporcionou auxílio em algumas atividades do sistema, como a codificação em Java do sistema.

Existem vários módulos para estender a NetBeans IDE a fim de facilitar implementações específicas, um deles será descrito a seguir.

3.3.1.2 NetBeans Mobility Pack

O *NetBeans Mobility Pack* é um módulo possível de ser integrado no NetBeans IDE. Ele permite o desenvolvimento de aplicações J2ME, possuindo como característica de desenvolvimento, componentes a disposição do usuário para serem incluídos na MIDLet, gerando o fluxo, conforme Figura 15, o qual realiza a geração automática de grande parte do código.

No momento de implementação do fluxo, ao incluir um componente é gerado o código específico do componente. Para cada componente é necessário informar qual ação o sistema deve acionar quando o usuário escolhe um determinado comando, gerando as associações de

um componente para o outro, no fluxo. No código da MIDlet pode ser personalizado o que for necessário.

Na Figura 15, conforme destacado com o número 1, existe um componente do tipo *Form* que representa o formulário, ou seja, a tela que é visualizada no dispositivo. Nesta tela podem ser incluídos itens do tipo caixa de texto, lista de itens, imagem, etc., e pode ser incluído também ações do tipo Ok, Sair, Cancelar. Destacado pelo número 2, existe um *SplashScreen* que representa uma tela de *splash* ao iniciar a aplicação com o título da aplicação. Nesta tela é necessário configurar o tempo que ela deve aparecer, texto e imagem que ela pode exibir. Destacado pelo número 3, existe um *WaitScreen* que representa a tela para aguardar enquanto é processado algum comando acionado. Nesta tela deve ser configurado o componente que deve ser mostrado na tela no caso de sucesso da ação e no caso de falha. Destacado pelo número 4, existe um *Alert* que representa um alerta de alguma ação bem sucedida ou não. Nesta tela deve ser configurada a mensagem que deve aparecer na tela. Destacado pelo número 5, existe um *List* que representa uma lista com informações para mostrar na tela em forma de lista.

O apêndice E demonstra o código gerado, o qual está destacado, e o código que foi necessário complementar para personalizar a aplicação, sem destaque. Vale ressaltar que não é permitido editar o código gerado. O código demonstrado é do destaque com o número 1, na Figura 15.

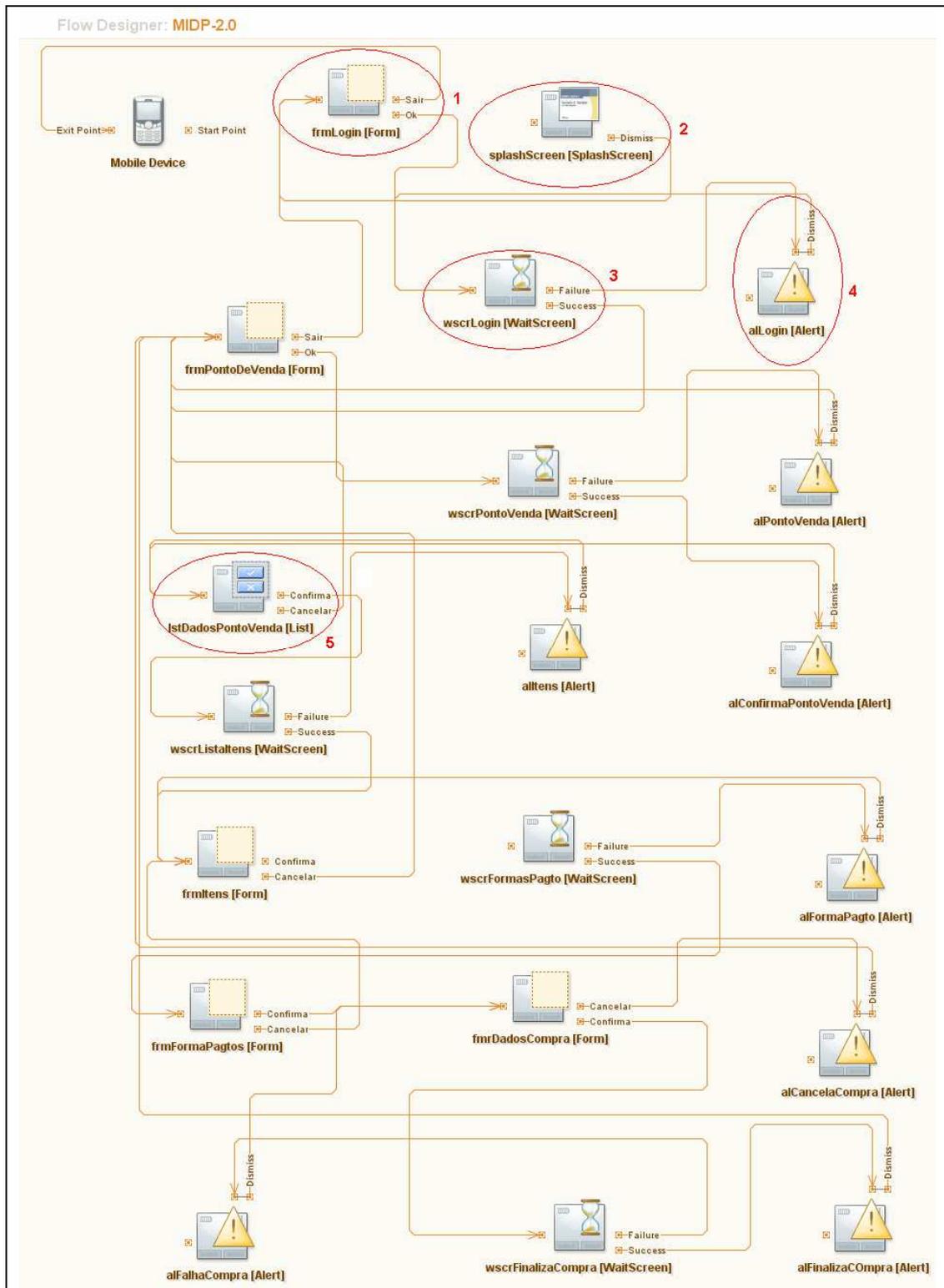


Figura 15 – Fluxo NetBeans Mobility Pack

3.3.1.3 Sun Java Wireless Toolkit

O *Sun Java Wireless Toolkit* é um ambiente de desenvolvimento de aplicações J2ME fornecido pela Sun Microsystems, o qual pode ajudar a gerenciar projetos do J2ME. Ele foi utilizado juntamente com o ambiente de desenvolvimento NetBeans IDE a fim de executar as MIDlets em um emulador.

3.3.1.4 Biblioteca kSOAP2 e kXML

Segundo Rosa (2005, p. 66), kSOAP2 e kXML são bibliotecas para a utilização de SOAP e XML otimizadas para serem executadas em uma aplicação J2ME. Estas bibliotecas foram utilizadas no sistema a fim de oferecer a comunicação do dispositivo móvel com o *Web Services*, decodificando envelopes SOAP no formato XML. O apêndice B demonstra a utilização da biblioteca kSOAP2

3.3.1.5 Tomcat

O *Web Container Tomcat* é um servidor de aplicações, o qual está incluso no NetBeans IDE para ser utilizado na execução das aplicações Java. Foi utilizado para iniciar e encerrar o serviço para a execução do sistema.

3.3.1.6 Hibernate

Segundo Linhares (2005, p. 1), o Hibernate é uma ferramenta de mapeamento objeto/relacional para Java. A idéia principal deste tipo de ferramenta é abstrair do

desenvolvedor as tarefas de consulta, inserção, exclusão e atualização do banco de dados relacional, ou seja, poupar o desenvolvedor de fazer instruções SQL complexas para incluir dois registros relacionados. É bastante útil para sistemas que possuem um modelo rico, onde a maior parte da lógica de negócios fica na própria aplicação Java. O apêndice C demonstra um dos arquivos de mapeamento do sistema e um dos métodos que o utiliza.

3.3.1.7 Apache Struts

Segundo Menéndez e Nascimento (2006, p. 18), o Struts é um *framework Model View Controller* (MVC), o qual separa as camadas de apresentação, negócios e controle de fluxo, sendo isso uma característica fundamental para um sistema.

A utilização do MVC reduz a duplicação de código, centraliza o controle e torna a aplicação mais robusta, e de fácil manutenção. De acordo com o padrão *Front Controller*, o elemento de controle centralizado de uma aplicação MVC intercepta as requisições dos clientes (navegadores de Internet no caso de aplicações *Web*), seleciona e executa a lógica de negócios apropriada e redireciona a requisição para o próximo passo no fluxo da aplicação. Esse próximo passo pode ser uma página *Java Server Pages* (JSP) para a apresentação da resposta final à requisição do cliente.

No Struts, o principal elemento de controle é o *servlet ActionServlet*; essa classe lê os mapeamentos do arquivo de configuração *struts-config.xml*, chamados *action mappings*, das ações da aplicação. Esses mapeamentos são então usados pelo *ActionServlet* para transformar requisições http em ações da aplicação.

3.3.1.8 Padrões de projeto implementados

Foram utilizados três padrões de projeto no sistema, com a intenção de oferecer qualidade e praticidade, sendo eles o padrão MVC, *Facade* e *Singleton*. A seguir serão descritos detalhes dos padrões.

3.3.1.8.1 Padrão MVC

Este padrão define três camadas:

- a) *view*: formulários ou páginas JSP, é o que é apresentado ao usuário refletindo o *Model*;
- b) *controller*: responde a solicitações do usuário e despacha para o *Model*. Muitas vezes, acaba implementado na camada de aplicação para sistemas com apenas uma interface;
- c) *model*: objetos de negócio.

Este padrão foi utilizado no desenvolvimento do sistema. Sua utilização foi de grande importância para separar as classes nas camadas de visão, controle e modelo, definindo com clareza a responsabilidade de cada classe.

3.3.1.8.2 Padrão *Facade* e *Singleton*

O padrão *Facade* é muito utilizado para definir camadas de aplicação. Ele cria um ponto de entrada para um processo. Já o padrão *Singleton* é utilizado para garantir a existência de objetos únicos no sistema, não permitindo a criação de mais de um objeto da mesma classe. Ambos foram utilizados na classe *MCommerceWebFs*, conforme demonstra o

apêndice D.

Mais detalhes sobre estes padrões pode ser encontrado em Gamma (2000).

3.3.1.9 Segurança na comunicação com o Web Service

Segundo Wikipédia, HTTPS é uma implementação do protocolo HTTP sobre uma camada Secure Sockets Layer (SSL) ou do Transport Layer Security (TLS). Essa camada adicional permite que os dados sejam transmitidos através de uma conexão criptografada e que se verifique a autenticidade do servidor e do cliente através de certificados digitais. A porta usada por norma para o protocolo HTTPS é a 443. O protocolo HTTPs é normalmente utilizado quando se deseja evitar que a informação transmitida entre o cliente e o servidor seja visualizada por terceiros.

Para que o sistema utilizasse comunicação de forma segura com o Web Service, foi utilizado um certificado digital no servidor e o protocolo HTTPs para estabelecer a comunicação com o servidor de aplicações.

3.3.2 Operacionalidade da implementação

A seguir serão demonstradas as telas do módulo central e do módulo MIDP comprador, passo a passo da utilização de cada módulo.

3.3.2.1 Módulo central

Este módulo do sistema é utilizado pelo gerente e pelo vendedor, através da *Web*, por telas de cadastros a fim de disponibilizar as informações ao consumidor, no momento da

compra.

A Figura 16 apresenta a tela de *login* do sistema, para o usuário ter acesso ao sistema.



Figura 16 – Tela de login

A Figura 17 apresenta a tela de cadastro de consumidor, gerente e vendedor. Nesta tela o gerente pode fazer uma consulta específica ao informar o código, o nome e/ou o login do usuário e clicar no botão Pesquisar, conforme destacado com o número 1. Pode ser feita a exclusão de algum usuário, ao clicar na opção Excluir de algum item, conforme destacado com o número 2. Pode ser feita a inserção de um usuário, ao clicar no botão Inserir novo usuário, conforme destacado com o número 3. Pode ser feita a alteração de um usuário, ao clicar no nome do usuário, conforme destacado com o número 4, alterar a informação e clicar no botão Salvar.

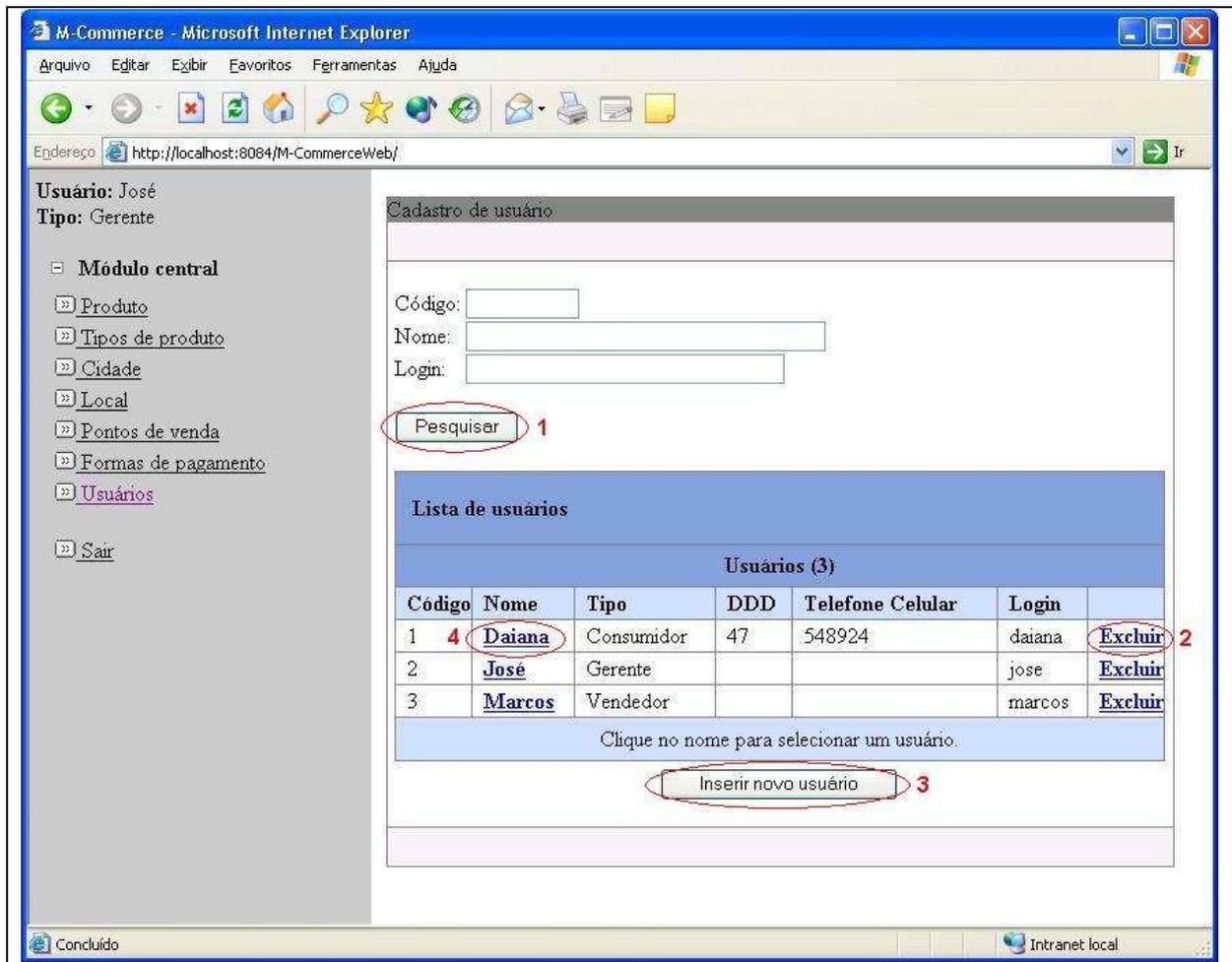


Figura 17 – Tela de cadastro de usuário

As próximas telas do módulo central, que serão apresentadas a seguir, possuem o mesmo comportamento ao consultar, excluir, incluir e alterar um registro.

A Figura 18 apresenta a tela de cadastro de formas de pagamento. Nesta tela o gerente pode fazer a inclusão, a alteração, a exclusão ou a alteração de uma forma de pagamento.

Usuário: José
Tipo: Gerente

☐ **Módulo central**

- ☐ Produto
- ☐ Tipos de produto
- ☐ Cidade
- ☐ Local
- ☐ Pontos de venda
- ☐ **Formas de pagamento**
- ☐ Usuários
- ☐ Sair

Cadastro de forma de pagamento

Código:

Tipo:

Lista de formas de pagamento

Formas de pagamento (3)

Código	Nome	
1	Cartão crédito	Excluir
2	Cartão pré pago	Excluir
3	Conta pós pago	Excluir

Clique no nome para selecionar uma forma de pagamento.

Concluído Intranet local

Figura 18 – Tela de cadastro de forma de pagamento

A Figura 19 apresenta a tela de cadastro de tipos de produto. Nesta tela o gerente pode fazer a inclusão, a alteração, a exclusão ou a alteração de um tipo de produto.

Usuário: José
Tipo: Gerente

☐ **Módulo central**

- ☐ Produto
- ☐ [Tipos de produto](#)
- ☐ Cidade
- ☐ Local
- ☐ Pontos de venda
- ☐ Formas de pagamento
- ☐ Usuários
- ☐ Sair

Cadastro de tipo de produto

Código:
Descrição:

Lista de tipos de produto

Tipos de produto (2)		
Código	Nome	
1	Refrigerante	Excluir
2	Guloseimas	Excluir

Clique no nome para selecionar um tipo de produto.

Concluído Intranet local

Figura 19 – Tela de cadastro de tipo de produto

A Figura 20 apresenta a tela de cadastro de produtos. Nesta tela o gerente pode fazer a inclusão, a alteração, a exclusão ou a alteração de um produto.

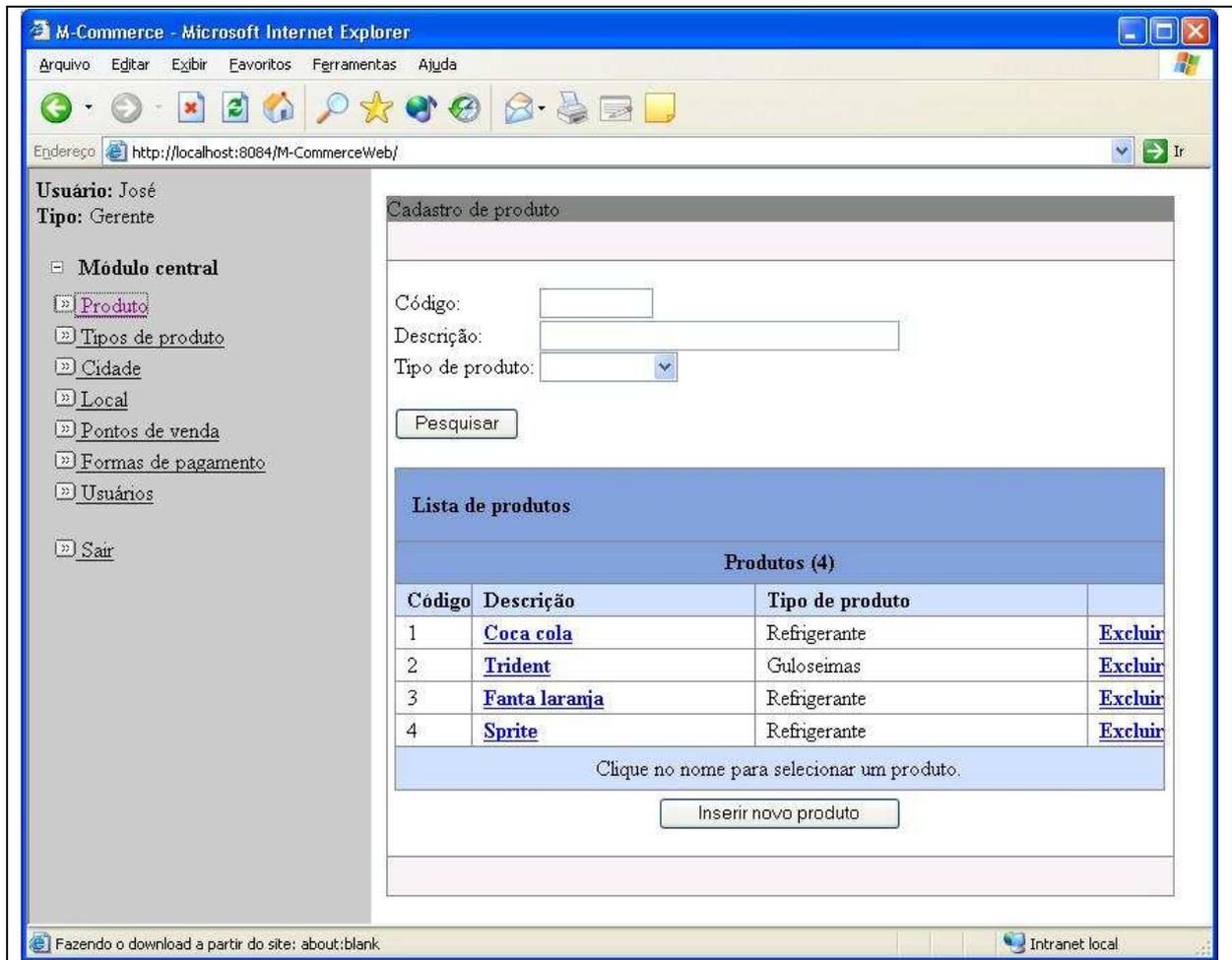


Figura 20 – Tela de cadastro de produto

A Figura 21 apresenta a tela de cadastro de cidades. Nesta tela o gerente pode fazer a inclusão, a alteração, a exclusão ou a alteração de uma cidade.

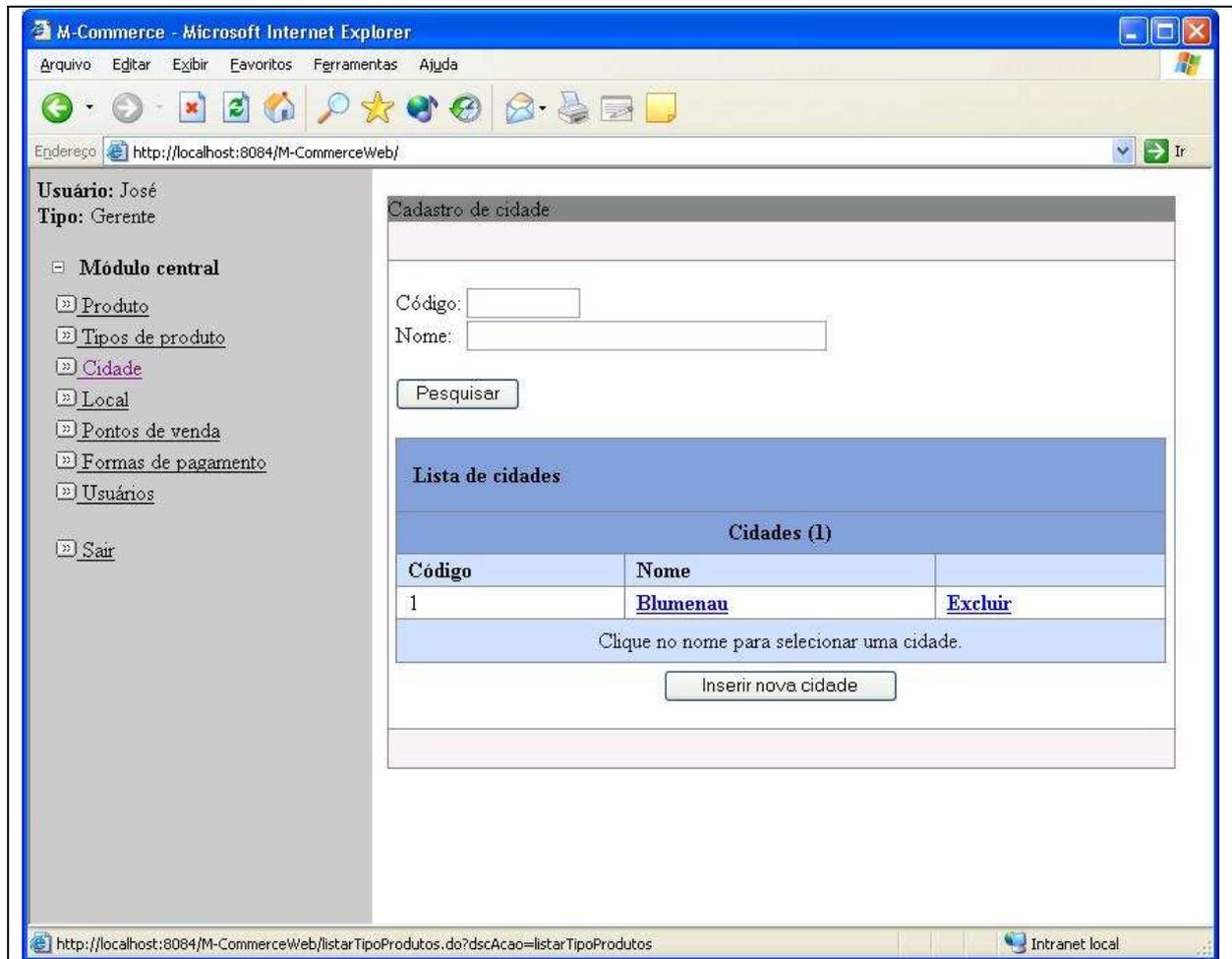


Figura 21 – Tela de cadastro de cidade

A Figura 22 apresenta a tela de cadastro de locais de ponto de venda. Nesta tela o gerente pode fazer a inclusão, a alteração, a exclusão ou a alteração de um local do ponto de venda.

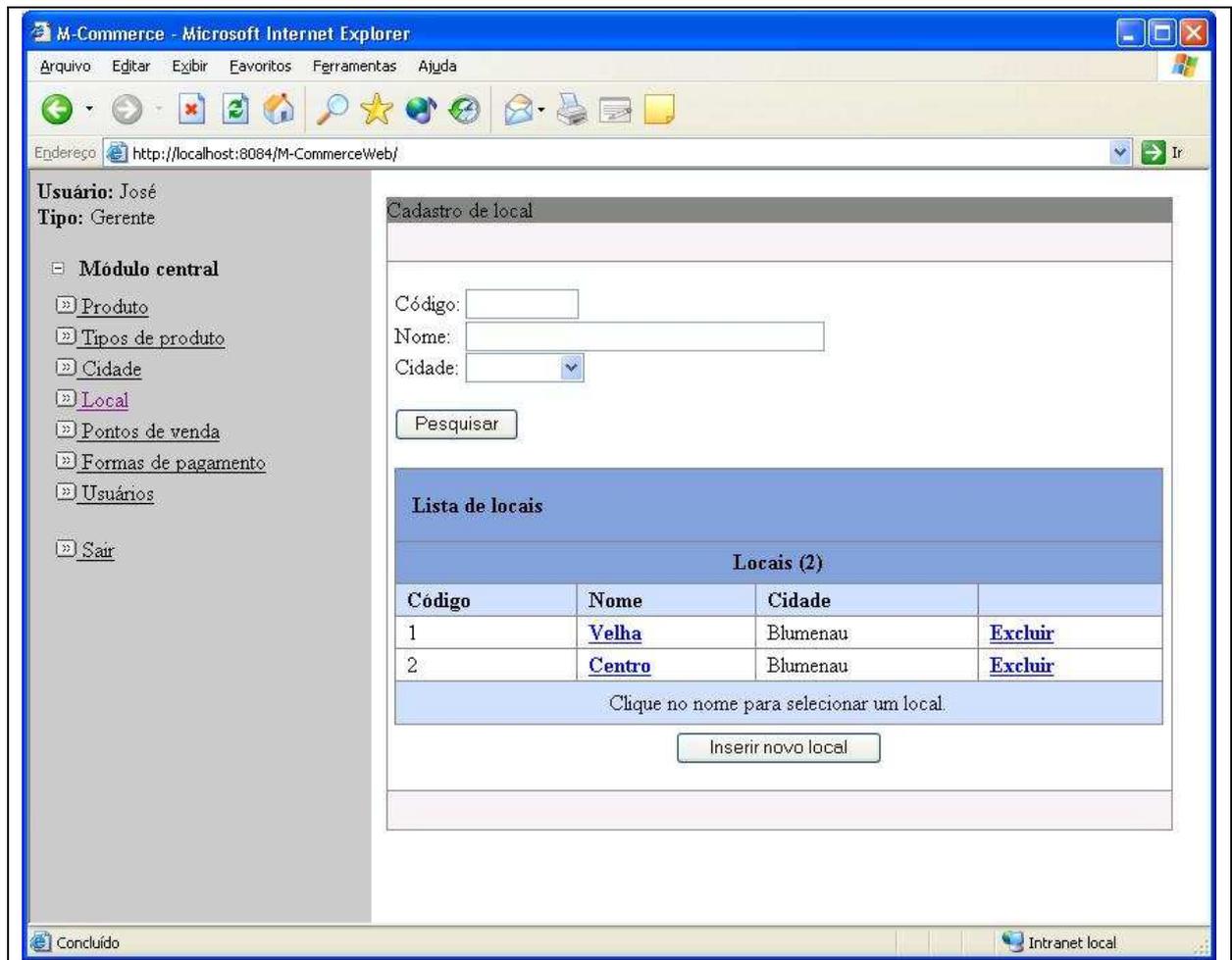


Figura 22 – Tela de cadastro de local

A Figura 23 apresenta a tela de cadastro de ponto de venda. Nesta tela o gerente pode fazer a inclusão, a alteração, a exclusão ou a alteração de um ponto de venda.

The screenshot shows a web browser window titled 'M-Commerce - Microsoft Internet Explorer'. The address bar shows 'http://localhost:8084/M-CommerceWeb/'. The page content is as follows:

Usuário: José
Tipo: Gerente

Módulo central

- Produto
- Tipos de produto
- Cidade
- Local
- Pontos de venda
- Formas de pagamento
- Usuários
- Sair

Cadastro de ponto de venda

Código:
Nome:
Local:

Lista de pontos de venda

Pontos de venda (2)

Código	Nome	Local	Cidade	
1	Supermercado Angeloni Velha	Velha	Blumenau	Excluir
2	Shopping Neumarkt	Centro	Blumenau	Excluir

Clique no nome para selecionar um ponto de venda.

At the bottom of the browser window, the status bar shows 'Concluído' and 'Intranet local'.

Figura 23 – Tela de cadastro de ponto de venda

A Figura 24 apresenta a tela de cadastro de produtos no ponto de venda. Nesta tela o vendedor pode fazer a inclusão, a alteração, a exclusão ou a alteração de um produto no ponto de venda.

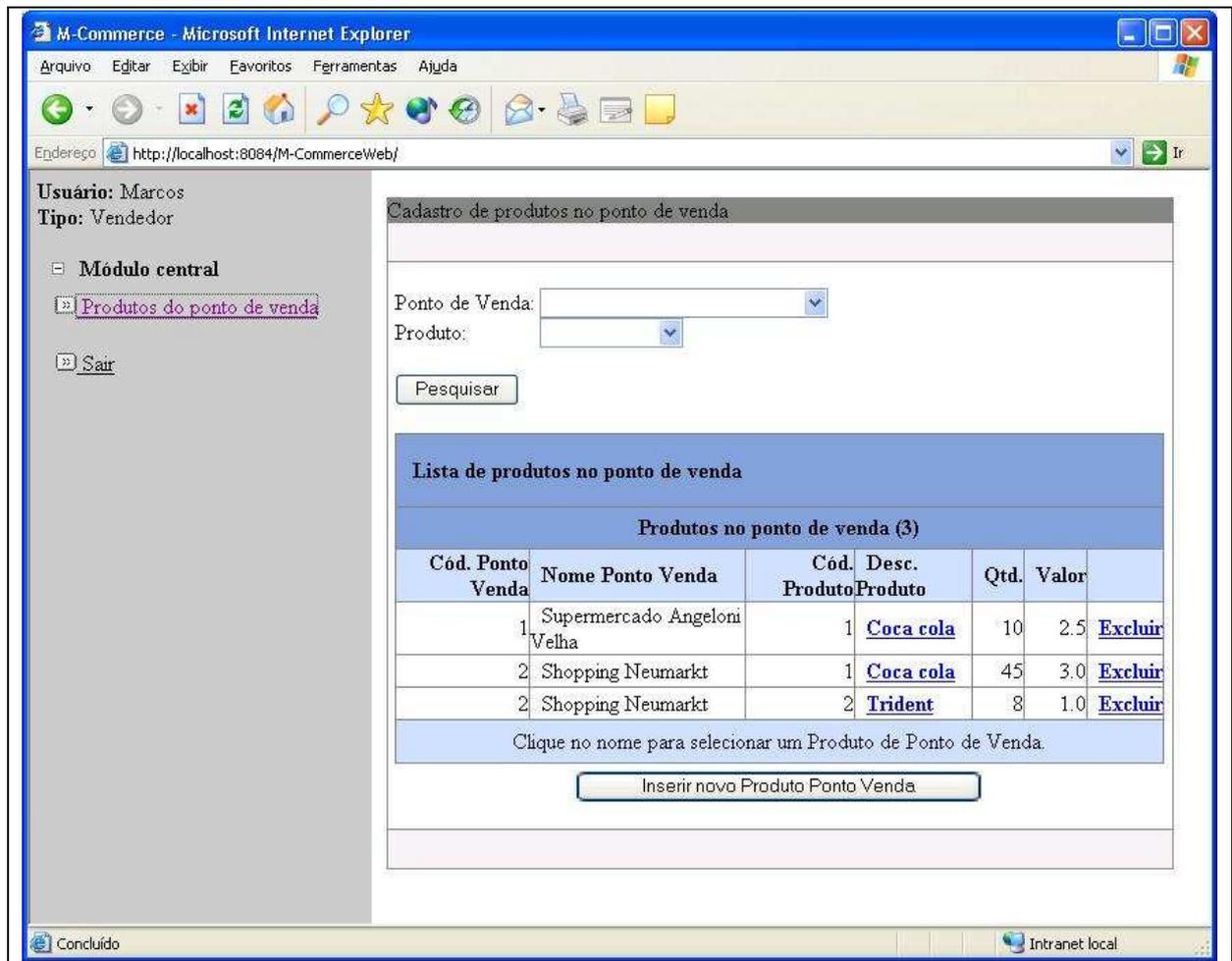


Figura 24 – Tela de cadastro de produto no ponto de venda

3.3.2.2 Módulo MIDP comprador

Este módulo do sistema é utilizado pelo consumidor, através do celular, por telas de informações necessárias para a compra do produto desejado.

A Figura 25 apresenta a tela inicial ao iniciar o módulo MIDP comprador.

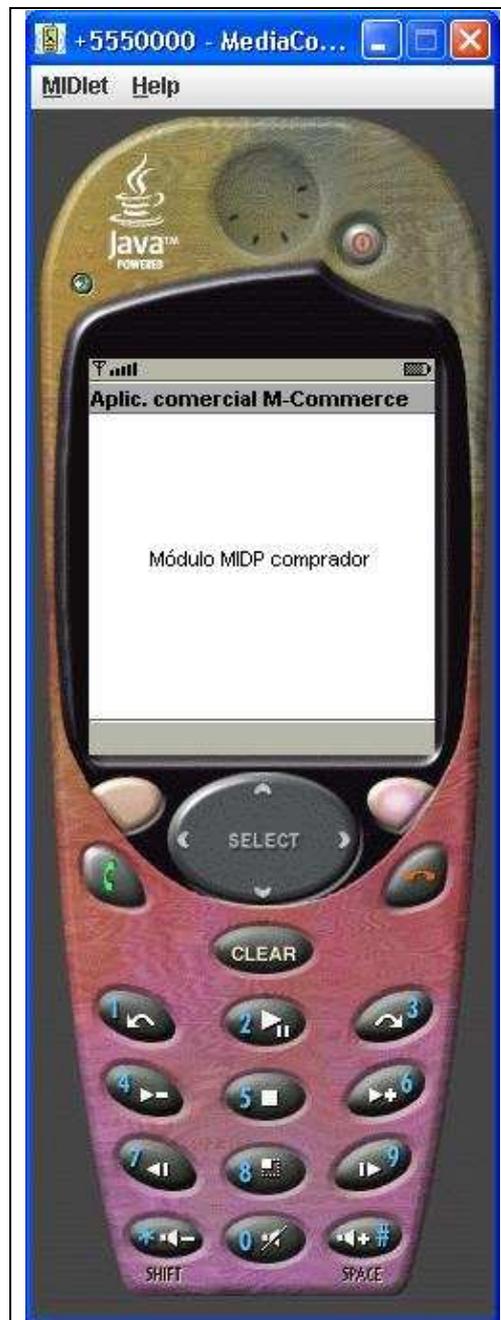


Figura 25 – Tela inicial

A Figura 26 apresenta a tela de identificação do consumidor. Nesta tela o consumidor deve informar seu usuário e sua senha.

Ele pode selecionar a opção Sair para não iniciar a compra ou pode selecionar a opção Ok para validar o consumidor. Caso o consumidor não esteja cadastrado ou o usuário não seja um consumidor irá aparecer mensagem de Usuário não liberado ou dados inválidos. Mas caso

estiver válido, o sistema irá para a tela de identificação do ponto de venda.



Figura 26 – Tela de identificação do consumidor

A Figura 27 apresenta a tela de identificação do ponto de venda. Nesta tela o consumidor deve informar o código do ponto de venda que deseja realizar a compra.

Ele pode selecionar a opção Sair para voltar à tela de identificação do consumidor ou pode selecionar a opção Ok para confirmar o ponto de venda. Caso o ponto de venda não esteja cadastrado irá apresentar mensagem de Ponto de Venda inválido. Mas caso esteja cadastrado, o sistema irá apresentar tela de aviso para confirmação do ponto de venda.



Figura 27 – Tela de identificação do ponto de venda

A Figura 28 apresenta a tela de aviso para o consumidor confirmar os dados do ponto

de venda informado, para garantir informou o ponto de venda que realmente deseja realizar a compra. Ele pode selecionar a opção *Done* para verificar os dados do ponto de venda.



Figura 28 – Tela de aviso para conferir ponto de venda

A Figura 29 apresenta a tela de conferência dos dados do ponto de venda informado para o consumidor confirmar.

Ele pode selecionar a opção *Cancelar* para voltar à tela de identificação do ponto de venda ou pode selecionar a opção *Confirma* para confirmar o ponto de venda informado.



Figura 29 – Tela para conferir ponto de venda

A Figura 30 apresenta a tela de itens para compra. Nesta tela o consumidor deve informar a quantidade do(s) produto(s) que deseja comprar. Para produtos que não deseja

comprar, deve deixar a quantidade em branco.

Ele pode selecionar a opção Cancelar para voltar à tela de identificação do ponto de venda ou pode selecionar a opção Confirma para confirmar os itens e quantidade que deseja comprar. Caso não informe nenhuma quantidade, irá apresentar mensagem de Informe a quantidade de pelo menos um item. Mas caso informe, o sistema irá apresentar tela de formas de pagamento.



Figura 30 – Tela para informar produto e quantidade

A Figura 31 apresenta a tela de formas de pagamento. Nesta tela o consumidor deve escolher a forma de pagamento da compra. Ele pode selecionar a opção Cancelar para voltar à tela de itens para compra ou pode selecionar a opção Confirma para confirmar a forma de pagamento escolhida. Ao confirmar o sistema irá apresentar tela com dados da compra.



Figura 31 – Tela para escolher forma de pagamento

A Figura 32 apresenta a tela para conferir os dados de compra e confirmar a compra. Nesta tela o consumidor deve confirmar a compra. Ele pode selecionar a opção Cancelar para voltar à tela de identificação do ponto de venda ou pode selecionar a opção Confirma para confirmar a compra. Caso cancele, irá apresentar mensagem “Sua compra foi cancelada”. Mas caso confirme, o sistema irá apresentar mensagem “Finalizando a compra”.



Figura 32 – Tela para confirmar a compra

A Figura 33 apresenta a tela que exibe a mensagem “Finalizando a compra”. Nesta tela o consumidor deve aguardar até que sua compra seja confirmada ou não.



Figura 33 – Tela de mensagem de finalizando a compra

A Figura 34 apresenta a tela de finalização da compra. Nesta tela o consumidor é informado do número da compra finalizada, caso a compra seja efetuada com sucesso. Neste momento o ponto de venda recebe um e-mail com os produtos e quantidades compradas.



Figura 34 – Tela de confirmação da compra

A Figura 35 apresenta o e-mail que o ponto de venda recebeu. Neste caso foi cadastrado um e-mail de conta pessoal no ponto de venda do sistema, para demonstração do recebimento real do e-mail, para então o produto comprado ser disponibilizado.

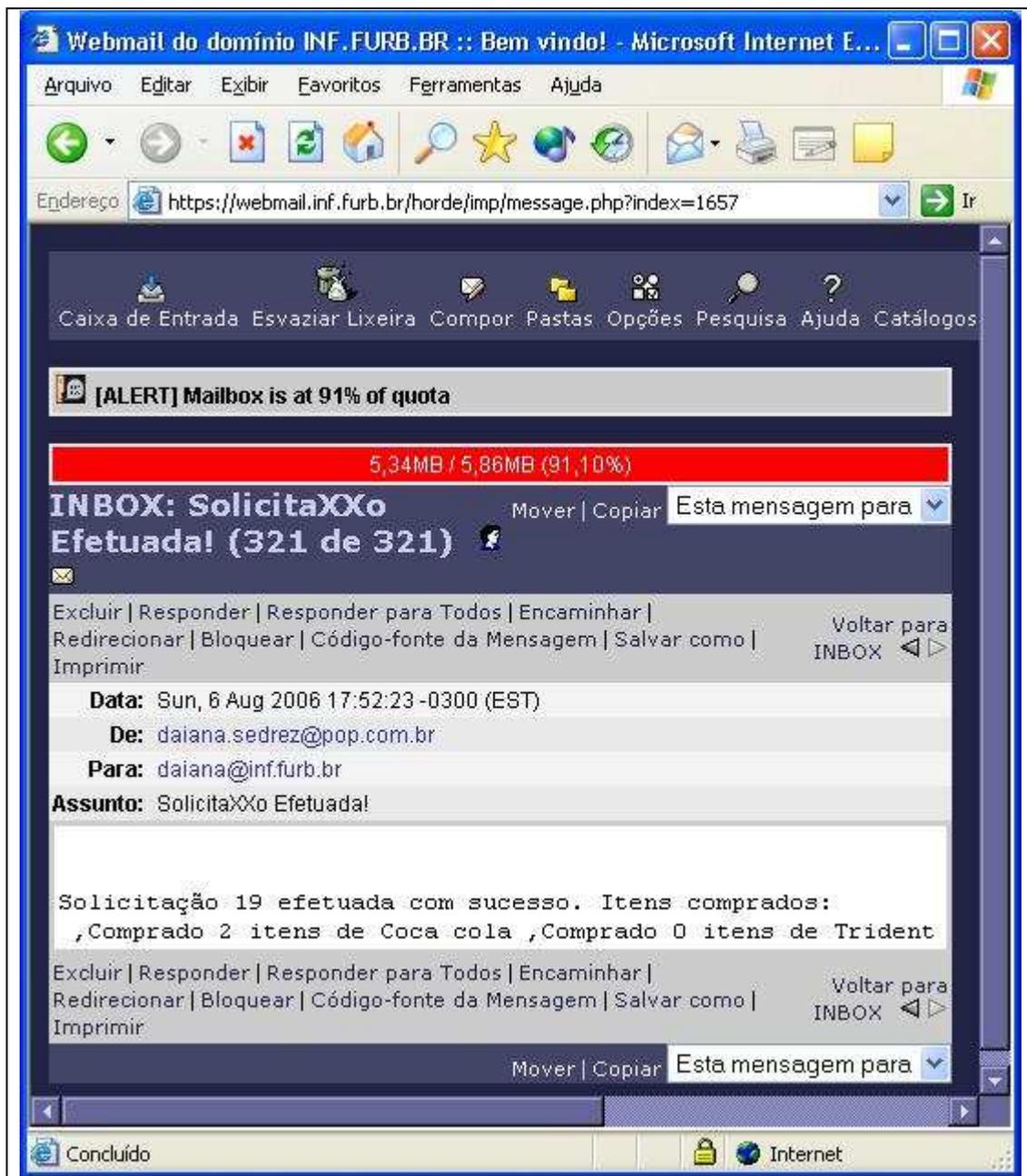


Figura 35 – Tela de e-mail recebido pelo ponto de venda

3.4 RESULTADOS E DISCUSSÃO

Com o objetivo de desenvolver uma aplicação utilizando o conceito de *M-Commerce*, o sistema se mostrou adequado.

O desenvolvimento do módulo MIDP vendedor não foi realizado. Para um dispositivo

móvel do ponto de venda receber um sinal referente à solicitação de algum produto para compra em uma determinada quantidade e disponibilizar o produto, necessitaria de um aplicativo que recebesse essas informações, interpretasse e gerasse outro sinal para o conhecimento de que o produto foi disponibilizado. Devido à complexidade que existe nesse processo e ao escopo do trabalho, optou-se por enviar ao dispositivo do ponto de venda um e-mail com os produtos e quantidades, como sinal de que foi realizada uma compra, confirmando que o produto comprado deve ser disponibilizado ao comprador.

Os testes dos requisitos do módulo central e do módulo MIDP comprador foram realizados com sucesso. Os testes do módulo MIDP comprador foram realizados em um emulador disponibilizado pela ferramenta *Sun Java Wireless Toolkit*, realizando a comunicação com o *Web Services* de forma correta. O resultado foi satisfatório, no entanto, no decorrer do trabalho houve alguns problemas como o pouco conhecimento dos assuntos tratados no tópico de ferramentas utilizadas do capítulo do desenvolvimento, como por exemplo o Hibernate, biblioteca kSOAP2, e também a troca do ambiente de desenvolvimento Eclipse para NetBeans devido a melhor interação e agilidade do último ambiente em relação ao desenvolvimento do módulo MIDP comprador. Os *design patterns* utilizados precisaram ser estudados em detalhes para ser aplicado neste trabalho, os quais proporcionaram resultados positivos, agilizando o desenvolvimento bem como organizando a estrutura do sistema na utilização dos três padrões citados no subitem técnicas e ferramentas utilizadas do item implementação.

O Quadro 3 apresenta o relacionamento de alguns trabalhos correlatos apresentados na fundamentação teórica, com este trabalho. Como pode ser observado, este trabalho trouxe alguns diferenciais em relação aos outros trabalhos, como a aplicação de um conceito novo e atual nunca antes referenciado, que se trata do *M-Commerce*, a utilização de um framework de mapeamento objeto-relacional chamado de Hibernate e a utilização de uma ferramenta

auxiliar para aplicação J2ME que se chama *NetBeans Mobility Pack*.

Funcionalidades	Este projeto	Rosa (2005)	Schmitt Junior (2004)
Aplicação desenvolvida com J2ME	Sim	Sim	Sim
Comunicação com Web Services	Sim	Sim	Não
Aplicação de Design Patterns	Sim	Sim	Sim
Armazenamento das informações do celular	Não	Não	Sim
Aplicação de um conceito de sistema atual e nunca referenciado	Sim	Não	Não
Utilização de framework de mapeamento objeto-relacional	Sim	Não	Não
Utilização de ferramenta auxiliar para aplicação J2ME	Sim	Não	Não

Quadro 3 – Funcionalidades de cada trabalho

4 CONCLUSÕES

Este trabalho é mais uma prova que o comércio móvel já faz parte do nosso dia a dia, e cada vez mais fará parte do cotidiano de todas as pessoas portadoras de um dispositivo móvel habilitado. Hoje se envia mensagens SMS, um serviço que se paga via o celular, e isto é o comércio móvel: fazer o pagamento por um produto ou serviço através do celular. A compra de refrigerantes, ingressos de cinema, ticket de estacionamento já é realizada por dispositivos móveis no exterior, com certeza logo estaremos realizando isto também no Brasil.

Os objetivos do trabalho não foram completamente atendidos, pois não ocorreu o desenvolvimento do módulo MIDP vendedor, ou seja, o módulo que executaria no dispositivo do ponto de venda não foi implementado como originalmente especificado. Foi apenas implementado para o dispositivo móvel do ponto de venda receber um e-mail com os itens e quantidades dos produtos, finalizando o ciclo de compra.

O ambiente de desenvolvimento foi alterado no decorrer no trabalho, conforme detalhado no item resultados e discussões, para melhorar o tempo de desenvolvimento e simplificar a implementação. Outras ferramentas, técnicas e *design patterns* proporcionaram bons resultados durante sua utilização, tornando-se possível a conclusão com resultados positivos ao implementar um sistema com base no conceito de *M-Commerce* disponibilizando mais uma fonte de pesquisas aos interessados.

4.1 EXTENSÕES

As principais sugestões de extensões e modificações para este trabalho são as seguintes:

- a) implementar no módulo central uma consulta das solicitações de compra, para o

gerente ter controle dos produtos comprados;

- b) implementar módulo MIDP para o ponto de venda, para o ponto de venda obter conhecimento de recebimento de um pedido, ler os itens comprados e então disponibilizar os itens comprados.

REFERÊNCIAS BIBLIOGRÁFICAS

ALENCAR, P. Brasil tem 89,4 milhões de celulares. **Plantão INFO**, São Paulo, abr. 2006. Disponível em: <<http://info.abril.com.br/aberto/infonews/042006/24042006-5.shl>>. Acesso em: 04 maio 2006.

ALEXANDER, S. D. D.; SOARES, C. C.; VIANA, D. F. B. **MobileShop**: projeto de uma ferramenta de m-commerce para vendas de produtos. 2002. 124 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro de Ciências Exatas e Tecnológicas, Universidade da Amazônia, Belém.

ALMEIDA, L. B. Introdução à J2ME e programação MIDP. **Mundo Java**, Rio de Janeiro, n. 5, p. 20-27, 2004.

BARBOSA, A. **Dispositivos móveis poderão servir como porta-moedas eletrônicos ou instrumentos de finalização financeira**. [S.l], abr. 2006. Disponível em: <<http://www.estadao.com.br/tecnologia/internet/noticias/2006/abr/28/145.htm>>. Acesso em: 03 maio 2006.

CALL TO CALL. **O cliente CTBC pode comprar refrigerante pelo celular**. [S.l], jul. 2002. Disponível em: <<http://www.calltocall.com.br/agenda.asp?ID=225&cat=Call%20to%20Call%20News>>. Acesso em: 12 set. 2005.

CUNHA, D. **Web services, SOAP e aplicações web**. [S.l]: dez. 2002. Disponível em: <http://devedge-temp.mozilla.org/viewsource/2002/soap-overview/index_pt_br.html>. Acesso em: 11 out. 2005.

DORNAN, A. **Wireless communication**: o guia essencial de comunicação sem fio. Tradução Fábio Freitas. Rio de Janeiro: Campus, 2001.

FRANCO JUNIOR, C. F. **E-business**: Internet, tecnologia e sistemas de informação na administração de empresas. 3. ed. São Paulo: Atlas, 2005.

GAMMA, Erich. **Padrões de projeto: soluções reutilizáveis de software orientado a objetos**. Porto Alegre : Bookman, 2000. xii, 364p, il. Tradução de: Design patterns: elements of reusable object-oriented software.

KNUDSEN, Jonathan. **Wireless Java**: developing with J2ME. 2.ed. New York: Apress, 2003, xviii, 364 p.

LINHARES, M. **Introdução ao hibernate 3**. [S.l]: [2005?]. Disponível em: <<http://www.guj.com.br/java/tutorial.artigo.174.1.guj>>. Acesso em: 03 mar. 2006.

MENÉNDEZ, A. I. M.; NASCIMENTO, G. S. Framework MVC. **Mundo Java**, Rio de Janeiro, n. 9, p.18-25, 2006.

MENEZES, R. **Câmara Brasileira de Comércio Eletrônico**. São Paulo, [2003?]. Disponível em: <<http://www.camara-e.net/newsletter/2004/newsletter01setembro04.htm>>. Acesso em: 12 set. 2005.

MONTENEGRO, C.; PEREIRA, C. Java de ponta a ponta do J2ME ao J2EE. **Mundo Java**, Rio de Janeiro, n. 12, p. 28-43, 2005.

MOSKORZ, R. R. **M-Commerce: estratégias para difusão e implantação**. 2002. 118f. Dissertação (Mestrado em Engenharia de Produção) – Programa de Pós-graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis.

MUCHOW, J. W. **Core J2ME tecnologia e MIDP**. Tradução João Eduardo Nóbrega Tortello. São Paulo: Pearson Makron Books, 2004.

OBSERVATÓRIO DO COMÉRCIO. **O futuro do “m-commerce”**. [S.l], [2000?]. Disponível em: <<http://www.dgcc.pt/164.htm#51>>. Acesso em: 12 set. 2005.

PAMPLONA, V. F. **Um protótipo de motor de jogos 3D para dispositivos móveis com suporte a especificação mobile 3D graphics API for J2ME**. 2005. 83f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

PAMPLONA, V. F. Web services via J2SE e J2ME. **iMasters**, [S.l], nov. 2004. Disponível em: <<http://www.imasters.com.br/artigo.php?cn=2741&cc=151>>. Acesso em 11 out. 2005.

PIO, F. **GoWap desenvolve sistema que permite comprar refrigerante via celular**. São Paulo, dez. 2001. Disponível em: <http://www.panoramabrasil.com.br/noticia_completa.asp?p=conteudo/txt/2001/12/21/20096759.htm>. Acesso em 12 set. 2005.

PRADO, E. **A explosão dos serviços móveis**. [S.l], set. 2005. Disponível em: <<http://old.idgnow.com.br/AdPortalv5/ColunistaInterna.aspx?GUID=BE4F5E04-499C-4003-9867-D7CD59D29817&ChannelID=21080130>>. Acesso em 03 maio 2006.

REUTERS. **Carteiras “sem fio” aproximam-se da realidade nos EUA**. [S.l], jul. 2005. Disponível em: <<http://tecnologia.terra.com.br/interna/0,,OI610100-EI4803,00.html>>. Acesso em: 12 set. 2005.

RODRIGO, P. **B2B, B2C, B2G: entendendo a sopa de letras**, [S.l], jul. 2005. Disponível em: <http://www.imasters.com.br/artigo/3371/ecommerce/b2b_b2c_b2g_entendendo_a_sopa_de_letras>. Acesso em 23 maio 2006.

ROSA, F. **Protótipo de um diário de classe em dispositivos móveis utilizando J2ME**. 2005. 99 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

RYDLEWSKI, C. A vida sem fio. **Veja**, São Paulo, v. 1874, n. 40, p. 101-110, out. 2004.

SCHMITT JUNIOR, A. J. **Protótipo de front end de controle de acesso usando J2ME**. 2004. 69 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

TEIXEIRA, C. A. **Dois bilhões de celulares no mundo**. [S.l], out. 2005. Disponível em: <<http://www.torque.com.br/index.php?modulo=textos&secao=artigos&codTexto=1653&pagina=1&sequencia=1&codCategoria=#>>. Acesso em 03 maio 2006.

WIKIPÉDIA. **Protocolo HTTPS**. [S.l], [2005?]. Disponível em: <<http://pt.wikipedia.org/wiki/HTTPS>>. Acesso em 14 jun 2006.

APÊNDICE A – Especificação dos casos de uso

Objetivo	Efetuar login no módulo central
Ator	Usuário
Pré-condições	O usuário deve estar cadastrado no sistema
Cenário Principal	<ol style="list-style-type: none"> 1. O sistema solicita login e senha 2. O usuário informa login e senha 3. O sistema valida login e senha
Cenário de Exceção	Se no passo 3, login e senha estiverem inválidos, o sistema apresenta mensagem “Usuário/Senha Incorreta.”
Pós-condições	O sistema mostra opções de menu

Quadro 4 – UC01.01 Identificar usuário

Objetivo	Manutenção dos consumidores
Ator	Gerente
Pré-condições	O gerente deve estar logado no sistema
Cenário Principal	1. O gerente requisita o cadastro dos consumidores; 2. O sistema apresenta as opções que podem ser realizadas: inclusão, alteração, exclusão ou consulta de um consumidor; 3. O gerente seleciona a opção desejada ou finaliza o caso de uso.
Cenário Alternativo: Incluir um consumidor	a. O gerente solicita a inclusão de um consumidor b. O sistema apresenta os campos a serem informados c. O gerente informa os dados necessários e solicita a inclusão do consumidor d. O sistema valida os dados informados. e. Se os dados informados estiverem válidos o sistema exibe mensagem de consumidor incluída com sucesso; caso contrário, exibe mensagem de erro
Cenário Alternativo: Alterar um consumidor	a. O gerente solicita a alteração de um consumidor cadastrada b. O sistema apresenta os campos com os dados do consumidor escolhida c. O gerente informa os dados a serem alterados e solicita a alteração do consumidor d. O sistema valida os dados informados e. Se os dados informados estiverem válidos o sistema exibe mensagem de consumidor alterada com sucesso; caso contrário, exibe mensagem de erro
Cenário Alternativo: Excluir um consumidor	a. O gerente solicita a exclusão de um consumidor cadastrada b. O sistema apresenta mensagem se deseja realmente excluir o consumidor c. Se o gerente confirmar a exclusão o consumidor será excluído
Cenário Alternativo: Consultar um consumidor	a. O gerente solicita a consulta de um consumidor cadastrada b. O gerente informa os dados que deseja consultar e solicita que o sistema mostre o consumidor de acordo com os dados informados.
Pós-condições	De acordo com a opção escolhida pelo gerente o sistema incluiu, alterou, excluiu ou consultou um consumidor

Quadro 5 – UC01.02 Cadastrar consumidor

Objetivo	Manutenção dos usuários
Ator	Gerente
Pré-condições	O gerente deve estar logado no sistema
Cenário Principal	<ol style="list-style-type: none"> 1. O gerente requisita o cadastro dos gerentes / vendedores; 2. O sistema apresenta as opções que podem ser realizadas: inclusão, alteração, exclusão ou consulta de um gerente / vendedor; 3. O gerente seleciona a opção desejada ou finaliza o caso de uso.
Cenário Alternativo: Incluir um gerente / vendedor	<ol style="list-style-type: none"> a. O gerente solicita a inclusão de um gerente / vendedor b. O sistema apresenta os campos a serem informados c. O gerente informa os dados necessários e solicita a inclusão do gerente / vendedor d. O sistema valida os dados informados. e. Se os dados informados estiverem válidos o sistema exibe mensagem de gerente / vendedor incluída com sucesso; caso contrário, exibe mensagem de erro
Cenário Alternativo: Alterar um gerente / vendedor	<ol style="list-style-type: none"> a. O gerente solicita a alteração de um gerente / vendedor cadastrada b. O sistema apresenta os campos com os dados do gerente / vendedor escolhida c. O gerente informa os dados a serem alterados e solicita a alteração do gerente / vendedor d. O sistema valida os dados informados e. Se os dados informados estiverem válidos o sistema exibe mensagem de gerente / vendedor alterada com sucesso; caso contrário, exibe mensagem de erro
Cenário Alternativo: Excluir um gerente / vendedor	<ol style="list-style-type: none"> a. O gerente solicita a exclusão de um gerente / vendedor cadastrada b. O sistema apresenta mensagem se deseja realmente excluir o gerente / vendedor c. Se o gerente confirmar a exclusão o gerente / vendedor será excluído
Cenário Alternativo: Consultar um gerente / vendedor	<ol style="list-style-type: none"> a. O gerente solicita a consulta de um gerente / vendedor cadastrada b. O gerente informa os dados que deseja consultar e solicita que o sistema mostre o gerente / vendedor de acordo com os dados informados.
Pós-condições	De acordo com a opção escolhida pelo gerente o sistema incluiu, alterou, excluiu ou consultou um gerente / vendedor

Quadro 6 – UC01.03 Cadastrar gerente / vendedor

Objetivo	Manutenção das formas de pagamento
Ator	Gerente
Pré-condições	O gerente deve estar logado no sistema
Cenário Principal	<ol style="list-style-type: none"> 1. O gerente requisita o cadastro das formas de pagamento; 2. O sistema apresenta as opções que podem ser realizadas: inclusão, alteração, exclusão ou consulta de uma forma de pagamento; 3. O gerente seleciona a opção desejada ou finaliza o caso de uso.
Cenário Alternativo: Incluir uma forma de pagamento	<ol style="list-style-type: none"> a. O gerente solicita a inclusão de uma forma de pagamento b. O sistema apresenta os campos a serem informados c. O gerente informa os dados necessários e solicita a inclusão da forma de pagamento d. O sistema valida os dados informados. e. Se os dados informados estiverem válidos o sistema exibe mensagem de forma de pagamento incluída com sucesso; caso contrário, exibe mensagem de erro
Cenário Alternativo: Alterar uma forma de pagamento	<ol style="list-style-type: none"> a. O gerente solicita a alteração de uma forma de pagamento cadastrada b. O sistema apresenta os campos com os dados da forma de pagamento escolhida c. O gerente informa os dados a serem alterados e solicita a alteração da forma de pagamento d. O sistema valida os dados informados e. Se os dados informados estiverem válidos o sistema exibe mensagem de forma de pagamento alterada com sucesso; caso contrário, exibe mensagem de erro
Cenário Alternativo: Excluir uma forma de pagamento	<ol style="list-style-type: none"> a. O gerente solicita a exclusão de uma forma de pagamento cadastrada b. O sistema apresenta mensagem se deseja realmente excluir a forma de pagamento c. Se o gerente confirmar a exclusão a forma de pagamento será excluída
Cenário Alternativo: Consultar uma forma de pagamento	<ol style="list-style-type: none"> a. O gerente solicita a consulta de uma forma de pagamento cadastrada b. O gerente informa os dados que deseja consultar e solicita que o sistema mostre a forma de pagamento de acordo com os dados informados.
Pós-condições	De acordo com a opção escolhida pelo gerente o sistema incluiu, alterou, excluiu ou consultou uma forma de pagamento

Quadro 7 – UC01.04 Cadastrar formas de pagamento

Objetivo	Manutenção dos tipos de produto
Ator	Gerente
Pré-condições	O gerente deve estar logado no sistema
Cenário Principal	<ol style="list-style-type: none"> 1. O gerente requisita o cadastro dos tipos de produto 2. O sistema apresenta as opções que podem ser realizadas: inclusão, alteração, exclusão ou consulta de um tipo de produto 3. O gerente seleciona a opção desejada ou finaliza o caso de uso.
Cenário Alternativo: Incluir um tipo de produto	<ol style="list-style-type: none"> a. O gerente solicita a inclusão de um tipo de produto b. O sistema apresenta os campos a serem informados c. O gerente informa os dados necessários e solicita a inclusão do tipo de produto d. O sistema valida os dados informados. e. Se os dados informados estiverem válidos o sistema exibe mensagem de tipo de produto incluído com sucesso; caso contrário, exibe mensagem de erro
Cenário Alternativo: Alterar um tipo de produto	<ol style="list-style-type: none"> a. O gerente solicita a alteração de um tipo de produto cadastrado b. O sistema apresenta os campos com os dados do tipo de produto escolhido c. O gerente informa os dados a serem alterados e solicita a alteração do tipo de produto d. O sistema valida os dados informados e. Se os dados informados estiverem válidos o sistema exibe mensagem de tipo de produto alterado com sucesso; caso contrário, exibe mensagem de erro
Cenário Alternativo: Excluir um tipo de produto	<ol style="list-style-type: none"> a. O gerente solicita a exclusão de um tipo de produto cadastrado b. O sistema apresenta mensagem se deseja realmente excluir o tipo de produto c. Se o gerente confirmar a exclusão o tipo de produto será excluído
Cenário Alternativo: Consultar um tipo de produto	<ol style="list-style-type: none"> a. O gerente solicita a consulta de um tipo de produto cadastrado b. O gerente informa os dados que deseja consultar e solicita que o sistema mostre o tipo de produto de acordo com os dados informados.
Pós-condições	De acordo com a opção escolhida pelo gerente o sistema incluiu, alterou, excluiu ou consultou um tipo de produto

Quadro 8 – UC01.05 Cadastrar tipos de produto

Objetivo	Manutenção dos produtos
Ator	Gerente
Pré-condições	O gerente deve estar logado no sistema
Cenário Principal	<ol style="list-style-type: none"> 1. O gerente requisita o cadastro dos produtos 2. O sistema apresenta as opções que podem ser realizadas: inclusão, alteração, exclusão ou consulta de um produto 3. O gerente seleciona a opção desejada ou finaliza o caso de uso.
Cenário Alternativo: Incluir um produto	<ol style="list-style-type: none"> a. O gerente solicita a inclusão de um produto b. O sistema apresenta os campos a serem informados c. O gerente informa os dados necessários e solicita a inclusão do produto d. O sistema valida os dados informados. e. Se os dados informados estiverem válidos o sistema exibe mensagem de produto incluído com sucesso; caso contrário, exibe mensagem de erro
Cenário Alternativo: Alterar um produto	<ol style="list-style-type: none"> a. O gerente solicita a alteração de um produto cadastrado b. O sistema apresenta os campos com os dados do produto escolhido c. O gerente informa os dados a serem alterados e solicita a alteração do produto d. O sistema valida os dados informados e. Se os dados informados estiverem válidos o sistema exibe mensagem de produto alterado com sucesso; caso contrário, exibe mensagem de erro
Cenário Alternativo: Excluir um produto	<ol style="list-style-type: none"> a. O gerente solicita a exclusão de um produto cadastrado b. O sistema apresenta mensagem se deseja realmente excluir o produto c. Se o gerente confirmar a exclusão o produto será excluído
Cenário Alternativo: Consultar um produto	<ol style="list-style-type: none"> a. O gerente solicita a consulta de um produto cadastrado b. O gerente informa os dados que deseja consultar e solicita que o sistema mostre o produto de acordo com os dados informados.
Pós-condições	De acordo com a opção escolhida pelo gerente o sistema incluiu, alterou, excluiu ou consultou um produto

Quadro 9 – UC01.06 Cadastrar produtos para venda

Objetivo	Manutenção das cidades
Ator	Gerente
Pré-condições	O gerente deve estar logado no sistema
Cenário Principal	<ol style="list-style-type: none"> 1. O gerente requisita o cadastro das cidades 2. O sistema apresenta as opções que podem ser realizadas: inclusão, alteração, exclusão ou consulta de uma cidade 3. O gerente seleciona a opção desejada ou finaliza o caso de uso.
Cenário Alternativo: Incluir uma cidade	<ol style="list-style-type: none"> a. O gerente solicita a inclusão de uma cidade b. O sistema apresenta os campos a serem informados c. O gerente informa os dados necessários e solicita a inclusão da cidade d. O sistema valida os dados informados. e. Se os dados informados estiverem válidos o sistema exibe mensagem de cidade incluída com sucesso; caso contrário, exibe mensagem de erro
Cenário Alternativo: Alterar uma cidade	<ol style="list-style-type: none"> a. O gerente solicita a alteração de uma cidade cadastrada b. O sistema apresenta os campos com os dados da cidade escolhida c. O gerente informa os dados a serem alterados e solicita a alteração da cidade d. O sistema valida os dados informados e. Se os dados informados estiverem válidos o sistema exibe mensagem de cidade alterada com sucesso; caso contrário, exibe mensagem de erro
Cenário Alternativo: Excluir uma cidade	<ol style="list-style-type: none"> a. O gerente solicita a exclusão de uma cidade cadastrada b. O sistema apresenta mensagem se deseja realmente excluir a cidade c. Se o gerente confirmar a exclusão a cidade será excluída
Cenário Alternativo: Consultar uma cidade	<ol style="list-style-type: none"> a. O gerente solicita a consulta de uma cidade cadastrada b. O gerente informa os dados que deseja consultar e solicita que o sistema mostre a cidade de acordo com os dados informados.
Pós-condições	De acordo com a opção escolhida pelo gerente o sistema incluiu, alterou, excluiu ou consultou uma cidade

Quadro 10 – UC01.07 Cadastrar cidade para o local de ponto de venda

Objetivo	Manutenção dos locais dos pontos de venda
Ator	Gerente
Pré-condições	O gerente deve estar logado no sistema
Cenário Principal	<ol style="list-style-type: none"> 1. O gerente requisita o cadastro dos locais 2. O sistema apresenta as opções que podem ser realizadas: inclusão, alteração, exclusão ou consulta de um local 3. O gerente seleciona a opção desejada ou finaliza o caso de uso.
Cenário Alternativo: Incluir um local	<ol style="list-style-type: none"> a. O gerente solicita a inclusão de um local b. O sistema apresenta os campos a serem informados c. O gerente informa os dados necessários e solicita a inclusão da cidade d. O sistema valida os dados informados. e. Se os dados informados estiverem válidos o sistema exibe mensagem de local incluído com sucesso; caso contrário, exibe mensagem de erro
Cenário Alternativo: Alterar um local	<ol style="list-style-type: none"> a. O gerente solicita a alteração de um local cadastrado b. O sistema apresenta os campos com os dados da cidade escolhido c. O gerente informa os dados a serem alterados e solicita a alteração da cidade d. O sistema valida os dados informados e. Se os dados informados estiverem válidos o sistema exibe mensagem de local alterado com sucesso; caso contrário, exibe mensagem de erro
Cenário Alternativo: Excluir um local	<ol style="list-style-type: none"> a. O gerente solicita a exclusão de um local cadastrado b. O sistema apresenta mensagem se deseja realmente excluir o local c. Se o gerente confirmar a exclusão o local será excluído
Cenário Alternativo: Consultar um local	<ol style="list-style-type: none"> a. O gerente solicita a consulta de um local cadastrado b. O gerente informa os dados que deseja consultar e solicita que o sistema mostre o local de acordo com os dados informados.
Pós-condições	De acordo com a opção escolhida pelo gerente o sistema incluiu, alterou, excluiu ou consultou um local

Quadro 11 – UC01.08 Cadastrar local para ponto de venda

Objetivo	Manutenção dos pontos de venda
Ator	Gerente
Pré-condições	O gerente deve estar logado no sistema
Cenário Principal	<ol style="list-style-type: none"> 1. O gerente requisita o cadastro dos pontos de venda 2. O sistema apresenta as opções que podem ser realizadas: inclusão, alteração, exclusão ou consulta de um ponto de venda 3. O gerente seleciona a opção desejada ou finaliza o caso de uso.
Cenário Alternativo: Incluir um ponto de venda	<ol style="list-style-type: none"> a. O gerente solicita a inclusão de um ponto de venda b. O sistema apresenta os campos a serem informados c. O gerente informa os dados necessários e solicita a inclusão do ponto de venda d. O sistema valida os dados informados. e. Se os dados informados estiverem válidos o sistema exibe mensagem de ponto de venda incluído com sucesso; caso contrário, exibe mensagem de erro
Cenário Alternativo: Alterar um ponto de venda	<ol style="list-style-type: none"> a. O gerente solicita a alteração de um ponto de venda cadastrado b. O sistema apresenta os campos com os dados do ponto de venda escolhido c. O gerente informa os dados a serem alterados e solicita a alteração do ponto de venda d. O sistema valida os dados informados e. Se os dados informados estiverem válidos o sistema exibe mensagem de ponto de venda alterado com sucesso; caso contrário, exibe mensagem de erro
Cenário Alternativo: Excluir um ponto de venda	<ol style="list-style-type: none"> a. O gerente solicita a exclusão de um ponto de venda cadastrado b. O sistema apresenta mensagem se deseja realmente excluir o ponto de venda c. Se o gerente confirmar a exclusão o ponto de venda será excluído
Cenário Alternativo: Consultar um ponto de venda	<ol style="list-style-type: none"> a. O gerente solicita a consulta de um ponto de venda cadastrado b. O gerente informa os dados que deseja consultar e solicita que o sistema mostre o ponto de venda de acordo com os dados informados.
Pós-condições	De acordo com a opção escolhida pelo gerente o sistema incluiu, alterou, excluiu ou consultou um ponto de venda

Quadro 12 – UC01.09 Cadastrar pontos de venda

Objetivo	Manutenção dos produtos em um ponto de venda
Ator	Vendedor
Pré-condições	O vendedor deve estar logado no sistema
Cenário Principal	<ol style="list-style-type: none"> 1. O vendedor requisita o cadastro dos produtos no ponto de venda 2. O sistema apresenta as opções que podem ser realizadas: inclusão, alteração, exclusão ou consulta de um produto no ponto de venda 3. O vendedor seleciona a opção desejada ou finaliza o caso de uso.
Cenário Alternativo: Incluir um produto no ponto de venda	<ol style="list-style-type: none"> a. O vendedor solicita a inclusão de um produto no ponto de venda b. O sistema apresenta os campos a serem informados c. O vendedor informa os dados necessários e solicita a inclusão do produto no ponto de venda d. O sistema valida os dados informados. e. Se os dados informados estiverem válidos o sistema exibe mensagem de produto no ponto de venda incluído com sucesso; caso contrário, exibe mensagem de erro
Cenário Alternativo: Alterar um produto no ponto de venda	<ol style="list-style-type: none"> a. O vendedor solicita a alteração de um produto no ponto de venda cadastrado b. O sistema apresenta os campos com os dados do produto no ponto de venda escolhido c. O vendedor informa os dados a serem alterados e solicita a alteração do produto no ponto de venda d. O sistema valida os dados informados e. Se os dados informados estiverem válidos o sistema exibe mensagem de produto no ponto de venda alterado com sucesso; caso contrário, exibe mensagem de erro
Cenário Alternativo: Excluir um produto no ponto de venda	<ol style="list-style-type: none"> a. O vendedor solicita a exclusão de um produto no ponto de venda cadastrado b. O sistema apresenta mensagem se deseja realmente excluir o produto no ponto de venda c. Se o vendedor confirmar a exclusão o produto no ponto de venda será excluído
Cenário Alternativo: Consultar um produto no ponto de venda	<ol style="list-style-type: none"> a. O vendedor solicita a consulta de um produto no ponto de venda cadastrado b. O vendedor informa os dados que deseja consultar e solicita que o sistema mostre o produto no ponto de venda de acordo com os dados informados.
Pós-condições	De acordo com a opção escolhida pelo vendedor o sistema incluiu, alterou, excluiu ou consultou um produto no ponto de venda

Quadro 13 – UC01.10 Selecionar produtos para venda

Objetivo	Efetuar login no módulo MIDP comprador
Ator	Usuário
Pré-condições	O usuário deve estar cadastrado no sistema
Cenário Principal	1. O sistema solicita login e senha 2. O usuário informa login e senha 3. O sistema valida login e senha
Cenário de Exceção	Se no passo 3, login e senha estiverem inválidos, o sistema apresenta mensagem “Usuário não liberado ou dados inválidos”
Pós-condições	O sistema mostra tela para selecionar ponto de venda

Quadro 14 – UC02.01 Identificar consumidor

Objetivo	Efetuar compra no módulo MIDP comprador
Ator	Consumidor
Pré-condições	O usuário deve estar cadastrado no sistema como consumidor
Cenário Principal	1. O consumidor requisita iniciar uma compra 2. O sistema apresenta tela de identificação do ponto de venda 3. O consumidor informa o ponto de venda e confirma 4. O sistema apresenta tela de aviso para conferir ponto de venda 5. O sistema apresenta tela de conferência do ponto de venda 6. O sistema apresenta tela para selecionar item que deseja comprar e sua respectiva quantidade 7. O consumidor informa o item e quantidade desejada para compra 8. O sistema apresenta tela de escolha da forma de pagamento 9. O consumidor escolhe uma forma de pagamento 10. O sistema exibe dados da compra 11. O consumidor confirma a compra 12. O sistema envia solicitação para o ponto de venda 13. O ponto de venda recebe a solicitação e disponibiliza o produto
Cenário de Exceção	Se no passo 3: o ponto de venda estiver inválido, o sistema apresenta mensagem “Ponto de venda inválido”; não ser informado o ponto de venda, o sistema apresenta mensagem “Ponto de venda deve ser informado”; sair da tela de identificação, retorna ao passo 1. Se no passo 5: cancelar a conferência, retorna ao passo 2. Se no passo 7: não informar quantidade em nenhum item, o sistema apresenta mensagem “Informe a quantidade de pelo menos um item”; cancelar a tela de definir quantidade dos itens, retornar ao passo 2. Se no passo 9: cancelar a tela de escolha de forma de pagamento, retorna ao passo 6. Se no passo 11: cancela a compra, o sistema apresenta mensagem “Sua compra foi cancelada” e retorna ao passo 2. Se no passo 12: Algum problema de envio ocorrer, o sistema apresenta mensagem “Não foi possível realizar a compra” e retorna ao passo 11.
Pós-condições	O consumidor recebe o produto comprado

Quadro 15 – UC02.02 Efetuar compra

APÊNDICE B – Utilização da biblioteca kSOAP2

Implementação de um método da classe MCommerceClientWS que utiliza a biblioteca kSOAP2.

```

        public Hashtable retornaDadosUsuario(String login, String
password) throws Exception {
            //System.out.println("MCommerceWS >>> retornaDadosUsuario():
" + login + " - " + password);

            this.http = new HttpTransport(url);
            this.http.debug = false;

            SoapObject method = new SoapObject("urn:soap-call",
"retornaDadosUsuario");
            method.addProperty("login", login);
            method.addProperty("password", password);
            SoapSerializationEnvelope envelope = new
SoapSerializationEnvelope(SoapEnvelope.VER10);
            envelope.bodyOut = method;

            try {
                this.http.call(null, envelope);
            } catch(Exception e) {
                e.printStackTrace ();
                throw e;
            }
        }
    }

```

APÊNDICE C – Arquivo de mapeamento do Hibernate

Arquivo de mapeamento PsCidade.hbm.xml.

```

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE hibernate-mapping PUBLIC
    "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">

<hibernate-mapping package="br.com.furb.mcommerce.web.model"
>
    <class
        name="PsCidade"
    >

        <id
            name="codigo"
            column="codigo"
            type="int"
            unsaved-value="null"
        >
            <generator class="increment">
            </generator>
        </id>

        <property
            name="nome"
            type="java.lang.String"
            update="true"
            insert="true"
            column="nome"
            not-null="true"
        />

        <set
            name="locais"
            lazy="true"
            inverse="true"
            cascade="none"
            sort="unsorted"
        >

            <key
                column="codigo_cidade"
                not-null="true"
            >
            </key>

            <one-to-many
                class="PsLocal"
            />

        </set>

    </class>

</hibernate-mapping>

```

Implementação de um método da classe MCommerceWebFs que busca os dados do banco, utilizando o Hibernate.

```

        public ArrayList getArrayListCidades(String codCidade, String
nomCidade) throws Exception {
            System.out.println("MCommerceWebFs >>> getArrayListCidades():
" + codCidade + " - " + nomCidade);

            Session session = HibernateUtility.getSession ();
            ArrayList al = new ArrayList();

            Criteria crit = session.createCriteria(PsCidade.class);
            if (codCidade != null && !codCidade.trim ().equals ("")) {
                crit.add( Expression.eq( "codigo", new
Integer(codCidade)) );
            }
            if (nomCidade != null && !nomCidade.trim ().equals ("")) {
                crit.add( Expression.like ( "nome", "%" + nomCidade +
"%" ) );
            }
            crit.addOrder (Order.asc ("codigo"));
            List lst = crit.list();

            Iterator it = lst.iterator();
            while (it.hasNext()) {
                HashMap hm = new HashMap();
                PsCidade cidade = (PsCidade) it.next();
                hm.put("COD_CIDADE", new Integer(cidade.getCodigo ()));
                hm.put("NOM_CIDADE", cidade.getNome ());

                al.add(hm);
            }

            session.close ();

            return al;
        }

```

APÊNDICE D – Padrão Facade e Singleton

```

/**
 * Classe fachada para o modelo de dados
 */
public class MCommerceWebFs {

    private static MCommerceWebFs instance;

    private MCommerceWebFs() {
    }

    public static MCommerceWebFs getInstance() {
        if (instance == null) {
            instance = new MCommerceWebFs();
        }
        return instance;
    }

    public Vector retornaDadosUsuario(String login, String password)
throws Exception {
        //IMPLEMENTAÇÃO
    }

    public Vector retornaDadosPontoVenda(String codPontoVenda) throws
Exception {
        //IMPLEMENTAÇÃO
    }

    public Vector retornaItensPontoVenda(String codPontoVenda) throws
Exception {
        //IMPLEMENTAÇÃO
    }

    public Vector retornaFormaPagtos(String codUsuario) throws
Exception {
        //IMPLEMENTAÇÃO
    }

    public String finalizaCompra(String codUsuario, String
codPontoVenda, Vector lstCodItens, Vector lstQtdItens, String
codFormaPagto) throws Exception {
        //IMPLEMENTAÇÃO
    }

    private void sendEmail(String to, String subject, String body)
throws IOException {
        //IMPLEMENTAÇÃO
    }

    public ArrayList getArrayListTipoProdutos(String codTipo, String
dscTipo) throws Exception {
        //IMPLEMENTAÇÃO
    }

    public void insereTipoProduto (String dscTipoProduto) throws
Exception {
        //IMPLEMENTAÇÃO
    }
}

```

```

    }

    public void excluiTipoProduto (String codTipoProduto) throws
Exception {
        //IMPLEMENTAÇÃO
    }

    public void alteraTipoProduto (String codTipoProduto, String
dscTipoProduto) throws Exception {
        //IMPLEMENTAÇÃO
    }

    public HashMap getDadosTipoProduto (String codTipoProduto) throws
Exception {
        //IMPLEMENTAÇÃO
    }

    public ArrayList getArrayListCidades(String codCidade, String
nomCidade) throws Exception {
        //IMPLEMENTAÇÃO
    }

    public void insereCidade (String nomCidade) throws Exception {
        //IMPLEMENTAÇÃO
    }

    public void excluiCidade (String codCidade) throws Exception {
        //IMPLEMENTAÇÃO
    }

    public void alteraCidade (String codCidade, String nomCidade)
throws Exception {
        //IMPLEMENTAÇÃO
    }

    public HashMap getDadosCidade (String codCidade) throws Exception
{
        //IMPLEMENTAÇÃO
    }

    public ArrayList getArrayListFormaPagtos(String codFormaPagto,
String tipoFormaPagto) throws Exception {
        //IMPLEMENTAÇÃO
    }

    public void insereFormaPagto (String tipoFormaPagto) throws
Exception {
        //IMPLEMENTAÇÃO
    }

    public void excluiFormaPagto (String codFormaPagto) throws
Exception {
        //IMPLEMENTAÇÃO
    }

    public void alteraFormaPagto (String codFormaPagto, String
tipoFormaPagto) throws Exception {
        //IMPLEMENTAÇÃO
    }

    public HashMap getDadosFormaPagto (String codFormaPagto) throws

```

```

Exception {
    //IMPLEMENTAÇÃO
}

    public ArrayList getArrayListUsuarios(String codUsuario, String
nomUsuario, String login) throws Exception {
        //IMPLEMENTAÇÃO
    }

    public void insereUsuario (String nomUsuario, String ddd, String
celular, String login, String password, String tipo, String[]
formasPagtoSel) throws Exception {
        //IMPLEMENTAÇÃO
    }

    public void excluiUsuario (String codUsuario) throws Exception {
        //IMPLEMENTAÇÃO
    }

    public void alteraUsuario (String codUsuario, String nomUsuario,
String ddd, String celular, String login, String password, String tipo,
String[] formasPagtoSel) throws Exception {
        //IMPLEMENTAÇÃO
    }

    public HashMap getDadosUsuario (String codUsuario) throws
Exception {
        //IMPLEMENTAÇÃO
    }

    public HashMap getInfoUsuario (String login) throws Exception {
        //IMPLEMENTAÇÃO
    }

    public ArrayList getArrayListProdutos(String codProduto, String
dscProduto, String tipoProduto) throws Exception {
        //IMPLEMENTAÇÃO
    }

    public void insereProduto (String dscProduto, String tipoProduto)
throws Exception {
        //IMPLEMENTAÇÃO
    }

    public void excluiProduto (String codProduto) throws Exception {
        //IMPLEMENTAÇÃO
    }

    public void alteraProduto (String codProduto, String dscProduto,
String tipoProduto) throws Exception {
        //IMPLEMENTAÇÃO
    }

    public HashMap getDadosProduto (String codProduto) throws
Exception {
        //IMPLEMENTAÇÃO
    }

    public ArrayList getArrayListLocais(String codLocal, String
nomLocal, String cidadeLocal) throws Exception {
        //IMPLEMENTAÇÃO
    }

```

```

    }

    public void insereLocal (String nomLocal, String cidadeLocal)
throws Exception {
        //IMPLEMENTAÇÃO
    }

    public void excluiLocal (String codLocal) throws Exception {
        //IMPLEMENTAÇÃO
    }

    public void alteraLocal (String codLocal, String nomLocal, String
cidadeLocal) throws Exception {
        //IMPLEMENTAÇÃO
    }

    public HashMap getDadosLocal (String codLocal) throws Exception {
        //IMPLEMENTAÇÃO
    }

    public ArrayList getArrayListPontosVenda(String codPontoVenda,
String nomPontoVenda, String localPontoVenda) throws Exception {
        //IMPLEMENTAÇÃO
    }

    public void inserePontoVenda (String nomPontoVenda, String
localPontoVenda, String telefone, String email) throws Exception {
        //IMPLEMENTAÇÃO
    }

    public void excluiPontoVenda (String codPontoVenda) throws
Exception {
        //IMPLEMENTAÇÃO
    }

    public void alteraPontoVenda (String codPontoVenda, String
nomPontoVenda, String localPontoVenda, String telefone, String email)
throws Exception {
        //IMPLEMENTAÇÃO
    }

    public HashMap getDadosPontoVenda (String codPontoVenda) throws
Exception {
        //IMPLEMENTAÇÃO
    }

    public ArrayList getArrayListProdutosPontoVenda(String
codPontoVenda, String codProduto) throws Exception {
        //IMPLEMENTAÇÃO
    }

    public void insereProdutoPontoVenda (String codPontoVenda, String
codProduto, String quantidade, String valor) throws Exception {
        //IMPLEMENTAÇÃO
    }

    public void excluiProdutoPontoVenda (String codPontoVenda, String
codProduto) throws Exception {
        //IMPLEMENTAÇÃO
    }

```

```
        public void alteraProdutoPontoVenda (String codPontoVenda, String
codProduto, String quantidade, String valor) throws Exception {
            //IMPLEMENTAÇÃO
        }

        public HashMap getDadosProdutoPontoVenda (String codPontoVenda,
String codProduto) throws Exception {
            //IMPLEMENTAÇÃO
        }
    }
```

APÊNDICE E – Código gerado pelo módulo NetBeans Mobility Pack

O código destacado foi gerado automaticamente pelo módulo, não sendo permitido sua edição, com exceção dos comentários feitos pelo módulo; e o código sem destaque foi implementação para personalizar a aplicação. Para alterar o código gerado deve ser alterado através do fluxo, nas propriedades existentes para cada componente.

```

        public void commandAction(Command command, Displayable
displayable) {
            // Insert global pre-action code here
            if (displayable == frmLogin) {
                if (command == cmdOk) {
                    // Insert pre-action code here
                    if (get_txtSenha().getString().equals("") |
get_txtUsuario().getString().equals("")) {
                        Alert alErro = new Alert("Atenção", "Usuário e
senha devem ser informados", null, AlertType.WARNING);
                        alErro.setTimeout(-2);
                        alErro.setCommandListener(new CommandListener() {
                            public void commandAction(Command command,
Displayable displayable) {
                                getDisplay().setCurrent(get_frmLogin ());
                            }
                        });
                        getDisplay().setCurrent(alErro);
                        return;
                    }
                    getDisplay().setCurrent(get_wscrLogin());
                    // Insert post-action code here
                } else if (command == cmdSair) {
                    // Insert pre-action code here
                    exitMIDlet();
                    // Insert post-action code here
                }
            }
        }

```

```

public Form get_frmLogin() {
    if (frmLogin == null) {
        // Insert pre-init code here
        frmLogin = new Form("Login", new Item[] {
            get_txtUsuario(),
            get_txtSenha()
        });
        frmLogin.addCommand(get_cmdOk());
        frmLogin.addCommand(get_cmdSair());
        frmLogin.setCommandListener(this);
        // Insert post-init code here
    }
    return frmLogin;
}

```