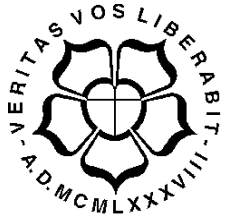
UNIVERSIDADE LUTERANA DO BRASIL
FACULDADE DE INFORMÁTICA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
CAMPUS GRAVATAÍ



ESTUDO SOBRE A INTERNET MÓVEL E O *M-COMMERCE*

Rodrigo Barcelos Lessa

Monografia desenvolvida durante a disciplina de Trabalho de Conclusão de Curso em Informática II e apresentada à Faculdade de Informática da Universidade Luterana do Brasil, campus Gravataí, como pré-requisito para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Leandro Bento Pompermaier

Gravataí, julho de 2001.

Universidade Luterana do Brasil – ULBRA
Faculdade de Informática
Curso de Bacharelado em Ciência da Computação – Campus Gravataí

Reitor:
Pastor Ruben Eugen Becker
Vice-Reitor:
Eng. Leandro Eugênio Becker
Diretor do Campus Gravataí:
Prof. Felício Korb
Diretor da Faculdade de Informática:
Prof. Miguel Rodrigues Fornari
Coordenador de Curso (Campus Gravataí):
Prof. Marco Antônio da Rocha
Coordenador das Disciplinas de Trabalho de Conclusão de Curso (Campus Gravataí):
Prof. Roland Teodorowitsch

Banca Avaliadora composta por: *Data da defesa: 11/07/2001.*
Prof. Leandro Bento Pompermaier *(Orientador)*
Prof. Vinicius Gadis Ribeiro
Prof. Marco Antônio da Rocha

Revisão Ortográfica: ***Gelson Peres***

CIP – Catalogação na Publicação

Lessa, Rodrigo Barcelos

Estudo sobre a Internet Móvel e o *M-commerce* / Rodrigo Barcelos Lessa. – Gravataí: Faculdade de Informática da ULBRA, 2001.
xiii, 88p.

Trabalho de Conclusão de Curso (Graduação) – Universidade Luterana do Brasil, Faculdade de Informática, Curso de Bacharelado em Ciência da Computação, Gravataí, 2001. Orientador: Prof. Leandro Bento Pompermaier.

1. Internet Móvel. 2. *M-Commerce*. 3. Redes *Wireless*. I. Trabalho de Conclusão de Curso. II. Lessa, Rodrigo Barcelos. III. Título.

Endereço:
Universidade Luterana do Brasil – Campus Gravataí
Estrada Itacolomi, 3.600 – Bairro São Vicente
CEP 94170-240 Gravataí-RS – Brasil

AGRADECIMENTOS

Agradeço a Deus que me concedeu tudo de bom até hoje, e principalmente nesses meses de trabalho intenso me concedendo forças para não desistir. Agradeço por tudo de bom que aconteceu na minha vida até hoje. Pelos dias e dias que fique usando o meu computador, pesquisando na Internet e lendo os mais diversos livros para fazer esse trabalho de conclusão. Agradeço também o apoio da minha família, namorada e amigos por me agüentarem durante esse período, acordando às 6 horas da manhã e virando os finais de semana na frente do computador.

Quero agradecer aos meus pais, José Carlos Fraga Lessa e Nilsa Barcelos Lessa, por terem me dado o presente mais importante que uma pessoa pode ganhar, a possibilidade de estudar e adquirir novos conhecimentos.

Também quero agradecer a todos os meus professores que me possibilitaram ter um crescimento acadêmico, profissional e principalmente pessoal. Em especial, agradeço ao professor Leandro, por me orientar nesse trabalho, pois sem as suas idéias e conselhos eu não conseguiria finalizar o trabalho proposto. Quero agradecer também aos professores Vinícius e Marco por participarem da minha banca, pelas melhorias sugeridas no TCI e também pelos conhecimentos que os dois me passaram durante o decorrer do curso.

Muito obrigado a todos.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

ABRANET	Associação Brasileira dos Provedores de Acesso
ANATEL	Agência Nacional de Telecomunicações
ANSI	*American National Standards Institute*
API	*Application Programming Interface*
ASCII	*American Standard Code for Information Interchange*
ASP	*Active Server Pages*
B2B	*Business-to-Business*
B2C	*Business-to-Consumer*
B2E	*Business-to-Employee*
CDMA	*Code Division Multiple Access*
CDPD	*Cellular Digital Packet Data*
CGI	*Common Gateway Interface*
CORBA	*Common Object Request Broker Architecture*
DTD	*Document Type Definitions*
ERP	*Enterprise Resources Planning*
FTP	*File Transfer Protocol*
GSM	*Global System for Mobile Communication*
HDML	*Handheld Markup Language*
HTML	*Hyper Text Markup Language*
HTTP	*Hyper Text Transport Protocol*
IDC	*International Data Corporation*
IIS	*Internet Information Server*
ISO	*International Organization for Standardization*
JSP	*Java Server Pages*
ODBC	*Open Data Base Connectivity*
OO	Orientação a Objetos

PDA	*Portable Data Assistent*
PDU	*Protocol Data Unit*
PIN	*Personal Identification Number*
RFC	*Request For Comments*
SMS	*Short Message Service*
SSL	*Secure Sockets Layer*
TCP/IP	*Transmission Control Protocol/Internet Protocol*
TDMA	*Time Division Multiple Access*
UML	*Unified Modeling Language*
URI	*Uniform Resource Identifier*
UDP	*Unreliable Datagram Protocol*
URL	*Uniform Resource Locator*
W3C	*World Wide Web Consortium*
WAE	*Wireless Application Environment*
WAP	*Wireless Application Protocol*
WBMP	*Wireless Bitmap*
WDP	*Wireless Datagram Protocol*
WML	*Wireless Markup Language*
WSP	*Wireless Session Protocol*
WTA	*Wireless Telephony Application*
WTAI	*Wireless Telephony Application Interface*
WTLS	*Wireless Transport Layer Security*
WTP	*Wireless Transport Protocol*
WWW	*World Wide Web*
XML	*Extensible Markup Language*

RESUMO

O trabalho realiza um estudo sobre a Internet Móvel, principalmente sobre o protocolo WAP e sobre o *M-commerce.* Os estudos se basearam na bibliografia existente e em estudos de caso sobre aplicações de comércio eletrônico em redes *wireless.* Para atingir todos os objetivos do trabalho foi implementado um protótipo de *M-commerce* para avaliar as potencialidades, os problemas e as técnicas utilizadas com essa tecnologia. Os resultados obtidos foram à identificação dos problemas relacionados com a segurança do modelo de programação, a limitação das redes móveis e dos dispositivos móveis, as técnicas para desenvolver um site de *M-commerce* e as potencialidades de novas aplicações e utilização futura da Internet Móvel. Com a realização do trabalho foi possível identificar os recursos que estão disponíveis, o que precisa ser implementado para possibilitar o desenvolvimento de sistemas realmente diferenciado, os problemas da tecnologia móvel, o que está sendo desenvolvido com WAP, o avanço nas tecnologias móveis como a 3G e como implementar um sistema para a Internet Móvel desde a análise até o desenvolvimento.

ABSTRACT

The work accomplishes a study on Internet Mobile of furniture, mainly on the protocol WAP and on M-commerce. The studies based on the existent bibliography and in studies of in case about applications of electronic trade in nets wireless. To reach all the objectives of the work a prototype of M-commerce it was implemented to evaluate the potentialities, the problems and the techniques used with that technology. The obtained results went to the identification of the problems related with the safety of the programming model, the limitation of the movable nets and of the movable devices, the techniques to develop a site of M-commerce and the potentialities of new applications and future use of Internet Mobile of furniture. With the accomplishment of the work it was possible to identify the resources that are available, the one that needs be implemented to make possible the development of systems really differentiated, the problems of the technology mobile, what is being developed with WAP, the progress in the movable technologies as to 3G and how to implement a system for Internet Mobile of furniture from the analysis to the development.

1 INTRODUÇÃO

A Internet influenciou muito nas mudanças que ocorreram no mundo nos últimos anos, onde a informação está disponível a em todos os lugares a um simples clique, e as novas aplicações, como comércio eletrônico *B2B, B2C, B2E, C2C, WebEdi, e-procurement, Marketplace*, etc., que reduzem custos, melhoram os processos dos negócios e oferecem novas oportunidades de ganhos em novos mercados. Porém, mesmo com a popularização da Internet, o número de usuários ainda é pequeno na grande maioria dos países, principalmente nos mais pobres. E isso acontece por diversas causas, como pouca oferta de linhas telefônicas convencionais, o preço dos computadores, o custo do treinamento na utilização dos computadores, etc.

Para aumentar o número de usuários, foram criadas novas tecnologias para possibilitar o acesso à Internet sem ser preciso o uso de computadores. Entre elas a *WebTV*, o acesso via console de videogame e outros aparelhos eletrônicos. Mas foi com o surgimento de tecnologias *wireless*, que permitem que aparelhos móveis, como um telefone celular, *palmtop*, *PDA* acessem à Internet. Possibilitando assim, o acesso para milhares de novos usuários, principalmente via telefone celular, aparelho que é utilizado por muitos usuários.

Com a evolução da tecnologia *wireless*, principalmente no desempenho e segurança, o *M-commerce* será uma realidade, e com ele novos tipos de negócios, serviços e possibilidades de aumentar a competitividade das empresas. O *M-commerce* atende àquele consumidor que deseja realizar a sua compra em qualquer lugar, a qualquer hora.

1.1 MOTIVAÇÃO

A evolução da informática e das telecomunicações é algo constante e muito rápido. Com essa evolução, surgiu um novo termo, uma nova fase na economia mundial, chamada de Nova Economia. Essa Nova Economia se baseia na informação, principalmente baseada na Tecnologia da Informação e, claro, na Internet que vem tendo um grande crescimento nos últimos três anos.

Mas a Internet vai crescer muito mais, e não vai ser com o aumento das vendas de computadores pessoais, mas com novos equipamentos de acesso à Rede, como telefones celulares, PDAs, etc. Esses dispositivos são chamados de equipamentos de informação, pois é só ligá-los e acessar a Internet.

Existem várias pesquisas em desenvolvimento para acessar a Internet via *wireless*, como a 3G(Terceira Geração de Celulares) que está em testes pela NTT DoCoMo no Japão e por outras empresas na Europa, e podem atingir a velocidade de 2 *Mbps* na transmissão de dados. Mas a tecnologia que surgiu como padrão para acesso à Internet com dispositivos

móveis é o *WAP*, um protocolo desenvolvido pelo *WAP Fórum*, em 1997, que é formado pelas grandes empresas de telecomunicações no mundo, entre elas Motorola, Nokia, Ericsson. Com essa tecnologia o usuário pode navegar na Internet, enviar e receber e-mail, fazer compras, pagar contas, realizar transações bancárias, automatizar processos de pré-venda em venda on-line, etc.

Com a tecnologia *WAP*, é possível acessar páginas no padrão *WML*, que são criadas especialmente para as telas dos telefones celulares ou *palmtops* que tenham um *microbrowser* que reconheça a *WML*. Como na *Web* padrão, é só clicar nos *links* para navegar nas páginas *WML*.

Como a utilização de um telefone celular é feita por um grande número de pessoas, a sua utilização para acessar a Internet promete modificar os hábitos e as atitudes da maioria dessas pessoas. Na Figura 1, segundo a Anatel, pesquisas indicam que em 2003, existirão no Brasil mais de 45,5 milhões de usuários de telefones celulares.

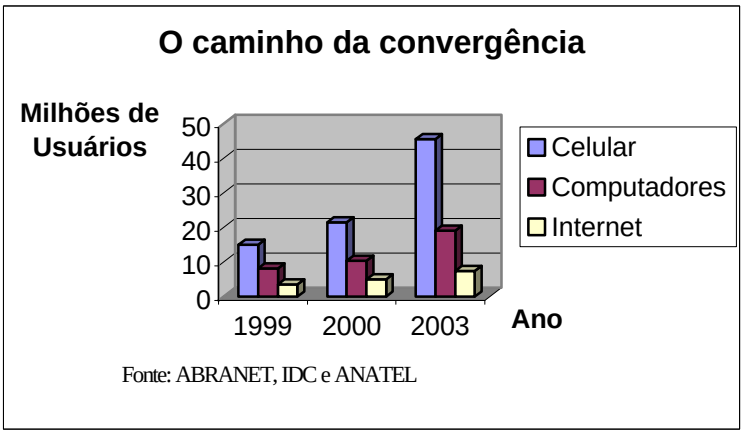


Figura 1 – Dados sobre o Mercado de Tecnologia no Brasil

Com o crescimento do acesso a Internet via *wireless*, a maioria das empresas terão que fazer com que suas aplicações Web (notícias, E-*commerce*, *webmail*) possam ser acessadas via *wireless*, pois elas terão que atender esses usuários que responderão pela metade dos acessos à Internet, e representarão mais de 50% dos usuários que realizam negócios na Internet, conforme o IDC.

Portanto, será analisada a tecnologia *WAP* juntamente com as tecnologias utilizadas para o desenvolvimento de sites de *E-commerce*, e a possível utilização na construção de um *website* de *M-commerce*, e onde serão analisados aspectos como segurança nas transações, o desempenho de acesso, a facilidade de uso, as técnicas para realizar a análise e o desenvolvimento e também os problemas dessa tecnologia.

1.2 OBJETIVOS

O objetivo principal desse trabalho é avaliar o acesso à Internet por equipamentos que utilizam tecnologia *wireless*, principalmente o protocolo *WAP*, que vem se destacando como padrão nesse tipo de tecnologia, como também fazer um estudo sobre a construção de soluções de comércio eletrônico para Internet acessada via *wireless*.

Este objetivo será alcançado através dos seguintes objetivos específicos:

- Estudo sobre o protocolo Wireless Application Protocol;
- Viabilizar um ambiente que utilize Internet para implantar serviços da Web tradicional e Wireless utilizando tecnologias *open source*;
- Estudar as ferramentas utilizadas para criar sites de M-commerce, como banco de dados MySQL rodando em Linux com servidor Web Apache, e tecnologias como PHP, WML, WMLscript e XML;
- Fazer a análise, projeto e implementação de um protótipo de site de M-commerce. Para tal, o negócio a ser implantado será escolhido no final da primeira etapa deste trabalho, principalmente para se ter conhecimento da tecnologia que será empregada e a possível utilização dos recursos da Ulbra para esse desenvolvimento;
- Avaliar a usabilidade do aplicativo WML, os aspectos relacionados com a segurança do aplicativo e os aspectos positivos e negativos da tecnologia wireless.

1.3 ORGANIZAÇÃO DO TEXTO

O trabalho está organizado na seguinte forma: o primeiro capítulo possui a Introdução do trabalho, a motivação da realização desse trabalho, os objetivos esperados e a organização do texto.

No segundo capítulo, é mostrado um estudo aprofundado das tecnologias w*ireless*, dando um enfoque especial no *Wireless Application Protocol*, que vem se tornando padrão rapidamente nessa área. Também será abordada a pesquisa sobre novas tecnologias *wireless*, sobre as futuras aplicações que serão disponibilizadas, e a integração com os microcomputadores.

No terceiro capítulo, é feita uma análise sobre a linguagem WML que é utilizada com o protocolo WAP, e suas diferenças e semelhanças com a linguagem HMTL.

No quarto capítulo, são apresentados os conceitos sobre o comércio eletrônico, o *M-commerce* é enfatizado, as técnicas utilizadas, as vantagens e desvantagens em relação ao comércio eletrônico tradicional.

No quinto capítulo, as técnicas e mecanismos de segurança nos sites de *M-commerce* são apresentados.

No sexto capítulo, é apresentado o projeto de *M-commerce* proposto, chamado E-Móvel.

Por último, no sétimo capítulo, são apresentadas as considerações finais do TC.

2 WIRELESS APPLICATION PROTOCOL

O *Wireless Application Protocol* foi desenvolvido pelo WAP Fórum fundado em 1997, que é formado pelas empresas de comunicação Nokia, Ericsson, Motorola, entre outras. Suas especificações possibilitam o desenvolvimento de aplicações e serviços para redes de comunicação *wireless*. O *WAP* especifica a estrutura e os protocolos de rede para dispositivos móveis, tais como telefones celulares, *pagers*, *PDAs*, etc.

A arquitetura Internet *World Wide Web* proporciona um modelo de programação poderoso, como pode ser visto na Figura 2, onde está representado o modelo de programação WWW (Fórum, 1999). O conteúdo e as aplicações são apresentados em um formato padrão de dados, e visualizados através de um *Web browser*. O *Web browser* é uma aplicação de rede, ele envia requisições de objetos de dados para um servidor de rede, e esse servidor responde com os dados codificados, usando um formato padrão.

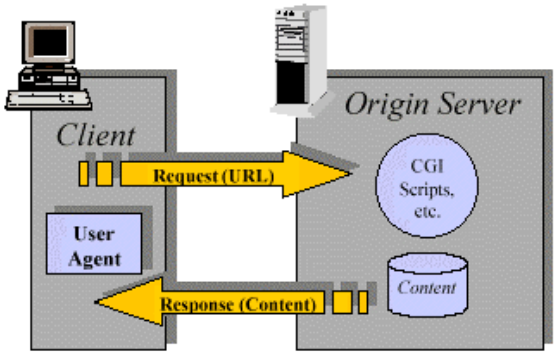


Figura 2 – Modelo de Programação *World Wide Web*

O modelo de programação *WAP* é similar ao modelo *WWW*. Ele está representado na Figura 3 conforme Fórum (1999). Também oferece muitos benefícios para os desenvolvedores de aplicação, pois o modelo de programação já é familiar. A arquitetura também é similar, e a utilização de ferramentas já existentes (por exemplo, *Web Server*, ferramentas *XML*, linguagens de *script*, etc.).

O modelo *WAP* trata os conteúdos *WAP* dentro de uma recomendação semelhante aos já definidos no ambiente WWW. O conteúdo é transportado, usando um conjunto de protocolos de comunicações padrões baseados nos protocolos de comunicação da WWW. Um *microbrowser*, denominação dada aos *browsers* dos dispositivos móveis, é a interface do usuário, da mesma maneira que o *browser* padrão da *WWW* (Fórum,1999).

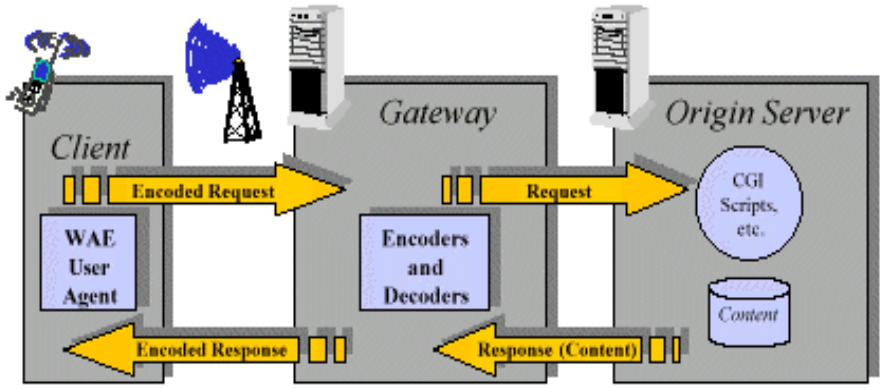


Figura 3 – Modelo de Programação WAP

Esta infra-estrutura permite que os usuários de terminais móveis tenham acesso a uma grande variedade de conteúdo e aplicações *WAP*, e também permite que os desenvolvedores possam construir aplicações e serviços que rodem em um grande tipo de terminais móveis. Como vimos, os dois modelos, *WAP* e *WWW*, são muito semelhantes, mas existem duas diferenças importantes, segundo Mann (2000):

- Sempre vai existir um servidor Gateway entre o agente de usuário (por exemplo, microbrowser) e o servidor de conteúdo na Internet. Este servidor Gateway faz a translação dos protocolos WAP vindos do agente usuário para o HTTP para comunicar com o servidor de conteúdo e vice-versa. Ele também compila dinamicamente os programas WML e WMLscript vindos do servidor de conteúdo, para depois enviá-los para o agente de usuário;
- A comunicação entre o agente de usuário e o servidor Gateway WAP é feita com os protocolos WAP. O mais importante desses protocolos é o Wireless Session Protocol (WSP) que é o binário compactado do HTTP 1.1.

Segundo Arehart (2000), os artigos escritos sobre *WAP* que estão na Internet ou em conferências falam sobre *WAP Proxy*, *Wap Gateway* ou *Wap Server*, e às vezes fazem confusão sobre o que são esses elementos. O termo *WAP Server* ou servidor de aplicação nada mais é do que o servidor de rede onde as informações e aplicações *Web/WAP* estão armazenados.

Na Figura 4, Arehart (2000) apresentou a comparação entre as camadas do protocolo *WAP* e dos protocolos utilizados na *WEB*.

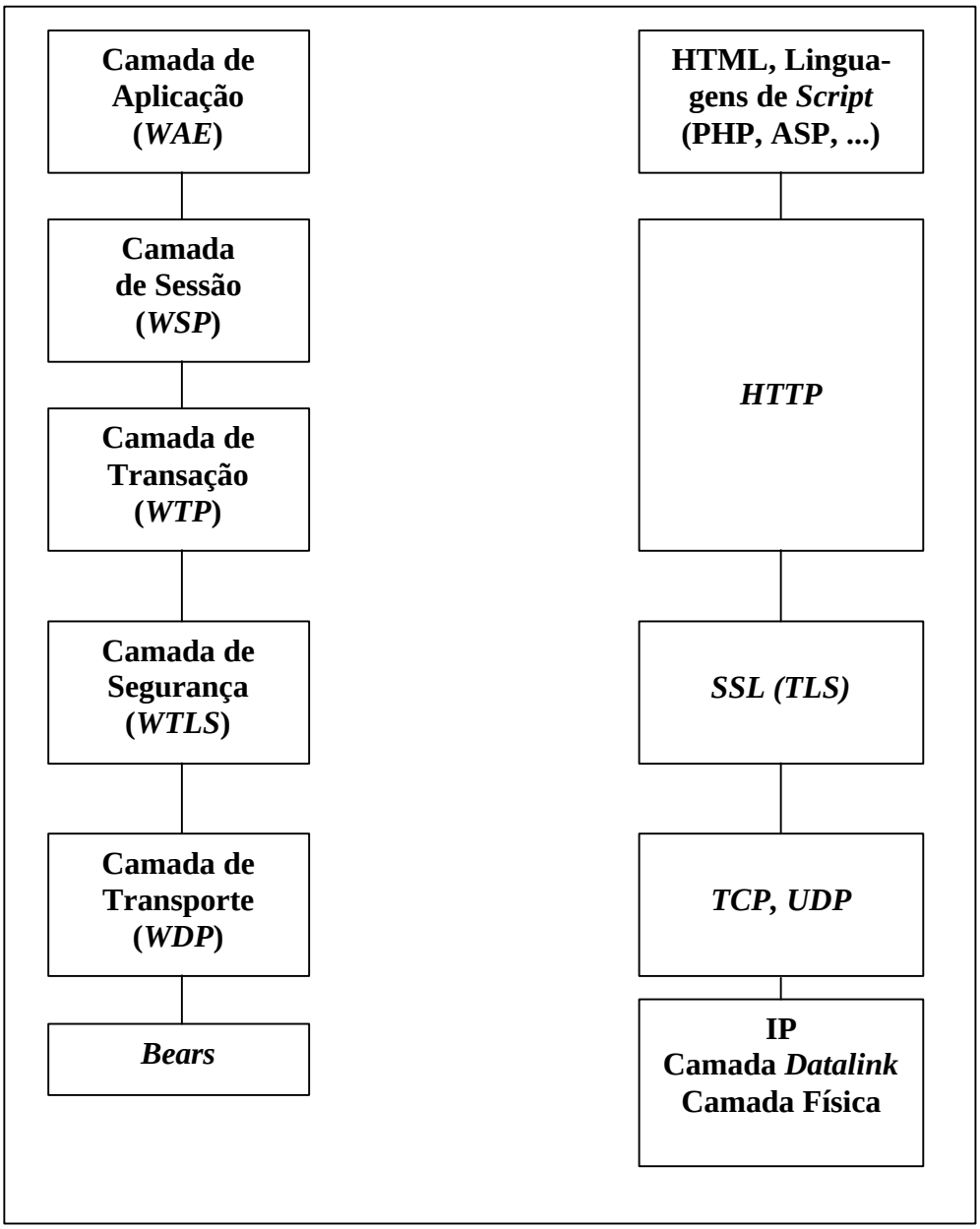


Figura 4 – Comparativo entre o Padrão *WAP x WEB*

2.1 CAMADA DE APLICAÇÃO (*WAE*)

A camada de aplicação, ou *Wireless Application Environment (WAE),* tem como objetivo estabelecer um ambiente que permita aos operadores e os fornecedores de serviços construírem aplicações que possam alcançar uma gama de diferentes plataformas de redes *wireless*, de maneira eficiente e útil. Segundo Fórum (1999), o *WAE* é uma coleção de especificações tecnológicas que são novas ou foram baseadas em tecnologias já existentes. Algumas dessas tecnologias, que foram utilizadas no desenvolvimento da *WAE*, são:

- *Unwired Planet's Hand Helt Markup Language* (HDML);
- W3C *Hypertext Markup Language* (HTML);
- ECMA-262 *Standard* "*ECMAScript Language Specification*";

- IMCs formato de troca de dados de calendário (*vCalendar*) e formato de troca de dados de agenda telefônica (*vCard*);
- Um conjunto de tecnologias WWW como *URLs* e o HTTP;

A arquitetura de *WAE* inclui todos os elementos da arquitetura *WAP* relacionados à especificação de aplicação e execução. Neste momento, a arquitetura *WAE* é focada nos aspectos do lado do cliente da arquitetura do sistema de WAP; isto é, artigos relativos a agentes de usuário.

Na *WWW* existe um modelo lógico muito flexível. As Aplicações apresentam um conteúdo para o cliente em formatos de dados padrão que são navegados por agentes de usuário no lado cliente, conhecido como *Web browsers*. Tipicamente, um agente de usuário envia um pedido, requisitando um ou mais dados, chamado de objetos de dados (ou conteúdo) para um servidor de origem. Um servidor de origem responde com os dados pedidos expressados em um dos formatos padrão conhecido do agente de usuário, por exemplo, *HTML* (Mann, 2000).

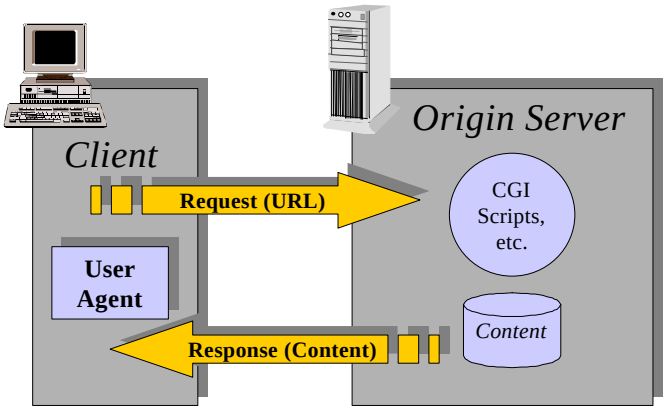


Figura 5 – Modelo WWW

No WAE é adotado um modelo muito semelhante ao modelo da *WWW*, conforme a Figura 6 (Fórum, 1999). Todo o conteúdo é especificado em formatos que são semelhante aos formatos padrão da Internet. O conteúdo é transportado utilizando um protocolo semelhante ao *HTTP* no domínio *wireless*, o *WSP*. A arquitetura do *WAE* permite que todo o conteúdo e serviços, que estejam armazenados em um servidor *Web* de origem, podem ser incorporados usando tecnologias já consagradas, por exemplo, PHP ou JSP. Todo o conteúdo é localizado utilizando o *URL*, padrão *WWW*. O *WAE* assume a existência da funcionalidade do *Gateway* que é responsável por codificar e decodificar os dados transferidos de e para o cliente móvel.

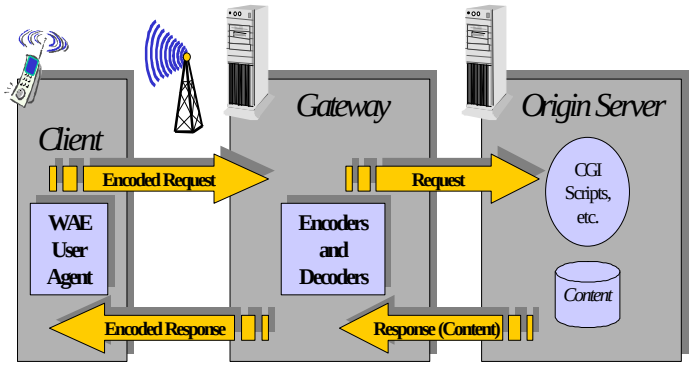


Figura 6 – Modelo do WAE

Normalmente, o agente de usuário no terminal cliente inicia um pedido para um conteúdo. Porém, nem todo o conteúdo enviado ao terminal será o resultado de um pedido feito no lado do terminal. Segundo Fórum (1999), o *WTA* inclui mecanismos que permitem que servidores de origem entreguem conteúdo gerado ao terminal sem o pedido desse terminal, como ilustrado na Figura 7. Em alguns casos, o que o servidor de origem entrega ao dispositivo móvel pode depender das características desse dispositivo móvel (Arehart, 2000).

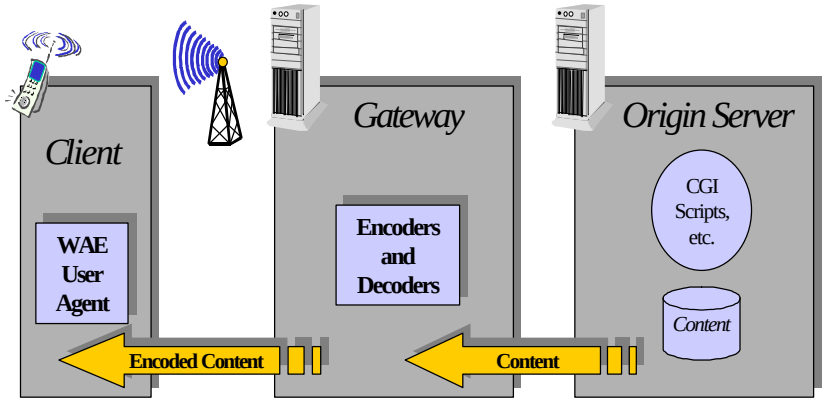


Figura 7 – Modelo WAE Baseado em Tecnologia Push

O mecanismo de nomes *URL* usado no *WAE* foi motivado pelos seguintes cenários, segundo Fórum (1999):

- um serviço seguro (por exemplo, acesso à conta bancária), onde uma conexão segura que usa WTLS, precisa obrigatoriamente de um Gateway seguro controlado pelo provedor de conteúdo;
- um provedor de conteúdo que quer prover um Gateway de *caching*, que vai fazer *cache* do conteúdo codificado para melhorar desempenho.

A maioria das conexões entre o *browser* e o *Gateway* usa *WSP*, indiferentemente do protocolo do servidor de destino. O *URL* é usado para distinguir o conteúdo desejado, sempre especificando o protocolo usado pelo servidor de destino. Além de executar a conversão de protocolo, traduzindo requisições de outros protocolos para o *WSP* e as respostas em *WSP*. O *Gateway* também executa conversão de conteúdo.

O *WAE* está dividido em duas camadas lógicas (Fórum 1999):

- Agentes de usuário, que incluem itens como browsers, agenda telefônica, editores de mensagem, etc.;

- Serviços e Formatos que incluem elementos comuns e formatos acessíveis para agentes de usuário como WML, WMLScript, formatos de imagem, e formatos *vCard* e *vCalendar*, etc.

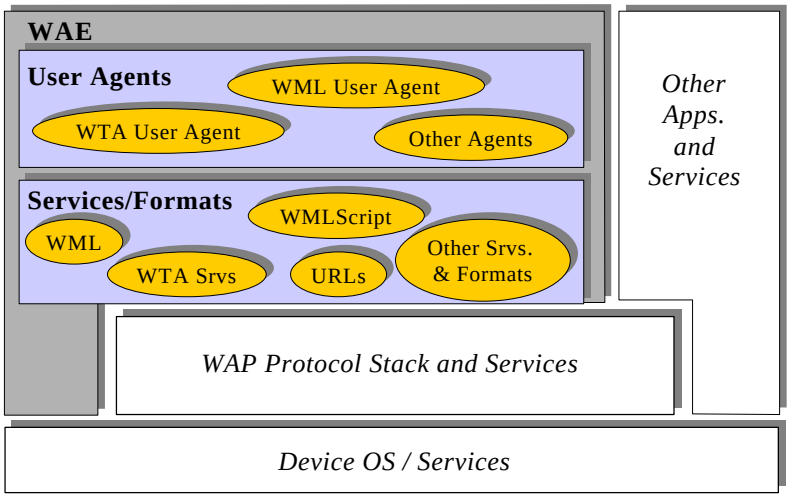


Figura 8 – Componentes do WAE

O agente de usuário *WML* é o agente de usuário fundamental do *WAE*. Porém, o *WAE* não é limitado a um agente de usuário *WML*. O *WAE* permite a integração de agentes de usuário de domínio específico com determinadas arquiteturas. Em particular, um agente de usuário *Wireless Telephony Application (WTA)* foi especificado como uma extensão à especificação do *WAE*, usado em ambientes de telefonia móveis. As extensões do *WTA* permitem aos desenvolvedores ter acesso e interagir com características dos telefones móveis (por exemplo, controle de chamada), como também outras aplicações existentes nos telefones, como agenda telefônica e aplicações de calendário.

2.1.1 Características dos Agentes de Usuário

Para aperfeiçoar o modelo cliente-servidor *WAE*, várias características são enviadas do agente de usuário para o servidor de origem *WAP*. Estas características permitem ao servidor de origem evitar o envio de conteúdo impróprio ao agente de usuário. Eles também provêem o servidor e *Gateway*, meios de personalizar a resposta para um agente de usuário em particular. Os cabeçalhos de conteúdo *WSP/HTTP 1.1* são utilizados para executar a negociação de conteúdo e definir um conjunto de caracteres codificados e as configurações da linguagem. O servidor de origem ou *Gateway* WAP pode necessitar, ou pode querer, modificar as respostas baseado em características do agente de usuário. Para cada tipo de mídia *WAP* incluído no cabeçalho *Accept WSP/HTTP*, o agente de usuário deve incluir um parâmetro, chamado *uaprof*, especificando o *URI* para um perfil que especifica as características de agente de usuário (Fórum, 1999).

Alguns *Gateways* podem ter conteúdo recebido dos servidores de origem em *cache*. Se um agente de usuário solicita o conteúdo que um *Gateway* tem em *cache* e a solicitação contém cabeçalhos característicos, o *Gateway* não deve prover conteúdo de *cache* para o agente de usuário, a não ser que pelo menos uma das três condições seguintes seja verdadeira:

- os cabeçalhos característicos especificados na solicitação são idênticos ao usado quando o Gateway recebeu o conteúdo inicialmente;

- os cabeçalhos característicos especificados na solicitação são idênticos ao usado quando o Gateway recebeu o conteúdo inicialmente, com exceção do parâmetro *uaprof*;
- o Gateway pode garantir por outros mecanismos (por exemplo, análise de metadado HTTP) que uma nova solicitação para o servidor de origem que usa o cabeçalho *Accept* do agente de usuário resultaria no mesmo conteúdo que o Gateway tem no *cache*.

O *WTA* especifica um conjunto básico de aplicações para serviços de telefonia. Esse conjunto básico precisa de um agente de usuário *WTA*. O agente de usuário *WTA* amplia as capacidades do agente de usuário *WML*, adicionado às capacidades para conectar com serviços de rede móveis que interagem com componentes definidos. Esses componentes são:

- um mecanismo de armazenamento consistente, o Repositório, para armazenar o conteúdo que executa serviços WTA no cliente;
- um mecanismo que controla eventos para prover eventos de rede (por exemplo, chamada entrante) dirigindo os serviços;
- uma interface local, com funções relacionadas à telefonia (por exemplo, configuração de chamada) no cliente.

2.1.2 Tipos de Mídia *WAE*

O *WAE* especifica ou adota vários formatos de conteúdo que facilitam a troca de dados. Os formatos mais importantes são os formatos *WML* codificado e os *bytecodes WMLScript*. A codificação *WML* e *WMLScript* faz a transmissão dos dados WML e WMLScript mais eficiente e minimiza os esforços computacionais que precisam executar no cliente. O Agente de usuário *WAE* “empurra” o dado baseado em seu tipo de conteúdo. Cada tipo de conteúdo especifica a estrutura dos dados e sua semântica, e o agente de usuário é responsável por interpretar o conteúdo de acordo com as regras especificadas para cada.

Outros formatos de conteúdo incluem o formato de troca de imagem e formatos específicos para aplicação. Em geral, o método de troca de dados depende do tipo dos dados e do agente de usuário envolvido. Os tipos de mídias são: Formato codificado *WML*; Formato codificado *WMLScript*; Formato de Cartão Eletrônico de Negócios (*vCard 2.1*); O Calendário Eletrônico e Formato de Troca Programado (*vCalendar 1.0*); Imagens; Mensagens de Múltiplas Partes.

O formato *WBMP* habilita informação gráfica a ser enviada a um grande número de dispositivos móveis. O formato *WBMP* é independente do terminal e descreve somente a informação gráfica. O formato WBMP é configurado de acordo com o valor *TypeField* com mapas para toda a informação codificada da imagem , como:

- Organização do *pixel* e codificação;
- Organização da Paleta e codificação;
- Característica de Compressão.

O *WAE* trabalha com os objetos agenda telefônica e calendário do dispositivo móvel. O *WAE* adotou os formatos de dados *vCard* e *vCalendar*. Estes formatos de dados são meios padrões da indústria para trocar lista telefônica, cartão de visita eletrônico e informação de calendário, e estão em uso em uma grande variedade de dispositivos e programas. Nesta especificação, esses dados da lista telefônica, cartão de visita eletrônico e informação de calendário estão codificados no formato *vCard*.

2.1.3 Wireless Telephony Application (WTA)

O *WTA* é uma coleção de extensões específicas de telefonia para chamada e mecanismos de controle das características que fazem os Serviços de Rede Móveis avançados disponíveis para os desenvolvedores e usuários finais. O *WTA* mistura as características e serviços de redes de dados com os serviços de redes de voz. Ela introduz mecanismos que asseguram acesso seguro a recursos importantes dentro dos dispositivos móveis. O padrão *WTA* permite o processamento de eventos em tempo real para o usuário final enquanto ele navega. Nesse padrão, o cliente e o servidor coordenam o conjunto de regras que controlam um evento por uma tabela de eventos. Os servidores de origem *WTA* podem ajustar as regras do cliente atualizando as tabelas de evento de um cliente se for preciso, como definido em .

Os principais objetivos do *Wireless Telephony Application* estão listados abaixo (Fórum, 1999):

- Habilitar os Operadores de Rede para prover serviços de telefonia avançados que são bem integrados e têm interfaces de usuário consistentes.
- Habilitar os Operadores de Rede para criar conteúdo feito sob medida para aumentar demandas e acessibilidade por vários serviços nas suas Redes.
- Habilitar os Operadores de Rede para alcançar um número maior de dispositivos através de características genéricas do WAE que permitam para o operador criar conteúdo independente de características específicas dos dispositivos e ambientes.
- Habilitar os desenvolvedores de três camadas para criar conteúdo independente de rede que tenham acesso às características básicas.

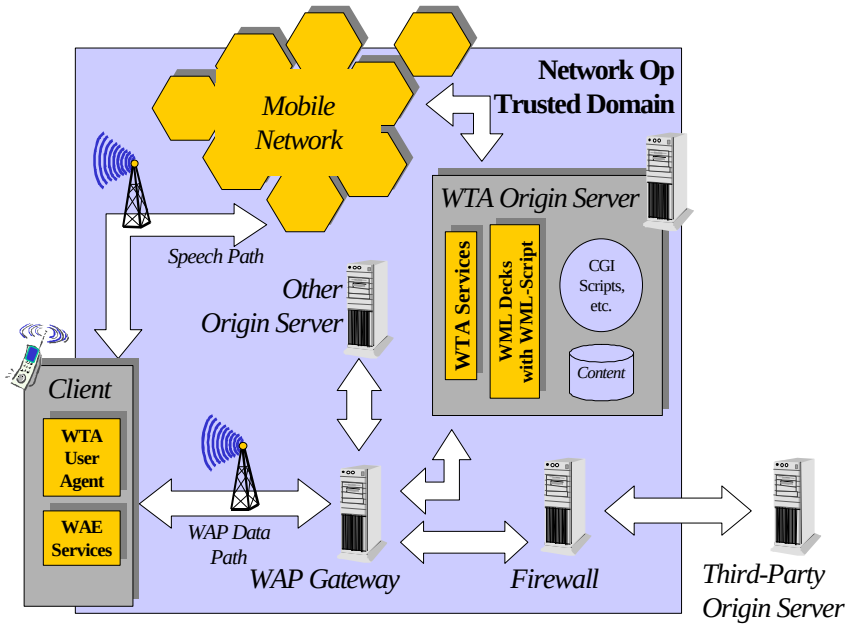


Figura 9 – Arquitetura Lógica *WTA*

Os elementos da rede lógica *WTA*, apresentados na Figura 9, são:

- Conteúdo e Geradores de Conteúdo;
- *Firewalls* (opcional).

O agente de usuário *WAE* é conectado à rede móvel usando conexões de sinais dedicados. O Servidor *WTA* (um servidor de origem) se comunica com o cliente usando a

pilha de protocolo *WAP*. O servidor *WTA* pode ser conectado à rede móvel e pode ser responsável para fornecer conteúdo a seus clientes.

2.2 CAMADA DE SESSÃO (WSP)

A camada de Sessão da família de protocolos do *WAP* é chamada de *Wireless Session Protocol (WSP)*. O *WSP* provê à camada de aplicação de nível superior do *WAP* uma interface consistente para dois serviços de sessão. O primeiro é um serviço orientado à conexão que opera sobre um protocolo de camada de transação *WTP*. O segundo é um serviço de *"connectionless"*, que opera sobre um serviço de transporte de *datagrama* seguro ou não-seguro, provendo as funcionalidades do *HTTP1.1* e incorpora características novas como sessões duradouras, uma facilidade comum para dados *"push"*, capacidade de negociação e *suspend/resume* de sessão. Os protocolos da família *WSP* são otimizados para redes com banda pequena, e com portadora com latência relativamente longa .

2.2.1 Arquitetura WSP

Um modelo de camadas do protocolo *WAP* é ilustrado na Figura 10, conforme Fórum (1999). O modelo de camadas dos protocolos *WAP* e as suas funções são semelhantes ao Modelo de Referência *ISO OSI* para camadas superiores. Entidades de Administração de camada controlam a inicialização, configuração, e condição de erro (como perda de conectividade devido ao terminal móvel estiver fora da área de cobertura).

O *Wireless Session Protocol* é um protocolo da família do nível de sessão para operações remotas entre um cliente e um *Proxy* ou servidor. O *WSP* foi projetado para funcionar na transação e em serviços de *datagrama*. A camada de segurança é para ser uma camada opcional sobre a camada de transporte. A camada de segurança preserva as interfaces de serviço de transporte. São assumidas a transação e a sessão ou entidades de administração de aplicação, para prover o suporte adicional que é exigido para estabelecer o contexto de segurança e conexões seguras (Fórum, 1999).

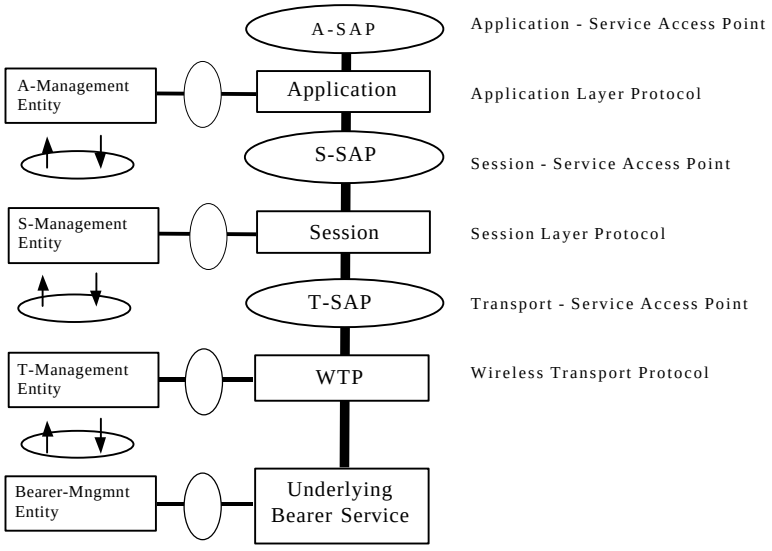


Figura 10 – Modelo de Camadas do Protocolo WAP

Segundo Fórum (1999) , o *WSP* provê meios para organizar a troca de conteúdo entre as aplicações cliente/servidor. Provê principalmente o meio para estabelecer uma sessão segura entre o cliente e o servidor e liberar essa sessão de uma forma ordenada; troca de conteúdo entre o cliente e o servidor usando codificação compactada; e a capacidade para suspender e retomar uma sessão. Os serviços e protocolos (*WSP/B*) são utilizados para as aplicações do tipo de navegação. O *WSP/B* define dois tipos de protocolos: um que provê sessão num modo de conexão de serviço em cima de um serviço de transação, e outro que provê serviços *"connectionless"* em cima de serviços de transporte de *datagrama*.

O *WSP* foi desenhado a partir do binário do *HTTP1.1*, em um formato compactado, conseqüentemente todas as requisições enviadas para um servidor e as respostas enviadas para o cliente podem incluir ambos os cabeçalhos e dados, e, sendo assim, todos os métodos definidos no *HTTP1.1* são suportados no *WSP/B*. O *WSP* é responsável por fazer a transferência dos dados para a camada de Aplicação. Os cabeçalhos de conteúdo do *HTTP1.1* são utilizados para definir o tipo do conteúdo, o conjunto de caracteres codificados, as linguagens, etc. Porém, codificações binárias compactadas dos cabeçalhos utilizados são definidas para reduzir o *overhead* do protocolo.

O WSP/B também implementa algumas capacidades adicionais ao *HTTP1.1*, como a capacidade para negociação entre dois pontos. O *WSP/B* também provê dois tipos de transferência dos dados, o *push* e o *pull*. O *pull* é feito quando se utiliza o mecanismo *request/response do HTTP1.1*. Ele provê três tipos de transferências com o mecanismo *Push*: os dados confirmados são empurrados (*push*) no contexto da sessão existente; os dados não confirmados são empurrados (*push*) no contexto da sessão existente; os dados não confirmados são empurrados (*push*) sem uma sessão existente. O mecanismo de transferência de dados confirmados com *push* permite ao servidor fazer o *push* de dados a qualquer hora para o cliente durante uma sessão. O servidor recebe confirmação que os dados *push* foram entregues. Os dados *push* não confirmados dentro de uma sessão existente provêem uma função semelhante com os dados *push* seguros, mas sem confirmação de entrega.

2.3 CAMADA DE TRANSAÇÃO (*WTP*)

Um protocolo de transação é definido para prover os serviços necessários para aplicações com navegação "interativa" (*request/response*). Durante uma sessão de navegação, o cliente pede informação de um servidor que pode ser fixo ou móvel, e o servidor responde com a informação. A dupla *request/response* é referenciada como uma "transação" nesse trabalho. O objetivo do protocolo *WTP* é entregar a transação com confiança. O *WTP* roda em cima de um serviço de *datagrama* e opcionalmente um serviço de segurança. O *WTP* foi definido como um protocolo orientado para uma transação leve, para implementação em clientes magros (estações móveis) e opera muito bem em cima de redes *wireless* de *datagrama*. Os benefícios de utilizar o *WTP* incluem (Fórum, 1999):

- Implementa confiabilidade nos serviços em cima de datagrama. O WTP alivia a camada superior de re-transmissões e reconhecimentos que são necessários se os serviços de datagrama forem utilizados.
- Implementa eficiência em cima de serviços orientados à conexão.
- O WTP é orientado à mensagem e foi projetado para serviços orientados a transações, como navegação (browsing).

A lista seguinte resume as características de *WTP*.

- Três classes de serviço de transação: Classe 0: Invocação incerta de mensagem, sem mensagem de resultado; Classe 1: Invocação segura de mensagem, sem mensagem de resultado; Classe 2: Invocação segura de mensagem, com uma mensagem de resultado segura.
- A Confiança é conseguida pelo uso de identificador de transação único, reconhecimento, remoção de duplicadas e re-transmissões.
- Orientado à Mensagem. A unidade básica de intercâmbio é uma mensagem inteira e não um fluxo de bytes (*stream*).
- Cancelamento de transação. O cancelamento pode ser ativado pelo usuário que cancela um serviço pedido.
- O protocolo permite transações assíncronas. O Responder (provedor de WTP que responde a uma transação é chamado de Responder) retorna o resultado quando os dados ficam disponíveis.

O *WTP* foi especificado para rodar em cima de um serviço de transporte de *datagrama*. O *WTP* não implementa mecanismos de segurança, para garantir segurança a utilização do *WTLS* é necessária. A unidade de dado do protocolo *WTP* fica situada em uma parte dos dados do *datagrama*. Considerando que os *datagramas* são incertos, o *WTP* é exigido para executar re-transmissões e enviar o reconhecimento, para prover um serviço seguro ao usuário de *WTP*. O *WTP* também é responsável pela concatenação (se possível) de múltiplas unidades de dados do protocolo, em um serviço de transporte de unidade de dados. O *Datagrama* do protocolo *WAP* é o *WDP*.

2.4 CAMADA DE SEGURANÇA (*WTLS*)

O protocolo da camada de Segurança na arquitetura de WAP é chamado a *Wireless Transport Layer Security (WTLS).* A camada *WTLS* pode operar em cima da camada de protocolo de transporte (WDP). A camada *WTLS* é modular e depende do nível de segurança exigido de determinada aplicação, se será utilizada ou não. O *WTLS* proporciona para a camada de nível superior do *WAP* uma interface de serviço de transporte seguro que preserva a interface de serviço de transporte. Além disso, o *WTLS* provê uma interface para administrar (por exemplo, criando e terminando) conexões seguras. O *WTLS* foi baseado no *SSL v3.0*, que é utilizado com os protocolos *Web* para garantir a segurança na troca de informações pela Internet.

2.5 CAMADA DE TRANSPORTE (*WDP*)

O protocolo da camada de Transporte na arquitetura do protocolo *WAP* é o *Wireless Datagram Protocol* (WDP). A camada *WDP* opera sobre os dados que são suportados pelos serviços de portadora de vários tipos de rede. Como um serviço geral de *datagrama*, o *WDP* oferece um serviço consistente ao protocolo de camada superior (Segurança, Transação e Sessão) do *WAP* e se comunica transparentemente com mais de um dos serviços de portadora disponíveis. Os protocolos da família *WAP* são projetados para uso em cima de portadores com pouca banda em redes de telecomunicações *wireless*. O *WDP* suporta vários exemplos de comunicação simultâneos de uma camada mais alta em cima de um único portador de serviço *WDP*. O número de porta identifica a entidade de camada mais alta sobre *WDP*. Esta pode ser outra camada de protocolo como o *Wireless Transaction Protocol (WTP)* ou o *Wireless Session Protocol (WSP)* ou uma aplicação como correio eletrônico.

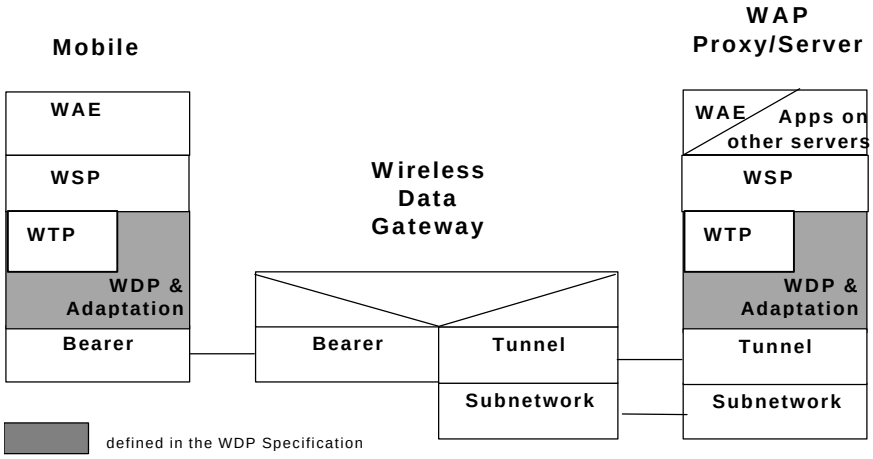


Figura 11 – Arquitetura *WDP*

Na Figura 11 (Fórum, 1999), as áreas sombreadas são as camadas de protocolo que a Especificação do *WDP* é aplicável. No *Mobile* (móvel), o protocolo *WDP* consiste nos elementos *WDP* comuns mostrados pela camada chamada *WDP*. A Camada de Adaptação é a camada do protocolo *WDP* que mapeia as funcionalidades do protocolo *WDP* diretamente sobre uma portadora específica. A Camada de Adaptação é diferente para cada portadora e as transações com as capacidades especificam as características daquele serviço de portadora. A camada da portadora é o serviço de portador como *GSM SMS*, ou *USSD*, ou pacote dados *CDMA*. No *Gateway* termina a camada de Adaptação e passa os pacotes *WDP* para um *WAP Proxy/Server*, por um protocolo de *Tunnelling* (Túnel) que é a interface entre o *Gateway* que suporta o serviço de portadora e o *WAP Proxy/Server*. A *SubNetwork* é qualquer tecnologia de rede comum que pode ser usada para conectar dois dispositivos de comunicação, exemplos são redes de grandes áreas de cobertura, baseadas em *TCP/IP* ou *X.25*, ou *LANs* com *TCP/IP* em cima de *Ethernet*. O *WAP Proxy/Server* pode oferecer conteúdo de aplicação ou pode agir como um *Gateway* entre as suítes de protocolos *wireless WTP* e a Internet padrão.

2.6 CONCLUSÃO SOBRE O PROTOCOLO *WAP*

O protocolo *WAP*, foi criado baseando-se no modelo *OSI*, e mesmo sendo baseado nesse padrão, tem muito para melhorar, principalmente em relação à qualidade de desempenho, pois 14.4 *Kbps* é uma banda muito limitada para suportar aplicações pesadas de *M-commerce*, informações on-line, localização, etc.

Outro problema que deve ser resolvido é em relação à segurança, principalmente no que se refere à utilização do *Gateway WAP*, que possibilita que pessoas não autorizadas tenham acesso as informações enviadas. O suporte para imagens ainda é muito pobre no *WAP* (versão 1.2), pois os gráficos preto e branco não possibilitam a criação de sistemas atraentes visualmente para os usuários, restringe a utilização de *marketing* via celular e tira a possibilidade de criar jogos para celulares, um segmento muito grande que não pode ser atendido com qualidade. Além disso, o suporte de arquivos com no máximo de 1,5 *Kb* torna complexo o desenvolvimento de aplicativos, tornando o desenvolvimento mais demorado e com um custo maior do que na Web.

Mesmo com todos os problemas, o WAP está trazendo novos serviços, e possibilitando que pessoas que não tenham acesso à computação, utilizem serviços como

home banking, compras on-line, consultar dados do Detran, consultar entregas feitas pelo Correio do Brasil, etc.

Certamente com o avanço da tecnologia *wireless*, o aumento de banda para 115 *Kbps* que já está disponível em vários países da Europa, e deve estar disponível no Brasil em 2002, possibilitará que o WAP disponibilize mais recursos para os desenvolvedores, como conteúdo multimídia, utilizando imagens, sons e vídeo. A utilização das redes por pacote, como novas regras de tarifas, facilitará o acesso à Internet pelos telefones celulares e por um custo muito mais baixo do que o atual. Também a integração com recursos de telefonia com voz e localização do dispositivo móvel futuramente serão disponibilizados aos desenvolvedores, possibilitando a criação de novas aplicações e fazendo com que a necessidade de utilização da Internet Móvel pelos usuários torna-se uma necessidade, como ocorre com a Web.

3 XML, WML E WMLSCRIPT

Este capítulo apresenta uma introdução às linguagens XML, WML e WMLScript.

3.1 XML

A linguagem *XML* é uma tecnologia para criação de documentos estruturados que podem ser trocados entre sistemas. No *XML*, diferente do *HTML*, o modo que dados são exibidos não é o descrito, mas a estrutura e a organização dos dados que são definidos. Como o resultado das diferenças entre os protocolos de sistema para sistema e como são implementados o armazenamento dos dados e a transmissão dos dados, a transferência dos dados entre diferentes plataformas ou aplicações sempre foram problemáticos. Por exemplo, a troca eletrônica de documentos entre dois negócios pode ficar problemática e, freqüentemente, isso significa o uso de arquivos de texto simples, como no caso dos sistemas de *EDI*. Diferenças aproximadas na segurança, nos sistemas operacionais e nas técnicas de armazenamento de dados causam problemas como a integridade dos dados que ficam complexas e caras para serem resolvidas (Arehart, 2000).

O *XML* foi um produto que o *W3C* se esforçou muito, criando uma linguagem que descreva documentos em um sistema de modo independente. Ela é um subconjunto do *SGML (Standart Generalized Markup Language)* – um mecanismo de armazenamento de documentos de propósito geral que foi definido para habilitar o armazenamento de um grande número de documentos. Em adicional ao conteúdo do documento, o *XML* contém o metadado (*metadata*) – dados que descrevem o conteúdo. O *XML* é uma linguagem de marcação (*markup*) que inclui regras relativas a como as *tags* podem ser utilizadas, mas as *tags* também podem ser definidas por uma determinada aplicação. Na Figura 12 (Arehart, 2000), um exemplo de uma receita em um documento *XML*.

```
<?xml version = “1.0”?>
<!--XML Exemplo -->
<recipe name = “cornflakes”>
      <ingredients>
            <ingredient>
                  <name>
                        Cornflakes
                  </name>
                  <quantity>
                        150g
                  </quantity>
            </ingredient>
            <ingredient>
                  <name>
                        Milk
                  <name/>
                  <quantity>
                        ¼ pint
                  </quantity>
            </ingredient>
      </ingredients>
<method>
            Place cornflakes in a bowl.
            Pour milk into cornflakes.
</method>
</recipe>
```

Figura 12 – Exemplo de Código XML

Uma *tag* é parte do documento delimitada pelos símbolos de maior e menor (< >). Um elemento é chamado de uma seção do documento e começa com a *tag* no formato <*tag*> e termina com *tag* no formato</*tag*>. Onde uma tag não delimita o conteúdo, ela tem a forma <tag/>. Neste exemplo, temos vários elementos, um elemento “*ingredient”*, um elemento “*recipe”*, um elemento “*method”*, e assim por diante. Sendo que a *tag* <*recipe*> contém um atributo que é o “*name*”. Um atributo é uma associação colocada dentro de uma *tag* para dar significado adicional àquele elemento (Arehart, 2000). O *XML* possui um componente importante que é o *“parser” XML* (analisador da gramática), utilizado para processar os documentos *XML*. Um documento precisa satisfazer certos critérios na ordem para ser analisado gramaticalmente com sucesso, e se ele satisfizer esses critérios, é dito que ele é bem formado. O *XML* é *“case sensitive”*, ou seja, as *tags* têm que ser definidas utilizando o mesmo tipo, pois para *o XML* <*hello*>, <*Hello*> e <*HELLO*> são *tags* diferentes.

Mas como saber quais são as regras que precisam se seguidas para o documento *XML* ficar bem formado? No exemplo acima, é óbvio que o elemento “*method”* não pode ser colocado dentro do elemento *“quantity”*, o que define isso é o *Document Type Definitions (DTD)*. Um documento *XM*L pode estar associado a um *DTD* que mais adiante vai indicar a sua estrutura; o *DTD* descreve as *tags* que devem estar dentro de um documento em conformidade com o *DTD*, e que *tags* podem estar aninhadas com outras *tags* e outras informações. O *parser XML* pode então utilizar as informações vindas do *DTD* para checar se a estrutura do documento está correta. Se o documento *XML* está com a estrutura correta em relação ao *DTD* associado ele é dito válido.

Os pontos fortes da linguagem *XML* são os seguintes:

- inteligência, o XML é inteligente para qualquer nível de complexidade. A marcação pode ser alterada de uma marcação mais geral como "<CÃO> Lassie

</CÃO>" para uma mais detalhista, como "<CÃO> <VENHA_PARA_CASA> <COLLIE> Lassie </COLLIE> </VENHA_PARA_CASA> </CÃO>".

- a informação conhece a si mesma. Não é necessária mais nenhuma idéia indesejável; adaptação, pois o XML é a língua-mãe de outras linguagens. Assim, linguagens como a WML, JaneML tornaram-se possíveis.
- a adaptação é infinita, podendo ser criadas novas marcações se necessário;
- a manutenção, pois o XML é fácil de manter, sendo que ele só contém idéias e marcações, e a folha de estilo e o link são enviados em separado e não dentro do documento; ligação, pois o XML possui um mecanismo de ligação bem avançado, porque além de ligar um objeto a outro, como no WML ou HTML, o XML pode ligar dois ou mais pontos a uma idéia;
- a simplicidade, um usuário de média experiência que olha o XML pode achá-lo difícil de acreditar no que vê. Comparada com a HTML, não. Comparada com a SGML, é um estudo de simplicidade. A especificação da SGML possui 300 páginas. A da linguagem XML, 33 páginas;
- a portabilidade, a razão da sua existência é a força e a portabilidade. A linguagem XML pode ser navegada com ou sem o seu DTD (*Document Type Definition*, ou Definição de Tipo de Documento – as normas que definem como as tags são estruturas nos documentos XML), tornando o *download* mais rápido. Tudo que um browser precisa para entender o XML é ter a noção que ela própria e a folha de estilos controlam sua aparência. Se uma validação estrita é necessária, o seu DTD pode acompanhá-lo e fornecer detalhes exatos da sua marcação (Arehart, 2000).

3.2 WML

A especificação do protocolo *WAP* define a *Wireless Markup Language (WML).* A *WML* é uma linguagem de construção baseada na *XML,* e é utilizada para especificar o conteúdo e a interface de usuário para equipamentos que têm pouca banda de transmissão de dados e limitações de processamento e de memória, como os telefones celulares e *PDAs.*

A *WML* inclui quatro áreas de funcionalidades:

- Apresentação do texto e layout: a WML inclui suporte para texto e imagem, incluindo uma variedade de comandos para formatação e definição de layout.
- Organização em *Deck*/*Card*: todas as informações em WML são organizadas dentro de coleções de *cards* (cartas) e *decks* (baralhos). As cards especificam uma ou mais unidades de interação com o usuário (por exemplo, um menu, ou uma tela de texto). Logicamente, um usuário navega entre uma série de *cards* WML, visualizando o conteúdo de cada uma, entrando com as informações requisitadas, fazendo escolhas, e movendo para outra card. As *cards* são agrupadas dentro de decks. Um *deck* WML é similar a uma página HTML que é identificada por uma URL e é uma unidade de transmissão de conteúdo.
- Navegação entre *cards* e *links*: a WML inclui suporte para a navegação explicita entre *cards* e *decks*. A linguagem WML também inclui provisão para o tratamento de eventos que ocorrem no equipamento, com propósito para utilizar na navegação ou mesmo para executar algum script. A WML também suporta *links* âncoras similares aos que existem no HTML.
- Parametrização de String e gerenciamento de estados: Todos os decks WML podem ser parametrizados usando um modelo de estado. Variáveis podem ser

usadas com "*strings*" e serem substituídas em tempo de execução. Essa parametrização faz com que fique mais eficiente a utilização dos recursos da rede.

A *WML* assume a mesma referência da arquitetura *HTML* e da *World Wide Web*. O conteúdo é nomeado usando *URLs* e é completo com outros protocolos padrão que tem a semântica do *HTTP*, como o *WSP*.

Na *WML*, as *URLs* são utilizadas em duas situações: para fazer a navegação (*hyperlink*) ou para utilizar um recurso externo (uma imagem ou um *script*).

3.2.1 Sintaxe *WML*

Para entender a estruturação da linguagem *WML*, deve-se entender primeiro a sintaxe do *WML*. A sintaxe do *WML* foi herdada do *XML* (Mann, 2000).

As principais definições da sintaxe *WML* estão listadas abaixo:

- Entidade: O texto WML pode conter entidade de caracteres numéricos ou texto. Estas entidades especificam um caracter específico no conjunto de caracteres do documento. As entidades são usadas para especificar caracteres no conjunto de caracteres do documento que deve ser ignorado em WML, ou que pode ser difícil de entrar em um editor de texto. Por exemplo, o "ampersand" (&) é representado pela entidade nomeada &. Todas as entidades começam com um ampersand e terminam com um ponto-e-vírgula;
- Elementos: Elementos especificam todos as marcações (markup) e informações estruturais sobre um *deck* WML. Os elementos podem conter uma *tag* no início, o conteúdo e uma *tag* no final. Os elementos têm uma das duas estruturas: <*tag*> conteúdo </*tag*>; ou <*tag*/>. Elementos com conteúdo são identificados com uma *tag* inicial <*tag*> e terminam com uma *tag* no final </*tag*>. Os elementos vazios são representados pela tag <*tag*/>, significando que o elemento não tem conteúdo;
- Atributos: os atributos especificam informações adicionais que serão utilizados pelo elemento. <tag atrib = 'lessa'/>;
- Comentário: os comentários em WML são definidos com a sintaxe: <!—exemplo de comentário -->;
- Variáveis: os *cards* e *decks* WML podem ser parametrizados, utilizando variáveis como visto anteriormente. As sintaxes possíveis: $identificador; $(identificador); $(identificador:conversão). Parênteses são necessários quando o espaço em branco não indica o final da variável;
- Case Sensitive: a WML é uma linguagem case sensitive, ou seja, diferencia tags e atributos escritos de forma diferenciada. <*Hello*> é diferente de <HELLO>;
- Seção CDATA: as seções CDATA são usadas para sair dos blocos de texto e são válidas em qualquer PCDATA; por exemplo, dentro de um elemento. <![CDATA [isso é <b> um teste]]>;
- Erros: a especificação XML define o conceito de documento XML bem formado. Os decks WML que violarem a definição do documento bem formado vai gerar um erro.

3.2.2 Tipos de Dados *WML*

Os tipos de dados suportados no *WML* estão listados abaixo (Mann, 2000):

- Caracter (CDATA) – texto que pode conter entidades numéricas ou texto. O CDATA só é usado em valores de atributos; PCDATA – texto que pode conter numérico ou pode nomear entidades de caráter. Este texto pode conter tags (PCDATA é o "CDATA analisado gramaticalmente"). O PCDATA só é usado em elementos; NMTOKEN – um símbolo de nome, contendo qualquer mistura de nomes de caracteres, como definido pela especificação do XML.
- *Length*: O tipo *length* pode ser especificado como um inteiro que representa o número de *pixel* do "canvas" (tela, papel) ou como uma porcentagem do espaço horizontal ou vertical disponível.
- *Vdata*: O tipo *vdata* representa uma "*string*" que pode conter variáveis referenciadas. Este tipo só é usado em valores de atributo.
- *Flow*: o tipo *flow* representa a informação no nível do *card*. Em geral, *flow* é usado com qualquer marcação que seja incluído.
- *HREF*: O tipo *HREF* referência a um *Uniform Resource Locator* relativa ou absoluta.
- *Boolean*: O tipo *Boolean* referência ao valor lógico que pode ser *TRUE* ou *FALSE*.
- *Number*: O tipo *Number* referencia um valor inteiro maior ou igual a zero.
- xml:*lang*: O atributo xml:*lang* especifica a linguagem natural ou formal de um elemento ou de seus atributos. O valor dos atributos é um código da linguagem de acordo com XML. O atributo identifica ao agente de usuário a linguagem usada no texto que pode ser apresentado ao usuário (por exemplo, o conteúdo de um elemento e valores de atributo).
- Atributos ID e *Class*: Todos os elementos WML têm dois atributos essenciais: o id e *class* que podem ser usados para tarefas como transformações no lado servidor. O atributo id provê em um elemento um único nome dentro de um *deck*. O atributo *class* se afilia em um elemento com uma ou mais classes. Múltiplos elementos podem ter uma *class* com o mesmo nome. Todos os elementos de um único *deck* com um nome de classe em comum são considerados parte da mesma *class*.
- *ContentType*: O tipo *ContentType* representa o tipo de mídia definido na RFC2045.

3.2.3 Estruturas *WML*

As seguintes estruturas fazem parte da WML:

- *Cards* (Cartas) e *Decks* (Baralhos): As informações da *WML* são organizadas em uma coleção de *cards* e *decks* , conforme já visto. O *card* especifica uma ou mais unidade da interação com o usuário, por exemplo, um menu de opções, uma janela de informações ou uma entrada de dados. Logicamente, um usuário navega através de uma série de *cards WML*, faz escolhas e move de um *card* para outro. Os *cards* são agrupados em *decks*. A menor unidade da *WML* que o servidor pode enviar para um *microbrowser* é um *deck.*
- *Templates*: O objeto *template* define um modelo que pode ser aplicado em todos os *cards* do mesmo *deck;* o uso desta facilidade justifica-se pela inserção de botões de navegação. Pode-se cancelar as características do *template* em um determinado *card* (Dias, 2000).

- *URLs*: O padrão de endereçamento da *WML* é o mesmo utilizado pela *HTML*, o protocolo *HTTP* traduz os endereços *WML* em requisições ao cliente, como se fossem requisições *HTML*. Existe um consenso que identificaria melhor os endereços de conteúdo *WML* e *HTML*, por exemplo, nomear os arquivos *HTML* com a extensão *html* e os arquivos *WML* com a extensão *wml* (Dias, 2000).
- *Links* Âncoras: As âncoras são ligações dinâmicas para arquivos ou imagens no servidor ou máquina local. Segundo Dias (2000), o formato é semelhante ao modelo WWW, por exemplo:

```
<img src= “lessa.wbmp”/>
<go href= “http://waptotal.com/”/>
<go href= “/teste/teste.wml”/>
```

3.3 WMLSCRIPT

Em muitos casos, é necessário realizar um processamento na aplicação. Para fazer isso, utiliza-se a linguagem de script WMLScript que trabalha em conjunto com a linguagem *WML*.

A linguagem *WMLScript* é baseada na linguagem *ECMAScript* que é uma linguagem de *script* padrão para muitos *scripts*, tanto no lado do servidor como no lado do cliente. Por causa da pouca banda de dados atualmente suportada pelo *WAP*, foi incluída uma especificação binária para otimização de dados pelo cliente. Também a linguagem *JavaScript* foi baseada no *ECMAScript*, fazendo com que *WMLScript* e o *JavaScript* sejam bem semelhantes, o que torna o aprendizado da *WMLScript* por um programador que conheça *JavaScript* bem fácil (Fórum, 1999).

A linguagem *WMLScript* foi implementada para ser utilizada somente no lado cliente, diferentemente do *JavaScript* e do *VBScript* que podem ser usadas no lado cliente e também no lado servidor. A utilização do processamento no lado cliente é por causa do problema de pouca banda de dados entre o cliente e o servidor. Se um processamento for feito no servidor e demorar muito para retornar o resultado, a aplicação irá abortar. Se o processamento é feito no lado cliente, então o problema de demora no resultado não existe. Assim como *JavaScript* e outras linguagens de *script*, o *WMLScript* roda em um *browser*. Diferentemente da *JavaScript*, *WMLScript* não fica dentro da linguagem de *markup* (*WML*). O código *WMLScript* fica em um arquivo próprio que é chamado de dentro de um *card WML*. A extensão do arquivo *WMLScript* é *.wmls* e a extensão do arquivo *WML* é *.wml*. A chamada do programa *WMLScript* é feita através de um *hyperlink* (Dias, 2000). O exemplo na Figura 13 mostra um *card WML* que chama uma função *WMLScript* (Dias, 2000).

O importante é observar como funciona a chamada da função calcula() que está no arquivo calcula.*wmls*. Na Figura 14 (Dias, 2000), é listado o código do programa calcula() escrito em WMLScript.

```
<?xml version= “1.0”?>
<!DOCTYPE wml PUBLIC “-//WAPFÓRUM//DTD WML 1.1//EN
http://www.wapFórum.org/DTD/wml_1.1.xml>
<--scriptchamador.wml-->
<wml>
      <card id= “main” title = “WMLScript”
      <p>
            O resultado do calculo “2+3” é: $(number)<br/>
            <a href= “chamada.wmls#calcula(2+3)”>
                  Calcular
            </a>
      </p>
      </card>
</wml>
```

Figura 13 – Exemplo de Card WML que Chama Função WMLScript

```
Extern calcula (a,b){
      var x; x = a + b;
      WMLBrowser.setVar(“number”,n);
      WMLBrowser.refresh(); }
```

Figura 14 – Código do Arquivo calcula.wmls

Analisando a linguagem *WMLScript*, foi identificado que programadores já familiarizados com a linguagem *JavaScript* praticamente não precisarão de nenhum treinamento especial para desenvolver programas em *WMLScript*, pois as estruturas das duas linguagens são semelhantes. É claro que o programador vai ter que ficar atento, principalmente em relação às limitações dos equipamentos móveis que rodam os programas *WMLScript.* Isso faz com que o programador tenha que modificar um pouco o seu modo de resolver alguns problemas, pois ele sempre tem que observar as limitações dos dispositivos clientes, e também tem que escrever uma aplicação que seja compatível com a maioria dos dispositivos móveis atuais que são de diferentes fabricantes, com telas de tamanho diferentes, e poder de processamento muito reduzido.

3.4 CONCLUSÃO

A linguagem WML, por ser um documento XML requer sempre que a sintaxe do código WML seja bem formatada, ou seja, não contenha informações que gerem erros com o DTD da linguagem. Um dos problemas enfrentados no desenvolvimento com o WML é que a aparência de algumas *tags é* diferente em *microbrowser* dos diversos fabricantes de dispositivos móveis. Outro problema é o suporte a caracteres com acento, pois é necessário utilizar uma codificação para que não ocorram problemas em alguns *microbrowsers.* A falta do suporte à multimídia traz uma grande limitação ao desenvolvedor no momento de desenvolver.

Com a evolução do WAP, e com a futura incorporação da linguagem cHTML no WML, os problemas e limitações vão diminuir, tornando o desenvolvimento menos complexo e com sistemas mais atraentes para os usuários.

4 *M-COMMERCE* (*MOBILE COMMERCE*)

O comércio eletrônico é considerado a terceira revolução depois da escrita e da invenção da máquina do prelo que proporcionou a revolução industrial. Não só por causa da redução dos custos de operação, mas também pela ampliação do mercado, porque uma empresa pode vender seus produtos para clientes de todo o mundo. Mas para uma empresa obter sucesso no comércio eletrônico, ela precisa ter uma estrutura bem definida e organizada, desde a produção, promoção, comercialização dos produtos, e principalmente a logística de entrega do produto. Pois um dos grandes problemas hoje do comércio eletrônico, além do medo da segurança existente nos sites, é a demora na entrega dos produtos, onde o sistema de logística é fundamental para satisfazer o cliente.

Existe uma grande variedade de sistemas de comércio eletrônico na Internet, seja *B2B, B2C, B2E, e-procurement, marketplace*, leilão virtual, cooperativas *on-line* de compras, etc. Todos em busca de fidelidade de consumidores com um poder aquisitivo alto, e redução nos seus custos de operação. Vamos verificar alguns desses modelos de negócios, antes de analisarmos o *M-commerce*, um termo ainda muito novo, mas que tem um grande potencial de crescimento.

4.1 SISTEMAS DE COMÉRCIO ELETRÔNICO

4.1.1 *Business-to-Business*

Segundo Albertin (2001), o *B2B* é o conceito para todas as transações de compra e venda de serviços, produtos e informações entre empresas via Internet. Segundo analistas, o *B2B* tende a crescer mais rapidamente do que o *B2C*, principalmente na relação entre as indústrias, pois elas possuem um planejamento e um programa de produção consistente, ou seja, sabem quanto vão produzir e consumir todos os dias. Já quando o *B2B* é entre a indústria e o varejo, os problemas de logística são maiores, pois o varejo não tem o total controle das vendas, portanto precisa de conexão mais flexível com a indústria, e a tecnologia do varejo é inferior a tecnologia da indústria. O *B2B* envolve os sistemas de *EDI*, transações integradas com sistemas de *ERP*, operações de crédito, etc.

4.1.2 *Business-to-Consumer*

O *B2C* é o conceito que envolve todas as transações de venda de produtos e serviços, entre a empresa e o seu cliente, utilizando a Internet. Nesse tipo de transação, o principal problema, e que dificulta o crescimento do *B2C*, é a confiança do cliente no sistema de

segurança do site de comércio eletrônico, além da facilidade de fazer a compra e o prazo de entrega dos produtos (Albertin, 2001).

4.1.3 *Business-to-Employee*

Com a tecnologia *wireless* surgiu um novo termo, chamado *B2E*, que é simplesmente a relação de negócios entre a empresa e seus funcionários, os quais poderão saber o que está acontecendo na empresa em qualquer lugar. A empresa por sua vez, vai saber onde está o empregado, e se ele está trabalhando, acessando o sistema, etc.

4.1.4 *E-Procurement*

Segundo Albertin, o *e-procurement* é uma espécie de leilão para a compra de produtos, sendo um dos principais recursos das grandes corporações e também dos governos para reduzir seus custos e melhorar o relacionamento com os seus fornecedores. O método adotado varia de acordo com o negócio e os investimentos disponíveis, mas o resultado obtido é sempre o mesmo, gerando uma redução dos preços em média 10% , além do maior número de fornecedores, e a maior agilidade no processo de compras, diminuindo a demora gerada por causa da burocracia. O que acontece é a substituição do processo de compras utilizando o telefone e fax, pela utilização de um sistema via Internet. Um *case* no Brasil é a empresa Volkswagen que utiliza o *e-procurement* na compra de todos os suprimentos da sua fábrica de São Paulo.

4.1.5 *E-MarketPlace*

O conceito de *e-marketplace* se refere aos portais que são utilizados para conectar fornecedores e compradores, onde são feitas as transações. Geralmente, esses portais são focados em um determinado mercado, por exemplo, o portal Latinexus, que é especializado na área de MRO, possibilitando ao pequeno e médio empresário fazer transações sem ter um sistema proprietário instalado, simplesmente vai ter uma *ASP (Application Solution Provider)* para as transações de compra e venda.

4.1.6 Cooperativas *On-line*

O conceito de cooperativas *on-line* ou *pool* de compras, surgiu da idéia de criar sites que oferecem produtos ou serviços, e quanto maior a procura por esses produtos, menor o preço final. O cliente se inscreve em um determinado grupo de compra, para comprar um determinado produto ou serviço, e quanto mais clientes se inscreverem para a compra desse produto ou serviço, mais barato ele se torna. O site fica responsável por fazer a negociação com o fornecedor do produto ou serviço e por entregar o produto ao cliente. Esse processo é semelhante ao *e-marketplace*, mas se destina aos consumidores finais (*B2C*). Um exemplo no Brasil é o ComDesconto que está em funcionamento desde dezembro de 1999.

4.2 SISTEMAS ELETRÔNICOS DE PAGAMENTO

Segundo Albertin (2001), as transações eletrônicas de comércio somente atingem o sucesso quando as trocas financeiras entre os compradores e os vendedores podem ser simples, seguro, barato e universalmente aceito. Os métodos de pagamento tradicional como

cheque, “doc” bancário e outros, não são adequados para a interação em tempo real dos sistemas de comércio eletrônico. Os sistemas de pagamento eletrônicos estão se tornando o ponto principal para as inovações nos negócios realizados pela Internet. O processo de pagamento e faturamento é um gargalo no rápido crescimento do comércio eletrônico.

Os principais sistemas de pagamento eletrônico são: o dinheiro eletrônico (*e-cash*), o cheque eletrônico, o cartão inteligente, o cartão de crédito e o cartão de débito.

4.2.1 Dinheiro Eletrônico

O dinheiro digital deve refletir as características fundamentais do dinheiro segundo a visão do consumidor: o anonimato, porque o comprador paga o vendedor, e apenas o vendedor fica sabendo dos detalhes da transação; e a liquidez, o dinheiro eletrônico seria aceito por todos as empresas relacionadas com o método de pagamento.

Existem dois tipos de dinheiro eletrônico: os cartões pré-pagos, um dos exemplos é os cartões de telefonia só que eles não tem liquidez; e os sistemas genuinamente eletrônicos, onde o dinheiro digital não existe na forma física, tornando-se útil nas transações via Internet, onde o dinheiro digital seria deduzido eletronicamente do comprador e creditado para o vendedor.

O *e-cash*, dinheiro eletrônico, é um novo conceito nos sistemas de pagamento eletrônicos pois combina as facilidades das redes de computadores com segurança e privacidade. Segundo Albertin (2001), o *e-cash* tem quatro propriedades: valor monetário; interoperabilidade; recuperabilidade e segurança. Algumas restrições que devem ser tomadas para diminuir os riscos com o *e-cash* são: limitar o tempo de validade do dinheiro eletrônico; o montante que pode ser armazenado e transferido pelo dinheiro eletrônico; o número de operações que podem acontecer com o dinheiro eletrônico antes que ele precise ser depositado novamente num banco; o número de transações que podem ser realizadas durante um período de tempo.

Para substituir o dinheiro em papel o dinheiro eletrônico tem que ter algumas qualidades típicas do dinheiro, que os cartões de crédito e débito não têm: ser negociável, ser moeda legal, ser um instrumento ao portador, poder ser guardado e utilizado por qualquer pessoa.

Segundo Albertin (2001), o Ecash é um sistema desenvolvido pela Digicash Co. Of Amsterdam e foi implementado por diversos bancos no mundo. Um estudo de caso da utilização do sistema Ecash foi o do banco Mark Twain Bank of Missouri dos Estados Unidos. Para poder realizar transações o comprador e vendedor tem que ter depósitos em contas no *WorldCurrency Acces* do Mark Twain Bank, sendo que essas contas *Acces* são como contas corrente normais e são garantidas pelo governo americano. O comprador pode então solicitar ao Mark Twain Bank para transferir fundos de suas contas *Acces* para suas contas Ecash. Em qualquer período o comprador pode acessar sua conta Ecash e retirar os fundos dessas contas e passar para os seus discos rígidos do seu computador pessoal, sendo o formato desses fundos são totalmente eletrônicos (binário) e criptografado. Para fazer um pagamento o comprador criptografa a quantidade de Ecash com um protocolo de criptografia seguro e envia o Ecash para o vendedor por um meio de comunicação de dados. O vendedor descriptografa o Ecash e armazena em seu disco rígido, depois ele acesso o banco para enviar esse Ecash e creditar em sua conta no banco.

Segundo Albertin (2001), o Ecash é privado, ou seja, mesmo com registros de retirada e depósito de Ecash, o banco não tem como saber onde o Ecash de um determinado cliente está localizado, sendo essa impossibilidade de rastrear o dinheiro é fundamental no sistema Ecash. O Ecash é baseado na criptografia por chave pública, sendo utilizada especificadamente o RSA da Data Security Inc..

O Anonimato nesse tipo de sistema traz riscos, pois se um *hacker* invadir o sistema toda a segurança do sistema é perdida, porque as transações não são confirmadas com um Banco Central.

4.2.2 Cartão Inteligente

Segundo Albertin (2001), os cartões inteligentes tem dinheiro armazenado, o saldo é guardado no próprio cartão e as compras realizadas são abatidas desse saldo. Esses cartões são muito utilizados em países como Alemanha, França, Japão para pagar ligações telefônicas públicas, transporte e programada de fidelidade de compradores.

Os cartões inteligentes oferecem benefícios tanto para vendedores como para compradores, reduzindo as despesas de manipulação de dinheiro e as perdas causadas por fraude, melhoram a facilidade e segurança de pagamento do comprador. Os cartões inteligentes podem ser de dois tipos: cartões de crédito baseado em relacionamento ou bolsas eletrônicas.

Um cartão inteligente baseado em relacionamento é uma melhoria dos serviços de cartões existentes e adição de novos serviços que uma instituição financeira fornece para seus clientes via cartão baseado em um *chip*. Os novos serviços são podem incluir acesso a múltiplas contas financeiras, programas de marketing, e outras informações que os portadores de cartões desejem armazenar.

O problema é que os cartões baseados em relacionamento utilizam crédito e o pagamento ocorre somente no final do ciclo de faturamento. Já as bolsas eletrônicas são cartões inteligentes adicionados de *microchips* programáveis que armazenam valores *e-cash* para as pessoas utilizarem em qualquer transação comercial. O comprador utiliza o cartão para realizar uma transação, e o vendedor com uma leitora de cartão de cartão verifica se o cartão é autentico e tem saldo suficiente para a transação. O valor da transação é então debitado no cartão inteligente e creditado em uma conta *e-cash* do vendedor.

Segundo Albertin (2001), os cartões inteligentes poderiam ser desenvolvidos para a Internet utilizando um *chip* de memória acoplado para armazenar uma chave privada de acesso e assinatura digital do usuário. E todas as autenticações e transações do usuário seriam realizadas através de uma leitora de cartão inteligente conectado em seu computador pessoal onde o usuário colocaria seu cartão inteligente, e entraria com o seu número de identificação pessoal (PIN) para realizar essa autenticação.

4.2.3 Cartão de Crédito

Segundo Albetin (2001), os cartões de crédito são responsáveis por mais de 90% de todas as transações realizadas na Internet. Sendo que as transações com cartão de crédito são consideradas mais seguras do que no mundo físico, mas certamente a segurança é um requisito fundamental para as transações comerciais eletrônicas.

A forma de utilização do cartão de crédito é muito simples, o vendedor mostra o produto e o seu valor, e o comprador confirma o pedido e entra com as informações do seu cartão de crédito para finalizar a transação.

Segundo Albertin (2001), os pagamentos com cartões de crédito nas redes *on-line* em três categorias:

- Pagamentos utilizando detalhes originais de cartões de crédito: é a troca de dados do cartão de crédito não criptografados sobre uma rede pública, seja telefônica ou a Internet. A utilização desse método não é recomendada pela falta de segurança e de privacidade;
- Pagamentos utilizando detalhes de cartões de crédito criptografados: faz sentido criptografar os dados do cartão de crédito antes de enviá-los, mas devem ser considerados aspectos como custo de transação;
- Pagamentos utilizando verificação de terceiros: uma solução para os problemas de segurança e verificação é a introdução de um terceiro na transação, que coleta e aprova os pagamentos de um cliente para o outro.

4.2.4 Cartão de Débito

O cartão de débito é uma nova forma de pagamento que está sendo testada e utilizada. Sendo a maior utilização em pontos de venda, como supermercados, postos de combustível, lojas de conveniência, entre outros.

A transação funciona como de forma semelhante à transação de cartão de crédito. Um exemplo seria: o vendedor passa o cartão em um terminal de transação onde as informações são lidas, o cliente entra com seu PIN e o terminal faz o roteamento através da rede para o banco do cliente para realizar a autorização da conta do cliente. Se aprovado pelo banco, os fundos são debitados na conta do cliente e transferidos para serem creditados no banco do vendedor. Essas transações são realizadas no sistema bancário e a segurança nessas transações é garantida pelos bancos.

Segundo Albertin (2001), as vantagens de utilizar os cartões de débito são: baixo custo por transação, maior conveniência do que a utilização do dinheiro em papel e melhor conveniência para as instituições que os utilizam. Para o cliente também é mais vantagem que a utilização de cheques, pois é um sistema mais seguro e mais fácil de utilizar.

4.3 ESTUDO SOBRE *M-COMMERCE*

O *e-commerce* já está sendo utilizados há alguns anos, mas somente nos últimos dois anos se solidificou como um grande mercado. Com o *M-commerce*, começa a surgir um novo tipo de mercado para a venda de produtos e serviços que pode ser feita de uma forma diferenciada, agregando novos valores, fazendo com que novos clientes façam compras eletrônicas, só que agora com equipamentos móveis. Como as projeções do crescimento do comércio eletrônico indicam um forte incremento nos próximos anos, todas as corporações no mundo já estão com sites de comércio eletrônico ou projetos para terem seus sites no ar. É um mercado muito importante, pois nos Estados Unidos a Internet movimenta US$ 301 bilhões, ficando atrás somente da indústria automobilística que movimenta US$ 350 naquele país, segundo os dados do Fundo Monetário Internacional e da Universidade do Texas no ano de 1999.

Baseados nesses números, os novos segmentos como o *M-commerce* tende a receber grandes investimentos, pois é um mercado ainda não explorado. Até o momento, o *M-commerce* no Brasil é utilizado principalmente em serviços de *home banking*. Praticamente todos os grandes bancos que atuam no Brasil, como o Banco do Brasil, Real ABN, Bradesco, Itaú já têm disponível o serviço de *mobile banking* para seus clientes, onde todas as transações bancárias podem ser feitas via Internet móvel. Sendo necessário nesse tipo de transação ter uma preocupação e muitos investimentos em relação à segurança, pois o sistema trabalha diretamente com as contas dos clientes, e qualquer falha nesse sistema pode desacreditar o produto e o próprio banco perante seus clientes. Além de trazer mais um serviço importante para o cliente, possibilitando que ele faça suas transações em qualquer lugar, a qualquer hora, o banco tem uma grande vantagem, pois o custo de uma transação na agência custa US$ 1,10 enquanto que uma transação via Internet móvel custa US$ 0,01, sendo que economia compensa todos os investimentos realizados pelos bancos, além de oferecer melhores serviços aos seus clientes, e obter vantagem em relação à concorrência.

4.3.1 *Cases* de *M-commerce* na Europa

Na Europa, a utilização da Internet Móvel é muito grande. Um exemplo é que o telefone celular liderou a lista de "brinquedos" no natal de 2000 na Inglaterra. Praticamente todos os telefones celulares vendidos são compatíveis com a tecnologia WAP, e existem muitas aplicações já disponíveis. Existem diversas publicações de revistas e livros que tratam do *M-commerce* na Europa. As empresas de telecomunicações acreditam tanto no desenvolvimento da Internet Móvel que na Inglaterra o leilão das licenças da tecnologia 3G rendeu 35 bilhões de dólares ao governo, somente para a utilização do espectro de rádio. Esse valor corresponde a 600 dólares por habitante da Inglaterra (Dornan, 2001). Mais os custos de construção da infra-estrutura, marketing e o subsídio dos telefones celulares e esse valor pode triplicar.

Mesmo assim, as empresas de telecomunicações não estão fazendo grandes investimentos para perder dinheiro. Com o crescente número de usuários de telefone celular e com o aumento da utilização da Internet Móvel, os gastos por cliente estão aumentando. Porém, as empresas acreditam que a receita para pagar todos esses custos e trazer lucros é o *M-commerce*. Segundo Dornan (2001), as pesquisas indicam que em 2003 existirão um bilhão de usuários de dispositivos *wireless* e a grande maioria com acesso à Internet, e que eles serão responsáveis pela movimentação de um trilhão de dólares no mundo.

Para um negócio ter sucesso na Internet Móvel utilizando a tecnologia disponível atualmente, alguns modelos de negócio têm mais chances de obter sucesso do que outros. A urgência faz com que seja um tipo de negócio interessante para transações em que o tempo é importante, por exemplo, um sistema de leilão on-line ou venda em grupo (cooperativa) on-line. Já um tipo de negócio de venda de livros ou hardware pode não obter sucesso. Outro fator importante considerado é a facilidade de realizar uma transação, pois com a limitação do tamanho da tela e a dificuldade para fazer a entrada de dados, o sistema deve ser simples, com poucas telas e com a mínima entrada de dados do usuário (Dornan, 2001).

Ainda assim, muitas empresas portaram seus modelos de negócio da Internet *Web* para a Internet Móvel, pois os custos envolvidos são relativamente baixos, e como a tecnologia *wireless* é considerada como o próximo grande acontecimento no mundo da tecnologia, as empresas, principalmente as empresas "pontocom", precisam ser vistas desenvolvendo algum sistema *wireless*, para manter a reputação no mercado de tecnologia. Esses negócios portados sem uma análise profunda têm grandes chances de fracasso, pois os modelos de negócio da

Internet Móvel ainda não estão bem definidos, mas são muito diferentes da Internet *Web*. Os negócios *wireless* que obterão mais sucesso serão aqueles que trarão diferencias em relação à Internet Web e também ao comércio tradicional, utilizando a vantagem do sistema móvel de ser acessível em qualquer lugar e a qualquer hora. Porém, a maioria dessas aplicações ainda não foi inventada, como aconteceu no princípio da Internet Web, e muitas serão inventadas somente depois do amadurecimento da tecnologia da Internet Móvel.

Desde 1999, muitas aplicações *wireless* estão em utilização na Europa, disponibilizando o acesso a dados e serviços em grandes corporações. Alguns *cases* são: o banco Deutsche Bank, o banco Nordea e a rede de hotéis Scandic Hotels. O Deutsche Bank, um banco que tem mais de 800 mil clientes que acessam o seu sistema de *home banking* on-line pela Internet, com a explosão dos serviços baseados em WAP em 1999, resolveu iniciar um projeto piloto em setembro de 1999. O banco utilizou a tecnologia desenvolvida pela Nokia para disponibilizar o acesso aos seus clientes dos serviços de *home banking* em seus celulares. Foi disponibilizado acesso à consulta de saldos da contas, transferências de valores, pagamento de contas, consulta sobre preço de ações. Os serviços são acessados com total segurança, garantida com a utilização do *WTLS* e do *SSL* para prover criptografia ponto-a-ponto nas transações. Os servidores *Gateway Wap* ficaram localizados dentro do *Deutsche Bank*, garantindo assim total confiança na segurança dos dados. O *Deutsche Bank* utilizou o *Nokia Active Server* para disponibilizar o acesso ao sistema de *home banking*, fazendo também a autenticação dos usuários através desse servidor. Os diretores do banco enfatizaram que o serviço permite que os clientes possam verificar a cotação de suas ações enquanto estão passeando em um parque, ou fazer transferências de fundos enquanto estão indo para casa de ônibus.

Já o sistema implementado pela Scandic Hotels, um grande grupo de hotelaria que tem 150 hotéis em 10 países diferentes na Europa. Em outubro de 1999, o grupo resolveu disponibilizar através de um sistema móvel a possibilidade de seus clientes fazerem uma reserva, alterar uma reserva, consultar sobre os quartos disponíveis, consultar as ofertas e informações diversas sobre turismo. Em janeiro de 2000, o Scandic Hotels foi a primeira companhia de hotelaria no mundo a oferecer a facilidade de realizar praticamente todas as transações em um telefone móvel WAP. Um dos grandes benefícios para o cliente foi que, quando se confirma uma reserva em um hotel, o sistema gera automaticamente uma mensagem SMS com todas as informações sobre a reserva e envia para o celular do cliente.

4.3.2 *Cases* de *M-commerce* no Brasil

No Brasil, já existem *cases* de *M-commerce.* Claro que em um mercado tão novo, e mesmo com tantas limitações tecnológicas, a tendência é de crescimento do *M-commerce*. Durante a quinta Jornada de Atualização Tecnológica, realizada pela Softsul em Porto Alegre nos dias 22 e 23 de novembro do ano 2000, foram apresentadas várias experiências e cases de empresas que já estão trabalhando com a Internet Móvel.

Uma palestra muito interessante sobre comércio eletrônico e área governamental, ministrada Everton Hagen, da Procergs, e que é um dos fundadores do serviço Via-RS, serviço pioneiro de acesso a *Web* comercial no Rio Grande do Sul. Ele demonstrou algumas das possibilidades da tecnologia *wireless*, por exemplo, os serviços de consulta da situação legal de um veículo por um usuário com um celular, como já está sendo utilizado pela polícia de Minas Gerais, ou a consulta de processos judiciários por advogados cadastrados no sistema da Procergs. A consulta de pessoas procuradas por um policial via celular em uma barreira policial é um serviço muito importante para o Estado. Hagen salientou que o governo do

estado do Rio Grande do Sul tem vários projetos voltados para o uso de tecnologias *wireless*, mas lembra que a filosofia utilizada hoje é sempre utilizar tecnologias livres, não proprietárias, para não ficar dependente de um fornecedor de soluções tecnológicas.

Outra palestra muito interessante foi sobre a utilização de Agentes Inteligentes juntamente com a tecnologia *wireless*, ministrada por Rogério Figurelli, dono da empresa Plugar, que trabalha no desenvolvimento de sistemas que utilizam agentes inteligentes na *Web*, e agora também na Internet móvel. Figurelli frisou que a única maneira de realmente ter uma personalização do serviço oferecido ao cliente é com o uso de agentes inteligentes que garantem uma informação que realmente interessa ao cliente, procurando e filtrando o conteúdo, e fornecendo somente o que o cliente realmente espera receber no seu celular.

Um case apresentado e muito importante foi o TAM *WAP-Ticket*, da companhia área TAM, apresentado por Marcos Teixeira, gerente de tecnologia de informação da empresa. Esse sistema tem duas funcionalidades básicas já implementadas, a consulta dos horários dos vôos entre as principais capitais e a compra de *tickets* para os vôos via dispositivo *WAP*. A TAM fez uma parceria com a empresa de telefonia Telesp em São Paulo que disponibiliza a seus assinantes *WAP* um link para a página *WML* da TAM e também fornece o serviço de *Gateway WAP* para a TAM. O pagamento do *ticket* pode ser pago somente com cartão de crédito. As principais vantagens que a TAM disponibiliza nesse tipo de compra são: o menor preço do *ticket* realizado no mercado pelas agências de viagens, e o passageiro depois de comprar o *ticket* via *WAP*, recebe o número do *ticket* e após vai diretamente para o setor de *ckeck-in*. Uma desvantagem é que o cliente deve se cadastrar primeiro no site *Web*, para depois poder acessar o site *WAP*. O sistema entrou em operação em setembro de 2000, e a venda de *tickets* via *WAP* ainda é muito pequena, principalmente porque a TAM não divulgou o seu sistema para seus clientes. Um dos pontos salientados por Teixeira, em que a TAM deve melhorar, é o seu marketing em relação aos sistemas e serviços disponíveis. Para o futuro, a TAM pretende utilizar o *WAP* e também a *Web* para personalizar cada vez mais seus serviços oferecidos aos seus clientes, garantindo assim a satisfação desses clientes e sua fidelidade.

Segundo Teixeira, o sistema *WAP-Ticket* da TAM está baseado nas camadas implementadas com o *BroadVision*, conforme a Figura 15 a seguir.

Essa arquitetura utilizada pela TAM pode ser utilizada em qualquer outra aplicação para implementar um site de *M-commerce*. A TAM utilizou as tecnologias XML, JAVA/JSP e C++/CORBA para implementar o seu sistema de vendas de *tickets* via *WAP*, com uma arquitetura em camadas separadas. Conforme Teixeira, outra definição que a TAM fez foi utilizar o *Gateway WAP* da própria operadora de telefonia. Essa definição é muito importante no aspecto de segurança, pois uma das vulnerabilidades do *WAP* é justamente na conversão dos pacotes feita pelo *Gateway WAP*.

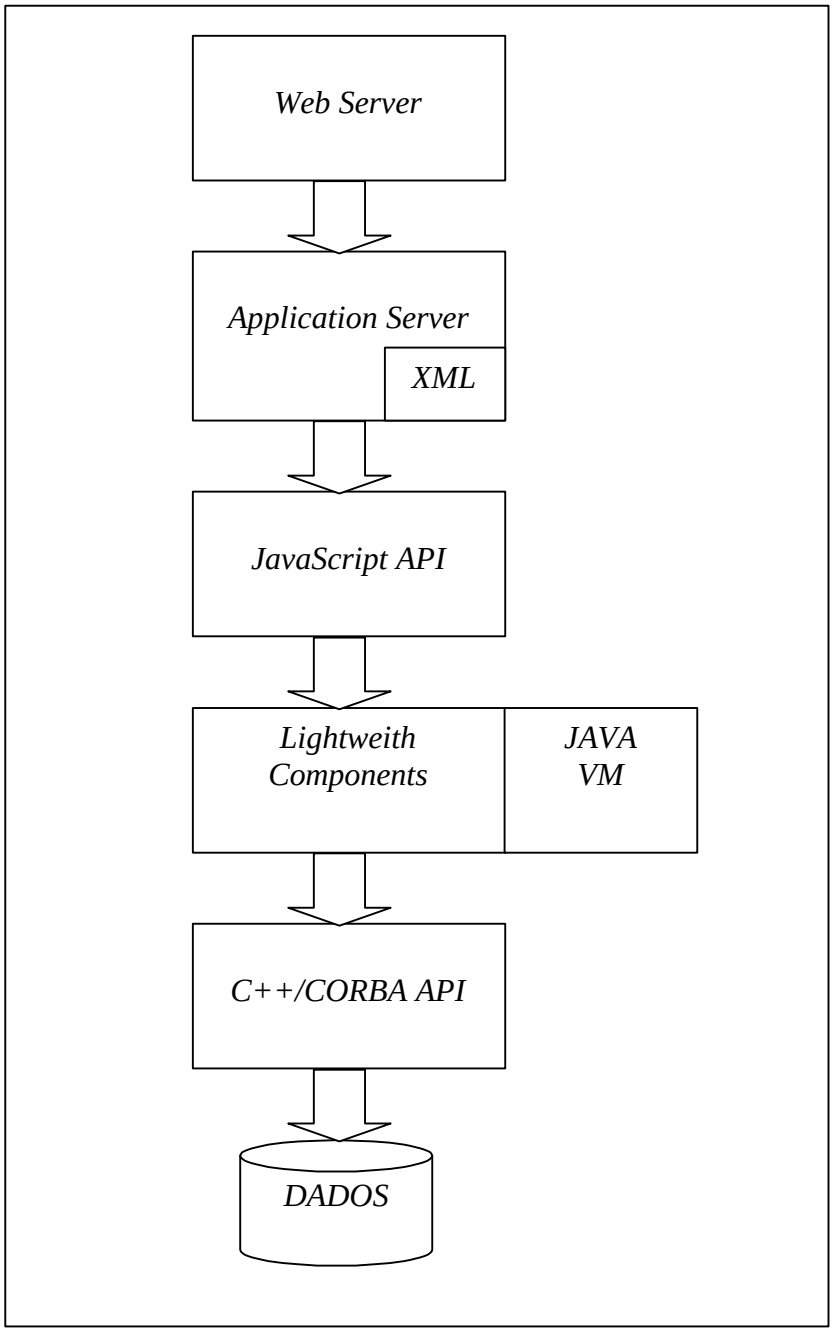


Figura 15 – Estrutura de Camadas Utilizada pela TAM no Projeto WAP-Ticket

4.3.3 Problemas no *M-commerce*

Existem alguns problemas para serem resolvidos na utilização da Internet Móvel para fazer o comércio eletrônico. Um dos grandes pontos positivos da tecnologia móvel é saber onde o cliente se encontra, com que fala, o que ele comprou no último acesso, em quais sites ele visitou. Essas informações são muito valiosas para a publicidade de um site *M-commerce*, porque com essas informações uma empresa pode oferecer um determinado produto para uma pessoa que gosta desse produto, ou que está localizado perto de uma loja que tem um determinado produto.

Segundo Dornan (2001), as operadoras de telefonia móvel podem registrar a localização exata de cada cliente, desde que o telefone esteja ligado e na área de cobertura. O cruzamento de dados com um mapa permite descobrir em que período do dia o cliente está em casa, o que ele faz em seu tempo de lazer, quais as lojas que ele visita normalmente. Com essas informações, vários tipos de análises podem ser feitos, como descobrir os amigos desse cliente, os gostos compartilhados desses amigos. Essas análises são possíveis tecnicamente, mas as operadoras não fazem por dois grandes motivos: o custo das ferramentas de exploração de dados é muito alto, e principalmente porque as informações sobre os clientes são secretas, e a maioria dos clientes não gostaria de ser "vigiada" pela empresa telefônica. Por isso, as empresas preferem não correr o risco de uma reação dos clientes, podendo essa reação ser até mesmo judicial.

Também os serviços de localização serão mais avançados no futuro, pois em vez de carregar um mapa com as ruas, as pessoas poderão apenas pressionar um botão em dispositivo móvel e visualizar o mapa completo com efeitos de zoom e uma seta apontando a localização exata, exatamente como ocorre com os receptores GPS conectados a um computador laptop que tem todos os mapas armazenados na memória. Com a miniaturização, a integração do dispositivo móvel com o receptor GPS poderá ocorrer, ou utilizará a própria rede celular. Com o aumento da banda, os mapas poderão ser transferidos por *download* e ficarem armazenados na memória localmente.

Outro problema existente na Internet Móvel e que dificulta o desenvolvimento do *M-commerce* é a falta de utilização do conteúdo multimídia nas aplicações, pois com as limitações das tecnologias existentes, é impossível utilizar imagens, vídeo e sons na Internet Móvel. A falta de imagens coloridas, som e vídeo causam frustração à maioria dos usuários, fazendo com que muitos não comprem um determinado produto porque o sistema não tem uma foto colorida demonstrando o mesmo.

Segundo Dornan (2001), outra área prejudicada com a falta de conteúdo multimídia e que tem um grande número de usuários é a área de jogos. Já existem muitos jogos disponíveis para WAP, mas são jogos simples em modo texto, com as imagens em *wbmp* que é o único formato suportado pelo WAP. Porém, com a utilização da comunicação baseada em pacotes existe a possibilidade de criar jogos com competição entre jogadores, tornando assim os programas bem mais atrativos. No Japão, os jogos para o *i-mode* , rede móvel daquele país, foram um dos grandes responsáveis pela explosão de utilização do acesso à Internet Móvel, existindo mais de 30 milhões de usuários atualmente, sendo que o *i-mode* opera com a banda de 9.6 *kbps.*

Outro problema na Internet Móvel que afeta também o *M-commerce* é o modelo de bilhetagem utilizado para cobrar o acesso à Internet pelos dispositivos móveis. Esse também é um dos grandes fatores que fez com que o protocolo WAP não tenha uma explosão de vendas no Brasil. Hoje, existem dois modelos de cobrança existentes, um proveniente do sistema de telefonia e o outro proveniente do sistema da Internet. As operadoras telefônicas cobram por minuto ou uma outra unidade de tempo. Sendo que isso é adequado em relação à comunicação com circuito comutado, pois o circuito precisa ser mantido aberto, esteja ele sendo usado ou não. Na Internet, os grandes provedores que possuem grandes *backbones* cobram pela quantidade de bits enviados pelas suas redes. Também as operadoras móveis utilizam esse tipo de cobrança em relação ao serviço de mensagens SMS (Dornan, 2001).

Esses dois modelos de cobrança não vão se sustentar no longo prazo, tanto no mercado de telecomunicações fixa como no móvel, pois a capacidade de banda se tornará tão barato que não será medida. Assim as operadoras farão a cobrança referente ao acesso ou ao

conteúdo. Muitas pessoas ou empresas já pagam tarifas fixas referentes ao acesso à Internet, dependendo do tipo de conexão que elas possuem. Esse tipo de modelo ainda não é utilizado na Internet Móvel. Empresas telefônicas, como a NTT DoCoMo do Japão, utilizam seus sistemas de bilhetagens existentes para cobrar pequenos pagamentos por conteúdo que um determinado cliente tenha acessado. Eles descobriram que as pessoas concordam em pagar a mais por serviços extras, como por exemplo, cotações de ações e horóscopo.

4.4 USABILIDADE DE UM SITE PARAR *M-COMMERCE*

A usabilidade de um sistema indica o grau de amigabilidade com o usuário do sistema em questão. A facilidade de utilização é um fator muito importante na usabilidade de um sistema. A interface da aplicação com o usuário é fundamental para o sucesso de um sistema ou seu fracasso, tanto que existem áreas da ciência da computação que estudam somente o desenvolvimento de interfaces com usuário. Os sistemas utilizam cada vez mais interfaces com usuários mais poderosas, com a grande utilização de gráficos, vídeo e sons. Mas os desenvolvedores de sistemas que utilizam WAP ainda não têm suporte para esses recursos multimídia, e ainda têm que enfrentar as limitações impostas pelos dispositivos móveis e pela banda de dados.

Por isso, o desenvolvimento de um sistema WAP deve ter algumas características básicas para não frustrar os usuários. Antes de começar a desenvolver o sistema, algumas questões devem ser pensadas: qual será o nível de conhecimento necessário do usuário para utilizar o sistema ? Quanto tempo o usuário precisa para aprender a utilizar o sistema? Qual o objetivo do sistema e qual o nível de frustração envolvido com a utilização do sistema ? (Arehart, 2000). Essas questões permitem que o desenvolvedor analise como deve ser projetado o sistema para que sua usabilidade fique aceitável pelos usuários.

As limitações dos equipamentos wireless já foram citadas no trabalho. Mas para o desenvolvedor, essas limitações são fundamentais no projeto de um sistema WAP. As telas dos telefones celulares são pequenas, normalmente com quatro linhas e quinze caracteres por linha, por isso, no desenvolvimento, deve ser observada essa restrição para garantir a utilização em qualquer dispositivo móvel. A entrada de dados é outra limitação, e com certeza para o desenvolvedor é uma grande limitação, pois os teclados dos telefones celulares possuem somente teclas numéricas, e para digitar um caracter do alfabeto é necessário mais de um pressionamento em uma tecla. A limitação de poder de processamento nos telefones celulares implica que a utilização da WMLScript no dispositivo deve ser feita com precaução, pois o processamento é muito limitado. A limitação de banda é outro fator importante no desenvolvimento de um sistema, pois a taxa de transferência atual de 9.6 *kbps* torna o acesso ao conteúdo lento; porém, isso deve ser alterado logo com a utilização do *GPRS* que utiliza a comutação por pacotes, e a transmissão de dados pode chegar a 114 *kbps*.

Outro problema no desenvolvimento WAP é que os dispositivos móveis suportam as especificações WAP, mas na prática a apresentação do conteúdo é diferente em modelo de dispositivo para outro. Isso afeta diretamente a usabilidade de um sistema. O ideal é na fase de desenvolvimento utilizar vários emuladores para testar o comportamento do sistema nos diversos dispositivos móveis, e, se preciso, adicionar ou retirar um determinado recurso para garantir a usabilidade, ou criar várias versões da aplicação para rodar em um determinado *microbrowser WAP*, sendo que os *microbrowsers* mais utilizados são o da Nokia e da Phone.com (Forta, 2001).

Um fator muito importante no desenvolvimento de um sistema WAP é pensar quem serão os usuários do sistema que será desenvolvido. O desenvolvedor deve ter a consciência que o usuário do sistema WAP não é necessariamente um usuário de um microcomputador. O usuário talvez nunca tenha utilizado um microcomputador, e isso torna fundamental o desenvolvimento de um sistema amigável com o usuário. Se o desenvolvedor conseguir identificar o perfil dos usuários que utilizarão o sistema, essa informação é muito importante na estruturação e decisão de funcionalidades do sistema.

4.4.1 Características Básicas de um Sistema Móvel

Para desenvolver um sistema utilizando WAP com uma boa usabilidade e em conseqüência um bom sistema, é necessário utilizar uma metodologia para o desenvolvimento do sistema. Para o desenvolvedor, não é fácil pensar na usabilidade de um sistema WAP. Muitas questões em que o desenvolvedor fica pensando: a aplicação é fácil de utilizar? A aplicação é intuitiva? A aplicação é eficiente? Ela tem o mínimo possível de entrada de dados do usuário ?

Segundo Arehart (2000), as principais características que uma aplicação WML deve possuir são as seguintes:

- garantir as funcionalidades essenciais do sistema, pois o desenvolvedor deve assegurar que o sistema tenha as funcionalidades que o usuário necessita, e não como ocorre na Web, onde o desenvolvedor, além das funcionalidades essenciais pode adicionar muitas funcionalidades extras.
- priorizar a fase de análise da aplicação, desenhar o sistema como uma estrutura de árvore, para facilitar o acesso às funcionalidades mais importantes e mais utilizadas no sistema.
- minimizar a entrada de dados pelo usuário é outra característica muito importante, pois os dispositivos móveis atuais não foram desenhados para a digitação de dados. Por esse motivo, a maioria dos sites wireless faz o cadastro dos usuários através da Web.
- a validação dos dados que o usuário informa também é uma característica importante no sistema, para evitar que o usuário entre com dados errados no sistema, podendo ser utilizada a linguagem WMLScript para validar no lado cliente, mas muitos dispositivos móveis não reconhecem essa linguagem
- a possibilidade de “voltar” para as páginas anteriores é sempre uma característica que o usuário utiliza, como ocorre na Web, só que no WML não existe a opção de “avançar” nas páginas em cache.

4.5 CONCLUSÃO SOBRE O *M*-COMMERCE

Analisando o que está acontecendo agora com o *M-commerce*, conclui-se que por ser uma área de comércio eletrônico muito nova, principalmente no Brasil, e que por ainda não existirem muitos sites especializados nessa área, o volume de transações é muito pequeno. Este foi um dos motivos para desenvolver o sistema de *M-commerce* E-Móvel nesse trabalho de conclusão.

Por ser uma tecnologia tão nova, o *WAP* ainda não traz o desempenho ideal para se implementar um site de *M-commerce* com muita informação, dicas, etc. O site tem que ser o mais “limpo” possível, e ser fácil de utilizar e principalmente fácil de comprar, pois um dos

grandes problemas dos sites de comércio eletrônico é que o cliente não gosta de ficar procurando entre os diversos *links* o produto desejado. Porém, existem muitas oportunidades de criar novas coisas, novos tipos de negócios, como a TAM fez com o *WAP-Ticket*, mesmo que a venda seja hoje de um bilhete por dia, o que importa é o diferencial, e assim que o mercado de *M-commerce* se firmar, as empresas que estão a mais tempo nesse negócio, terão muito mais chances de liderar em seu segmento de mercado do que seus concorrentes.

Outra limitação do WAP, que acaba tornando difícil o desenvolvimento de um sistema atraente para os usuários, é a falta de recursos multimídia. No Japão, a NTT DoCoMo utiliza o cHTML com suporte para alguns recursos multimídia como imagens gráficas coloridas e sons, sendo que esses recursos foram fundamentais para que a Internet móvel no Japão se tornasse uma realidade e fosse utilizada por milhões de usuários.

Esse fato fez com que o WAP Fórum inclua na versão 2.0 do WML características do cHTML, para suportar o conteúdo multimídia nas aplicações WAP. Com o aumento da banda de dados, com a inclusão do conteúdo multimídia, com as redes por pacotes e a alteração do método de cobrança do acesso *wireless* a dados, o *M-commerce* deve se tornar realidade.

5 SEGURANÇA NO *M-COMMERCE*

Com o uso da computação na área comercial, a complexidade na parte de segurança aumentou, pois uma única máquina de uma grande loja pode ter milhares de pessoas acessando simultaneamente, e muitas pessoas também tentando encontrar uma brecha no sistema para violar o mesmo. Mas hoje o problema não é somente em grandes empresas, pois qualquer empresa conectada na Internet tem que se preocupar com a segurança. Com o *WAP* e os novos mercados criados, o contexto da segurança deve ser colocado como prioridade na implementação de um *M-commerce*.

Um dos grandes problemas, que existem ainda hoje no comércio eletrônico em geral (*e-commerce ou M-commerce),* é a desconfiança das pessoas sobre a confiabilidade dos sistemas, e a falta de segurança em relação aos seus dados, por exemplo, o número do cartão de crédito. Essa preocupação realmente existe, porque o processo de segurança tem que ser muito complexo, para garantir a confiabilidade das transações, e depois a segurança dos dados armazenados, e também de documentos guardados na empresa de comércio eletrônico.

A necessidade de segurança não ocorre somente no *e-commerce* ou no *M-commerce*, pois hoje, em todas as instituições a integridade dos dados, e como manter esses dados secretos é prioridade máxima, e não somente transações comerciais. Mercados relacionados com inteligência, por exemplo, têm um valor muito alto comercialmente, pois pode fazer a diferença frente às concorrentes. É claro que essas companhias precisam proteger os seus sistemas de pessoas que tenham intenção de invadir seus sistemas. Assim, a segurança pode ter duas regras para ser mais confiável: habilitando mais alguma tecnologia, ou desabilitando alguma tecnologia.

Todas as pessoas sabem da necessidade de segurança robusta em um sistema, mas precisa ser analisado todo o contexto do ambiente a ser implantada a segurança. A segurança em um sistema, pela sua própria natureza, é muito difícil de ser testada. Um produto pode ser implementado seguindo toda uma especificação de segurança, pode ser escrito e testado, mas, mesmo assim, ele vai ser vulnerável a ataques. Os ataques são sempre inesperados, e podem ser de diferentes tipos, utilizando técnicas variadas. Nunca se pode dizer que uma aplicação é 100% segura, e sim que ela tem um nível de segurança, que evita ser tipos de ataques, e que não tem certos tipos de vulnerabilidades. Pode-se pensar que uma determinada aplicação seja segura para o seu propósito em particular, e pode até ser afirmado como seguro, por exemplo, o padrão atual de segurança fixa e móvel, o *TLS (Transport Layer Security)* e o seu equivalente *wireless WTLS (Wireless Transport Layer Security)* são seguros para muitas pessoas e organizações que realizam negócios com segurança na Internet fixa ou *wireless*. Mas isso não garante que essas tecnologias tenham situações não previstas, e que muitos *hackers* devem estar tentando quebrar nesse momento.

Segundo Schneier (2001), a segurança é uma corrente, e ela é tão segura quanto o seu elo mais fraco. Ou seja, a segurança deve ser analisada no ambiente como um todo, pois os maiores problemas de segurança não ocorrem com a matemática da criptografia, que atualmente são muito seguro, e sim no hardware, no software, nas redes e nas pessoas que utilizam um sistema. Qualquer sistema do mundo real, é uma complicada série de interconexões, sendo que a segurança precisa atender o sistema: seus componentes e conexões.

5.1 CONCEITOS DE SEGURANÇA

Na parte de segurança existem os conceitos básicos para garantir o nível mínimo de segurança aceitável. Segundo Schneier (1996), as principais características que um sistema deve possuir para ter o mínimo de segurança são: Autenticação, Integridade, Privacidade Autorização e Não-Repudio.

5.1.1 Autenticação

A autenticação é um processo bastante complexo, e não somente no mundo dos computadores. No mundo, a autenticação pode ser feita pela voz da pessoa, ou quando se olha para uma pessoa, e isso ocorre várias vezes por dia, por todas as pessoas. Outros animais utilizam protocolos de autenticação diferentes, como o olfato. Existem muitos tipos diferentes de autenticação para uma transação ou interação, e diferentes tipos de mecanismos são utilizados para diferentes tipos de indivíduos e organizações.

Em muitos casos é necessário identificar pessoas que nunca foram vistas antes, e nesses casos são utilizados os *token*s para validar a pessoa. Um *token* tem certas características que precisam ser aceitas como uma autenticação por *token*: ele é usado para ser reconhecido como um Certificado de Autoridade (CA), por exemplo, o governo, que tem uma política onde as pessoas têm uma identidade pessoal que é válida somente quando tem uma fotografia ou às vezes a assinatura. Um passaporte é um exemplo claro de uma autenticação por *token*. Similarmente, certificados digitais e assinatura digitais podem ser utilizadas para autenticar participantes de transações eletrônicas. O objetivo do protocolo de autenticação é capturar atividades chamadas de *spoofing*. Isso ocorre quando uma pessoa tenta se passar por outra. O *spoofing* é um problema que não ocorre no começo das transações, e sim no decorrer das transações. A maneira de diminuir esse problema, e fazer uma re-autenticação dos participantes durante a transação.

Segundo Schneier (1996), o método mais comum utilizado para fazer autenticação em sistema sistemas digitais é através da utilização de senhas.

5.1.2 Integridade

Segundo Schneier (1996), existe a possibilidade de que as mensagens podem ser interceptadas ao longo do caminho de comunicação, e também deve ser assumida a possibilidade de alteração dessas mensagens. Também deve ser garantida, a integridade da mensagem no caso de uma parte da mesma se perder no caminho.

5.1.3 Privacidade

A privacidade é uma das partes mais importante na segurança. Depois da autenticação concluída, a transação deve ser garantida que as informações das partes envolvidas na transação não sejam lidas por outras pessoas não autorizadas. É notória a facilidade para fazer uma interceptação de comunicação digital, e praticamente impossível de se impedir. Então para garantir a privacidade dos dados, eles são usualmente encriptados. A encriptação é o processo da codificação da informação para uma representação diferente que a utilizada normalmente.

5.1.4 Autorização

A autorização define quando um processo tem uma parte particular de direitos para uma ação com um respectivo objeto particular, em uma situação particular. Todos essas variáveis, devem ser levadas em conta no momento à conta de usuário e os seus direitos de acesso ao sistema. Para que a autorização seja garantida com sucesso, um mecanismo eficiente de autenticação deve ser utilizado.

5.1.5 Não-Repudio

Segundo Schneier (1996), um emissor não pode falsamente negar depois que ele enviou uma mensagem. Para isso, é utilizado um certificado digital de um terceiro confiável para assinar a mensagem. O serviço deve ser disponível somente para as pessoas autorizadas, e negado para as pessoas que não são autorizadas.

5.2 CRIPTOGRAFIA

A criptografia sempre foi uma área muito importante para os serviços de inteligências militares dos países. Nos Estados Unidos, ela foi muito utilizada durante a primeira e a segunda Guerra Mundial. Mas depois da segunda Guerra, a criptografia foi classificada como munição pelo governo dos Estados Unidos, e isso tornou ilegal a exportação de qualquer software ou código-fonte. Naquela época, as pessoas não precisavam utilizar criptografia, mas hoje a criptografia é grande área de negócios e muitas empresas e pessoas precisam utilizar para proteger seus dados e seus negócios.

Segundo Schneier (1996), a criptografia é o processo de encriptar uma mensagem ou texto simples para um texto cifrado que é inelegível para as pessoas. Um algoritmo criptográfico é uma função matemática usada para encriptar e para descriptar, mas podem ser duas funções: uma para encriptar e outra para descriptar.

Os algoritmos mais antigos eram mantidos em segredo e eram chamados de algoritmos restritos. Os algoritmos restritos eram mantidos em segredo para dificultar a quebra por pessoas não autorizadas. Eles eram utilizados por poucas pessoas, pois se alguém revelasse o segredo, todos teriam que alterar o algoritmo. Esses algoritmos não eram seguros para aplicações críticas, e eram utilizados para aplicações que necessitavam de baixa segurança.

Segundo Schneier (1996), os algoritmos modernos de criptografia utilizam uma chave (*key*) para resolver o problema do conhecimento do algoritmo. Essa chave pode ter um grande número de valores, e esse *range* de possibilidades dos valores da chave é chamado de espaço da chave (*keyspace*). Ambas as operações de encriptar e descriptar usam essa chave. Porém,

muitos algoritmos utilizam uma chave para encriptar diferente da chave para descriptar. Toda a segurança desses algoritmos é baseada na chave (ou chaves) e não baseados nos detalhes do algoritmo. Por isso, os algoritmos podem ser publicados e analisados sem problemas.

5.2.1 Algoritmos Simétricos (Chave Privada)

Existem dois tipos de algoritmos baseados em chaves: os algoritmos simétricos e os baseados na chave pública. Os algoritmos simétricos, também chamados de algoritmos convencionais, são os algoritmos onde a chave criptográfica é compartilhada entre o emissor e o receptor. A segurança de algoritmos simétricos está na chave. Possuindo a chave, uma pessoa pode criptografar ou descriptografar uma determinada mensagem que foi criptografada com essa chave.

Segundo Schneier (1996), a criptografia e a descriptografia com um algoritmo simétrico pode ser demonstrada da seguinte forma:

$E_K(M)=C$

$D_K(C)=M$

Onde E é a função para criptografar, o D significa a função para descriptar, o K representa a chave, o M representa a mensagem e o C representa a mensagem cifrada.

Os algoritmos simétricos podem ser divididos em duas categorias. Algumas operam em um *bit* (pode ser *byte*) por vez em texto simples. Esses algoritmos são chamados de algoritmos de fluxo (*stream algorithms*). Os outros algoritmos operam em grupos de bits chamados de blocos, e os algoritmos são chamados de algoritmos de bloco (*block algorithms*). Em algoritmos de computadores modernos, o tamanho de um bloco normalmente é de 64 *bits.*

Segundo Schneier (1996), o problema da utilização da chave privada na comunicação de dados, é garantir que a chave privada seja transmitida de um lado para o outro sem que algum bisbilhoteiro intercepte a mensagem e consiga a chave privada. Então uma alternativa é utilizar o algoritmo da chave pública para garantir a segurança na transferência da chave privada de um lado para o outro da comunicação.

Um exemplo bem simples, Alice e Bob querem se comunicar utilizando a criptografia da chave privada. Alice pergunta para Bob sobre o certificado que contém a chave pública dele. Alice então cria uma chave privada nova, criptografa essa chave com a chave pública de Bob e envia o resultado para ele. Bob utiliza sua chave privada para descriptografar a mensagem enviada que contém a chave privada de Alice. Com essa chave Bob pode começar a trocar mensagens com Alice utilizando a criptografia de chave privada ou também chamada de chave única.

Os algoritmos simétricos mais conhecidos são: DES, DES Triplo, RC4, RC5, Blowfish e IDEA. Sendo que o DES foi adotado como padrão desde 1977, e foi utilizado em milhares de produtos diferentes para diversos tipos de aplicações. O DES foi de domínio público mesmo antes de ser adotado como padrão. O fato de ser domínio público não afeta em nada a segurança, pois cada grupo de usuários escolhe a sua chave secreta. O AES (*Advanced Encryption Standard*) se tornará o algoritmo padrão de codificação do governo dos Estados Unidos. (Schneier, 2001).

5.2.2 Algoritmos de Chave Pública

Segundo Schneier (1996), os algoritmos de chave pública são conhecidos como algoritmos assimétricos. Esses algoritmos são feitos para utilizar uma chave criptográfica diferente da chave descriptográfica. Além disso, a chave criptográfica não pode ser calculada a partir da chave descriptográfica . Os algoritmos são chamados de chave pública porque a chave criptográfica pode ser pública. Uma pessoa estranha pode utilizar a chave criptográfica para criptografar uma mensagem, mas apenas uma pessoa que tem a chave descriptográfica pode descriptografar a mensagem. Nesses sistemas a chave criptográfica é chamada de chave pública (*public key*) e a chave descriptográfica é chamada de chave privada (*private key*). As chaves privadas também são chamadas de chaves secretas.

A criptografia utilizando à chave pública pode ser representada por:

$E_K(M)=C$

Como a chave pública e a chave privada são diferentes, a descriptografia utilizando uma chave privada correspondente a chave pública é representado por:

$D_K(C)=M$

Algumas vezes as mensagens podem ser criptografadas utilizando a chave privada e descriptografadas com a chave pública, sendo usado na assinatura digital.

Um exemplo de comunicação utilizando criptografia com chave pública: Alice e Bob querem trocar mensagens, utilizando a criptografia de chave pública. Cada um deles escolhe uma chave pública e uma chave privada ou secreta. As chaves de Alice serão denominadas de P_{alice} e S_{alice}, para especificar a chave pública e a chave privada de Alice respectivamente. E as chaves de Bob serão denominadas P_{bob} e S_{bob}, para especificar a chave pública e a chave privada de Bob respectivamente.

Cada um deles envia sua chave pública para um *chat* e guarda bem sua chave privada em seu PC. Alice tem uma mensagem, denominada M, para enviar para Bob. Ela, com a chave pública de Bob, encripta a mensagem com essa chave. Simbolicamente fica assim: P_{bob} (M). Então, ela envia a mensagem para Bob. As chaves de Bob e Alice não foram escolhidas arbitrariamente, elas possuem uma propriedade especial que traduz uma mensagem com uma chave e o resultado com outra chave e obtêm a mensagem original. Simbolicamente é: S_{alice} (P_{alice} (M))= P_{alice} (S_{alice} (M))=M. Não há outra maneira de obter a mensagem original senão com a chave privada.

No exemplo, Bob descriptografa a mensagem enviada por Alice com sua chave privada, que ele sabe que é $P_{bob}(M).S_{bob}$ (P_{bob} (M))=M, assim ele pode ler a mensagem original enviada por Alice.

Bob sabe que ninguém poderia ter lido sua mensagem, pois somente ele tem a chave privada de Bob. Mas ele não sabe se a mensagem foi enviada realmente por Alice, porque qualquer pessoa que estivesse no *chat* e tivesse lido a chave pública, poderia ter enviado mensagem com o nome de Alice.

Alice resolve enviar mais uma mensagem para Bob, só que ela não quer que ele fique em dúvida se quem enviou a mensagem foi ela ou se foi outra pessoa. Então, ela criptografa sua mensagem com sua chave privada e anexa o resultado depois da mensagem original, ficando como uma assinatura: M + S_{alice} (M). Alice envia a mensagem para Bob que recebe a mensagem e descriptografa a assinatura com a chave pública de Alice: $P_{alice}(S_{alice}$ (M))=M, então ele lê a mensagem.

Dois problemas graves nesse exemplo acima. Primeiro, se alguém intercepta a mensagem no canal de comunicação, ele pode ler a mensagem tranqüilamente. Segundo problema, sem encontrar Alice, Bob não pode ter certeza que a chave pública que ele recebeu é realmente de Alice. E se fosse outra pessoa que tivesse enviado a chave pública dela com o nome dela? Esse é um problema grave na utilização de criptografia de chave pública, pois uma pessoa pode gerar uma chave assimétrica (pública) se passando, por exemplo, pelo presidente Fernando Henrique Cardoso e enviar essa chave pública para diversas pessoas.

No exemplo, para resolver o problema, existe uma pessoa que todos confiam chamada Tom. Tom seleciona um conjunto de chaves e se oferece para assinar documentos com sua chave privada caso o dono do documento prove sua identidade. Então Alice faz com que Tom assine sua chave pública e então ela envia a chave assinada, chamada de certificado, no *chat* . Bob verifica se a assinatura da chave que encontrou no *chat* com a chave pública de Tom. Então ele sabe que Tom assinou essa mensagem e Tom deve ter certificado a identificação de Alice, sendo assim a chave publicada no site deve pertencer a Alice (Schneier, 2001).

Segundo Schneier (1996), as técnicas para fazer a utilização da criptografia com chave pública utilizando certificação foi implementada por Ronald L. Rivest, Adi Shamir e Leonard M. Adleman, sendo patenteada pelos mesmos sob o nome de RSA. O sistema RSA é utilizado por praticamente todos os esquemas de criptografia forte utilizadas na Internet, e para isso era preciso pagar uma taxa de licença de uso em todos os aplicativos que utilizam o RSA. Só que a patente do RSA expirou no dia 20 de setembro de 2000, e esse algoritmo deve tornar-se de domínio público, desde que o governo dos Estados Unido conceda a permissão.

5.2.3 Cripto Análise

Segundo Schneier (1996), o principal objetivo da criptografia é garantir que a mensagem (texto normal) ou uma chave fique secreta para um bisbilhoteiro (que pode ser um inimigo, um concorrente, um intruso, um *hacker*, etc.). Os bisbilhoteiros têm acesso completo no meio de comunicação entre o remetente e o receptor da mensagem.

A cripto análise é a ciência que estuda como recuperar uma mensagem para o formato de texto normal sem possuir a chave de descriptografia. Sendo que uma critpo análise bem feita pode recuperar a mensagem ou até mesmo uma chave. Um ataque é denominado quando uma tentativa de cripto análise é executada.

Alguns tipos de ataques mais utilizados são (Schneier, 1996):

- Ataque da mensagem criptografada: o bisbilhoteiro somente conhece a mensagem criptografada. O trabalho aqui é tentar recuperar a mensagem de texto normal através das possibilidades ou tentar deduzir a chave através das mensagens criptografada, para descriptografar outras mensagens criptografadas com as mesmas chaves.
- Ataque da mensagem normal: o bisbilhoteiro não tem acesso somente à mensagem criptografada, mas também tem acesso à mensagem de texto normal. O trabalho é tentar deduzir a chave utilizada para criptografar as mensagens ou um algoritmo para descriptografar uma nova mensagem que foi criptografada com a mesma chave.
- Ataque da mensagem criptografada escolhida: podem ser escolhidas diferentes mensagens criptografadas para fazer a descriptografia e ainda ter acesso a mensagens de texto normal que já foram descriptografadas. O trabalho aqui é deduzir a chave.

- Ataque da mensagem normal escolhida: o bisbilhoteiro não tem acesso somente a mensagem criptografada, mas também pode escolher à mensagem de texto normal que foi criptografada.

5.2.4 Funções de *Hash* Unidirecionais

Segundo Schneier (1996), as funções de *hash* unidirecionais são como impressões digitais, sendo pequenos pedaços de dados que podem servir para identificar objetos digitais muito maiores.

As funções *hash* unidirecionais são algoritmos públicos e não existe nenhuma chave secreta envolvida. Elas se chamam unidirecionais devido à sua natureza matemática, sendo que qualquer um pode calcular o *hash* unidirecional de um objeto digital, por exemplo um texto de um livro, porém com o *hash* desse livro é inviável computacionalmente criar outro livro que tenha o mesmo valor.

Segundo Schneier (2001), as funções de *hash* são muito utilizadas na criptografia e segurança de computador, e quase todo o protocolo da Internet as utiliza para encadear uma seqüência de eventos ou processar chave. Sendo que elas são essências para a assinatura digital. As funções de *hash* unidirecionais utilizadas atualmente são o MD5 que está sendo cada vez menos utilizado, o RIPEMD-160 que é um algoritmo europeu, e o SHA-1 que é a função padrão do governo dos Estados Unidos.

5.2.5 Geradores de Números Aleatórios

Os números aleatórios são utilizados por quase todos os sistemas de segurança de computador, porque a criptografia precisa de números aleatórios para gerar chaves, valores exclusivos em protocolos, etc. Se o sistema não garante a segurança na geração dos números aleatórios, o sistema inteiro cai (Schneier, 2001).

Segundo Schneier (2001), é impossível conseguir um número realmente aleatório a partir de uma máquina determinística com um computador, mas isso pode ser contornado, pois o que é preciso de um gerador de números aleatórios não é que os números sejam aleatórios de verdade, e sim que eles sejam irreprodutíveis e imprevisíveis, garantindo assim a segurança.

5.2.6 Códigos de Autenticação de Mensagens

Segundo Schneier (2001), os códigos de autenticação de mensagens (MACs) garantem a autenticação e a integridade, garantindo que a mensagem veio da pessoa da qual ela afirma ter vindo (autenticação) e que a mensagem não foi modificada no caminho (integridade). Mas um MAC não garante a privacidade, ou seja, qualquer um pode ler a mensagem, mas somente quem tem a chave MAC pode verificar se a mensagem foi alterada.

O MAC é um número anexado a uma mensagem digital. Eles utilizam uma chave privada compartilhada, como nos algoritmos simétricos. Um exemplo (Schneier, 2001): Alice compartilha uma chave com Bob, e quando ela deseja enviar uma mensagem para Bob, ela calcula o MAC da mensagem e o anexo junto a mensagem. E quando Bob recebe a mensagem, ele calcula o MAC com a chave compartilhada por Alice e compara com a chave MAC enviada por Alice. Se os MACs combinarem, está garantido que a mensagem foi

enviada por Alice, pois somente ela tem a chave privada para gerar o MAC e que a mensagem não foi alterada, pois o MAC só pode ser calculado através da mensagem completa e sem alterações.

Os MACs são muito utilizados na Internet, como por exemplo o protocolo *Ipsec*, que utiliza MACs para garantir que os pacotes IP não foram alterados entre o momento que eles são enviados e o momento em que chegam no destino. Os MACs são construídos a partir de funções *hash* unidirecionais ou algoritmos simétricos.

5.2.7 Assinatura Digital

A assinatura digital oferece um nível de autenticação para as mensagens. Ela utiliza um par de chaves, a chave pública e a chave privada, como ocorre na criptografia por chave pública. Também não se consegue derivar uma chave a partir da outra.

Segundo Schneier (2001), um exemplo seria: Alice tem uma mensagem em texto claro, e ela utiliza sua chave privada para codificar a mensagem. Como a chave privada de Alice pertence somente a ela, somente ela pode codificar a mensagem desse jeito. Então a mensagem codificada passa a ser a assinatura de Alice na mensagem. Com a chave pública de Alice, que todos podem ter acesso, qualquer um pode descriptografar a mensagem, verificando se Alice assinou a mensagem. Como a assinatura é uma função da mensagem, um falsificador não pode utilizar essa assinatura em outra mensagem. E a assinatura é uma função da chave privada de Alice, sendo o uso exclusivo de Alice.

Porém, os sistemas que existem são mais complicados. Alice não assina mensagens diretamente, assim como ela não codifica as mensagens com algoritmos de criptografia por chave pública (ela codifica uma chave da mensagem). Alice pega o *hash* unidirecional de uma mensagem e depois assina o *hash*, porque assinar o *hash* é muito mais rápido do que assinar a mensagem, e existem problemas de segurança com a assinatura diretamente de mensagens (Schneier, 2001).

Segundo Schneier (2001), os algoritmos de assinatura digital, na maioria dos casos, não codificam as mensagens que são assinadas. Alice faz um cálculo baseado na mensagem e em sua chave privada para gerar a assinatura, e essa assinatura vai anexada junto com a mensagem. Quando Bob recebe a mensagem, ele faz um cálculo baseado na mensagem, na assinatura e na chave pública de Alice para verificar a assinatura da mensagem. Mary, que não conhece a chave privada de Alice, pode verificar a assinatura, mas não pode falsificar a mensagem ou falsificar a assinatura de Alice em outra mensagem.

Existem vários algoritmos de assinatura digital, sendo que o mais utilizado é o *RSA*. Também o *DAS (Digital Signature Algorithm)* é muito utilizado pelo governo dos Estados Unidos. Outros algoritmos como o *ElGamal* podem ser encontrados em uso.

5.2.8 Protocolos

As ferramentas já discutidas nos artigos anteriores como criptografia simétrica, códigos de autenticação de mensagens, funções de *hash* unidirecionais, assinaturas digitais, geradores de números aleatórios e a criptografia por chave pública, são as ferramentas utilizadas pelos criptógrafos para montar soluções criptográficas para problemas reais. Segundo Schneier (1996), esses seis primitivos básicos são o núcleo de um protocolo criptográfico. Um exemplo, Alice deseja armazenar alguns dados, um protocolo que faz seria:

Alice escolhe uma frase de senha, e o programa de criptografia desmonta essa frase para obter uma chave secreta e depois usa um algoritmo por chave privada para criptografar o arquivo com os dados. Como resultado, o arquivo fica criptografado e somente quem possuir a senha tem acesso a esse arquivo.

Segundo Schneier (1996), um protocolo é basicamente uma série de etapas predeterminadas, que são completadas por duas ou mais pessoas, projetadas para realizar uma tarefa. Todos os envolvidos no protocolo precisam saber as etapas que devem ser seguidas. Um exemplo no mundo real seria o protocolo utilizado entre um comerciante e um cliente para comercializar uma dúzia de ovos. As etapas seriam estas:

- O cliente pede uma dúzia de ovos para o comerciante;
- O comerciante entrega a dúzia de ovos para o cliente;
- O cliente entrega o dinheiro para o comerciante;
- O comerciante registra a operação e entre o troco para o cliente.

Existe uma dependência entre as etapas, ou seja, se a etapa dois não completada com sucesso (o comerciante não entregou a dúzia de ovos para o cliente), então a etapa três não ocorrerá. Os protocolos utilizados no mundo digital têm que ser seguros, pois assim como no mundo real o cliente pode roubar a dúzia de ovos, o comerciante pode não entregar o troco para o cliente, etc., no mundo digital o anonimato das pessoas aumentam em um muito os riscos de fraude.

Os protocolos digitais utilizam a criptografia para garantir as mesmas coisas que acontecem no mundo no real: autenticar coisas, oferecer auditoria, manter segredos, etc. Na Internet, existem diversos tipos de protocolos de segurança. Outras redes também utilizam protocolos de segurança, como as redes de TV por assinatura, às redes de telefones celulares, entre outras.

Os protocolos digitais podem ser utilizados para diversos objetivos, como garantir a segurança em uma votação eletrônica via Internet, utilizar dinheiro digital, dar suporte a um sistema de assinatura de contratos simultâneos pela Internet, de modo que nenhuma parte envolvida esteja amarrada ao contrato a menos que a outra parte esteja, e muitas outras formas de utilização.

Segundo Schneier (2001), a criptografia na Internet é relativamente nova e foi necessária por causa do comércio eletrônico, e como a Internet por sua própria estrutura é totalmente desprotegida, a criptografia torna-se fundamental para proteger a Internet.

Os algoritmos mais utilizados no email, onde a criptografia utilizada primeiramente na Internet, é os *OpenPGP* e o *S/MIME*. A Netscape inventou logo no início da Web o *SSL*, para garantir a segurança nas transações de comércio eletrônico utilizando o seu *browser*. O *SSL* passou a se chamar *TLS* por causa da briga entre a Microsoft e Netscape pelo mercado de *browser*, sendo utilizado por todos os *browsers* para codificar as informações confidencias enviadas pela Internet. Muitos outros protocolos foram desenvolvidos para proteger os pacotes da rede IP, sendo que alguns protocolos, como o PPTP (*Microsoft Point-to-Point Tunneling Protocol*), possuem problemas.

5.3 SEGURANÇA NO *WAP*

A segurança no WAP foi implementada pelo WAP Fórum baseado no modelo que já existia na Internet convencional, usando assim as tecnologias já maduras na área de segurança. A possibilidade de trabalhar com a Internet e seus protocolos é um dos objetivos

mais importante no WAP, e também a implementação da segurança, que tem grande parte implementada diretamente na camada de transporte (TLS) na Web. No WAP, a implementação de segurança é feita na camada de *WTLS*, que é baseada na *TLS*, como já visto no capítulo sobre o protocolo *WAP*.

Na Europa, o padrão de telefonia móvel utilizado é o GSM. No GSM todos os dados entre o telefone móvel e a estação-base da empresa telefônica são criptografados utilizando-se o algoritmo A5 (Dornan, 2001) . Segundo Schneier (1996), esse algoritmo, que foi concebido na Europa, não poderia ser exportado para ser utilizados em sistemas de telefonia móvel em outros países que não eram da OTAN ou aliados, por causa do medo da utilização por terroristas. O A5 é um algoritmo de criptografia de fluxo, que consiste de três LFSRs. Os tamanhos dos registradores são 19, 22 e 23 e todos os polinômios de retorno são escassos, sendo que a saída é uma operação XOR dos três LFSR's. O A5 utiliza variáveis de controle de tempo. Sendo um algoritmo baseado em boas idéias e é muito eficiente (Schneier, 1996).

Segundo Arehart (2000), o funcionamento do modelo de segurança do WAP é representado como seguinte exemplo: uma conexão é feita pelo telefone móvel, que conecta com o provedor de acesso, normalmente a própria empresa de telefonia. Então, o telefone móvel chama e quando a operadora o recebe, ele é roteado para um modem e conectado em um servidor RAS (*Remote Access System*), exatamente como ocorre na Internet. O WAP Fórum recomenda que o dispositivo móvel utilize o protocolo PPP na comunicação com a estação base, mesmo que a portadora da rede for GSM que possui um nível bom de criptografia nessa comunicação. O servidor RAS é responsável pela autenticação, como ocorre na Internet, mas nesse ponto existe uma diferença, pois os pacotes que passam pelo servidor RAS são roteados para o *Gateway WAP*, e não são enviados diretamente para o servidor Web como ocorre na Internet. O *Gateway WAP* é o responsável pela conversão do código WML e *WMLScript* para binário, para ser enviado via wireless, além de fazer o papel de *Proxy* do telefone móvel, comunicando com o Web server sempre que o telefone desejar, utilizando o protocolo HTTP 1.1.

Normalmente, a operadora tem um servidor de Web dentro de sua rede, fazendo com que os pacotes nunca deixem a rede da própria operadora. Mas o cliente móvel pode desejar acessar o conteúdo que esteja em outra rede; nesse caso, o *Gateway WAP* envia os pacotes HTTP através do *Firewall* para o servidor de conteúdo de outra organização. A operadora de rede pode ter uma *DMZ (Demilitarized Zone)* que é uma rede sem um *Firewall* para garantir a segurança, no lugar onde o *Web Server* oferece serviços Internet e Wap para o público em geral.

Como o *Gateway WAP* utiliza o protocolo HTTP 1.1 para enviar os pacotes para o servidor destino, a utilização do TLS é necessária para garantir a segurança nessa comunicação. Mas o TLS não pode ser utilizado na comunicação entre o dispositivo móvel e o *Gateway WAP*, pois o TLS trabalha somente sobre um meio de transporte seguro, normalmente o TCP, mas o telefone não pode utilizar o TCP para se comunicar com o *Gateway*. Para resolver esse problema, o WAP Fórum criou um novo protocolo de segurança, baseado no TLS, e prove um nível de segurança semelhante, chamado WTLS que trabalha com UDP sob redes IP, ou trabalha com WDP sob redes não-IP.

Fica claro que os modelos de segurança Internet e WAP são semelhantes, mas existe uma diferença fundamental, a presença do *Gateway WAP* no modelo WAP. Outro fator é garantir a autenticação no modelo WAP. É difícil garantir a autenticação no modelo WAP, pois muitas aplicações guardam informações de segurança como o ID e a senha do usuário no próprio celular. Até mesmo o uso de certificado não é tão óbvio, porque o certificado é

utilizado na maioria das vezes para identificar o dispositivo móvel, e no caso do usuário perder o aparelho ou ser roubado, outra pessoa pode se passar por ela, e utilizar o certificado para realizar transações. Uma das soluções utilizadas nesse caso seria escrever aplicações que nunca guardem informações no dispositivo móvel, mas isso pode chatear o usuário, porque ele teria que entrar com os dados sempre que acessar o sistema. Outra solução seria utilizar um certificado no aparelho e mesmo assim utilizar a autenticação com um ID e senha.

Segundo Dorman (2001), um dos maiores problemas de segurança no modelo WAP está no *Gateway WAP*. Como o *Gateway WAP* é responsável pela conversão do WML e *WMLScript* em binário, isso implica os seguintes problemas:

- A conexão entre o dispositivo móvel e o Gateway WAP é garantida pelo WTLS, mas quando o Gateway tem que fazer a descriptação dos dados para fazer a decodificação e depois realizar novamente a encriptação . Ou seja, os dados ficam à disposição do administrador do Gateway WAP ou de pessoas que invadam esse sistema.
- O Gateway WAP analisa todos os dados no formato de texto normal (*cleartext*).

Mesmo sendo muito difícil de ocorrer interceptação de dados, pois o *Gateway* não escreve os dados em disco, ele faz a descriptação e encriptação totalmente na memória, que é logo reescrita. Porém, ainda existe o perigo, pois a rede da operadora pode oferecer riscos de segurança que não são de conhecimento da empresa cliente. A solução indicada nesse caso é garantir segurança total na rede da organização, e utilizar um *Gateway WAP* próprio, garantindo assim que as informações descriptografadas não sejam acessíveis para pessoas não autorizadas. Claro que isso eleva o custo do projeto, mas garante a segurança na transmissão das informações.

5.4 CONCLUSÃO SOBRE SEGURANÇA

A tecnologia de segurança WAP, mesmo que baseada em outras mais maduras, é passível de falhas de segurança, o que pode causar sérios danos a uma empresa, principalmente no comércio eletrônico, onde uma notícia de invasão pode simplesmente arruinar uma empresa. Para aplicações de *M-commerce* ou aplicações bancárias onde a segurança é fundamental, até que seja desenvolvido um sistema de criptografia de uma ponta a outra na comunicação WAP, a única maneira de garantir total segurança é ter a propriedade do *Gateway Wap.*

Atualmente na Europa, segundo Dornan (2001), muitos bancos tem o *Gateway WAP* próprios, e os seus clientes discam para um número do banco onde então ele é capaz de acessar a Internet. Como na Europa, a utilização de telefone móvel para fazer transações bancárias é muito grande, os bancos preferem investir em infra-estrutura própria e garantir a segurança referente ao *Gateway WAP* em vez de confiar a segurança de seus sistemas móveis para uma operadora telefônica. O problema é que o custo de acesso à Internet Móvel é normalmente cobrado por minutos acessados. Sendo que o serviço quando discado para a operadora telefônica é mais barato do que uma ligação normal de celular.

Desde o surgimento da Internet Móvel, as empresas, que produzem software antivírus, avisam sobre a possibilidade de ocorrem ataques de vírus a telefones celulares. Em Julho de 2000, a mídia informou que telefones celulares tinham sido atacados por um vírus chamado Timofônica. Mas na verdade, o Timofônica é um vírus de computador que envia mensagens SMS para telefones celulares e não um vírus de celular. Porém, com o aumento de capacidade

de processamento e memória na próxima geração de telefones celulares, existe grande possibilidade de vírus a serem criados para dispositivos móveis (Dornan, 2001).

Outro aspecto que é muito importante e defendido por pessoas consagradas no mundo da segurança, defende que os padrões devem ser abertos e públicos quando trata se de redes digitais, principalmente em relação à segurança. “Escolher um sistema proprietário é como ir a um médico sem graduação em Medicina, e cujo tratamento (que ele se recusa a explicar) não é reconhecido pela *American Medical Association*” (Schneier, 2001). Ou seja, Schneier pensa que somente a comunidade científica pode criar métodos e padrões de segurança, seja em um software, um sistema num sentido geral, ou um algoritmo criptográfico.

As pessoas, que compartilham das idéias de software de criptografia publicado livremente, como Schneier (2001), baseam-se na idéia que um bom “design” de sistema não precisa de segredos de seus detalhes, e ao contrário, um bom “design” de sistema é seguro mesmo se seus detalhes são públicos. Argumentar que uma solução secreta ou proprietária é por isso mais forte, é uma ilusão de segurança. Como Schneier (2001) definiu, segurança é um processo e não um produto. Por isso, as empresas que prometem vender segurança em um único produto proprietário estão enganando o cliente. Principalmente porque nesse processo está o fator humano que muitas vezes pode ser algo independente de tecnologia.

6 PROTÓTIPO DE *M-COMMERCE*

O protótipo deste TC baseia-se na idéia de uma cooperativa on-line, onde as pessoas podem realizar compras em grupo, para baixar o preço de um produto em até 40%, garantindo, assim, um bom lucro para o fornecedor por causa do volume, uma grande redução para os compradores, pois eles teriam que comprar o produto pelo preço de mercado, e garante uma margem de lucro boa para o *site* que faz a intermediação do negócio, no caso o *site E-Móvel* que é o nome desse projeto. O projeto foi baseado em *sites* de cooperativas, um novo tipo de negócio que foi inventado em 2000 para intermediar vendas em grupo na Internet. Alguns casos dessa implementação na Web são os sites ComDesconto (www.comdesconto.com.br) e Agrupate (www.agrupate.com.br).

O ComDesconto é um site do Grupo CoShopper.com que mantém sites de *e-commerce* em diversos países como o Japão, Portugal, Inglaterra, França, Espanha, entre outros. O site foi desenvolvido utilizando a linguagem PHP versão 3. Ele trabalha com o conceito de grupo de compras exatamente como o E-Móvel, só que no ComDesconto o cliente faz um pedido e espera o grupo de compras fechar, para então receber o produto; mas no E-Móvel, quando o cliente entra em um grupo de compras, o pedido é faturado automaticamente e o produto seria entregue imediatamente.

6.1 FUNCIONALIDADES DO E-MÓVEL

O cliente pode acessar o sistema E-Móvel como visitante, onde ele somente pode visualizar as ofertas oferecidas, ou pesquisar por um determinado produto, mas ele não pode realizar as operações de compra. Se o cliente desejar comprar um determinado produto, ele deve se cadastrar no E-Móvel. Ele pode se cadastrar através do dispositivo móvel, ou através do site Web do E-Móvel, facilitando o cadastro. Os dados pessoais que devem ser informados: o nome completo, cep, número da casa ou apartamento, cpf, telefone, data de nascimento, e-mail, nome de usuário e senha para acessar o sistema.

Em princípio, o cliente pode comprar os produtos de qualquer um dos grupos de compra disponíveis no site, desde que esse grupo não esteja completo. Quando o grupo estiver completo, o sistema deverá retirar o *link* para esse grupo automaticamente.

Mas o que é um grupo de compra? É um sistema de comércio eletrônico onde o número de participantes determina descontos no preço de um produto. Quanto mais participantes e pedidos de compra, menor o preço. O grupo de compra é formado por pessoas físicas que querem comprar um mesmo produto. Quando determinado número de pedidos é atingido, todos os participantes obtêm um desconto e assim se segue progressivamente até o desconto máximo. Todo o grupo de compra tem data marcada para encerramento. Quando a

data é atingida, o grupo se encerra. Outra característica importante é que o grupo de compra tem limite de produtos para venda naquele período, indicado na página do produto. Uma vez atingido este limite, o grupo se encerra.

O foco do sistema será atender um público mais seleto, que goste de cultura, viagens, e compras de produtos diferenciados em lojas de eletrodomésticos e supermercados. As vantagens de utilizar o sistema *WAP* seriam, o acesso móvel de qualquer lugar e a qualquer hora, os serviços e produtos oferecidos seriam com um preço bem abaixo do preço de mercado, principalmente nas viagens, hospedagens, nas entradas de cinema, teatro e shows, pois seriam oferecidos bilhetes daquele dia, por exemplo, um show no teatro que iria acontecer em um determinado dia, e faltam vender 50 bilhetes. Esses bilhetes serão vendidos no sistema de cooperativa, com um valor que pode ser até 40% mais barato que o valor de mercado, dependendo da quantidade de participantes dos grupos, garantindo assim o lucro do teatro e também que os espectadores paguem bem menos pelo bilhete. Além de bilhetes, outros produtos que podem ser vendidos como oferta relâmpago como *CDs*, *DVDs* e livros, eletrodomésticos, hardware, software e ofertas em redes de supermercados.

Quando o cliente realiza a compra, é gerada uma nota fiscal e o sistema E-Móvel envia um email para o cliente confirmando a operação com os dados da compra.

Os clientes também poderão dar sugestões de novos grupos ou sugestões sobre o site. Os clientes poderão comprar um produto pagando o valor do momento. Se mais clientes entrarem no grupo desse produto, o valor pago a mais será devolvido no cartão de crédito do cliente.

O cliente poderá pesquisar por qualquer produto que ele desejar, ou realizar uma pesquisa somente produtos que estarão em oferta relâmpago naquele instante de tempo.

Um diferencial do E-Móvel é que o cliente não tem nenhum custo pelo frete do produto, pois o E-Móvel realiza acordos comercias com transportadoras para viabilizar a entrega o mais rápido possível, com um custo baixo para o site. O custo da entrega dos produtos é diluído no valor dos grupos. Essa estratégia foi utilizada pelo ComDesconto no início das operações no Brasil para atrair clientes. Porém, depois de alguns meses, os diretores do site decidiram cobrar novamente pelo frete.

Outro diferencial em relação aos outros sites de venda em cooperativa, é que o E-Móvel processa o pedido no momento em que o cliente participa do grupo de compra, e não quando o grupo se encerra pelo limite de participantes ou porque foi atingida a data de fechamento do grupo.

Também é oferecido aos clientes, é a possibilidade de enviar um e-mail através do site E-Móvel. O cliente deve ser cadastrado no site para ter direito a esse serviço que também não tem custos para o cliente. O tamanho máximo para a mensagem é de 150 *bytes.*

O cliente na hora do cadastro poderá solicitar o recebimento de informações das ofertas relâmpagos via e-mail no seu celular, podendo assim fazer a compra de um determinado produto ou serviço, pois a maioria das ofertas do sistema vai ser rápida, aproveitando também a disponibilidade do sistema móvel. A forma de pagamento será somente cartão de crédito, pois quando o cliente tem direito à devolução de dinheiro, o valor será credito no cartão do cliente.

6.2 ARQUITETURA DO E-MÓVEL

O sistema E-Móvel utilizará as tecnologias utilizadas que serão o Sistema Operacional alemão SUSE Linux 7.1 no servidor de aplicação que também rodará o Servidor *Web Apache*, utilizando as linguagens *PHP*, *WML* no desenvolvimento. O servidor que conterá todos os dados da aplicação será um servidor Linux rodando o banco de dados MySQL, pois o projeto é implementar um protótipo de *M-commerce* com ferramentas *open source*. Futuramente, se for necessário alterar o SGBD, torna-se muito fácil, sendo necessário alteração mínima no código. O acesso para os clientes será feito via telefone celular ou por um emulador disponível na Internet, sendo possível utilizar o E-Móvel tanto com um dispositivo móvel, como também com um computador pessoal. Na Figura 16, está a arquitetura do site E-Móvel.

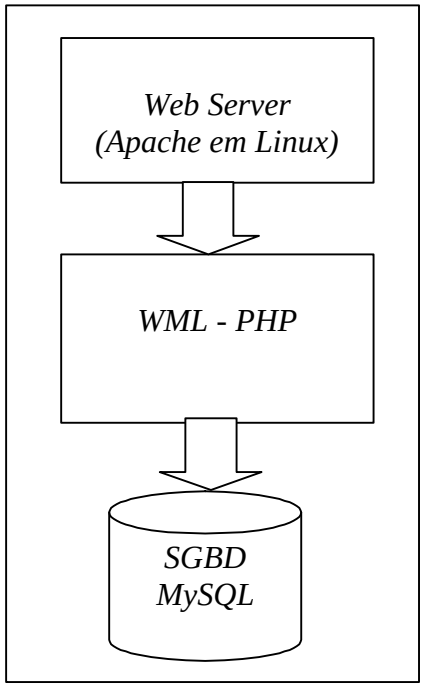


Figura 16 – Arquitetura do Site E-Móvel

Como linguagem de marcação, foi utilizado o WML que é a linguagem para criar as páginas que são visualizadas pelos *microbrowsers WAP* dos dispositivos móveis. É uma linguagem relativamente fácil de aprender, tendo algumas *tags* idênticas ao HTML. Mas o que torna complexo o desenvolvimento com *WML* não é a sintaxe, e sim as restrições de recursos impostas pela tecnologia atual dos dispositivos móveis, como a limitação de teclado, de tela, de banda de comunicação e falta de elementos gráficos. Isso torna a criação da interface com o usuário muito complicada, e a lógica de criação do site difícil e muito trabalhosa.

A utilização do *SUSE Linux* foi motivada por causa da grande quantidade de programas livres que vêm com essa distribuição do Linux, sendo umas das distribuições mais utilizadas, principalmente na Europa. Entre esses programas, destaca-se o Apache, PHP, MySQL, Java, *OpenSSL*, biblioteca de criptografia *Mcrypt*, e outros.

O servidor *HTTP Web* utilizado no protótipo foi o servidor Apache, que é *open source*, roda em multiplataforma, tem ótimo desempenho no atendimento de requisições da Internet e tem muitas funcionalidades incorporadas com a utilização de módulos de software, entre os

mais importantes o módulo *Open SSL* para suportar transações seguras. Foi escolhido também por ser o servidor Web mais utilizado na Internet, sendo que 62% dos servidores Web na Internet utilizam o Apache, e somente 20% utilizam o IIS da Microsoft, segundo Converse (2001).

A segurança do site foi baseada nas tecnologias mais utilizadas, ou seja, o SSL rodando no servidor HTTP Apache, e o WTLS rodando no servidor WAP *Gateway*, sendo necessário configurar o *microbrowser* do dispositivo móvel para utilizar a segurança *WTLS* na comunicação. Para acessar o site E-Móvel em modo seguro, o acesso as URLS começará com *HTTPS* em vez de *HTTP*. Isso evita ataque de curiosos ou por repetição do *Gateway WAP* para o aplicativo no servidor Web (isso se for possível configurar o servidor Web da Ulbra para aceitar *SSL*). Outra medida em relação à segurança seria instalar o banco de dados em um servidor separado do servidor Apache. Sendo colocado um *Firewall* entre o servidor Web e o servidor de banco dados, garantindo o acesso aos dados do sistema somente ao servidor Apache.

Em relação ao problema na conversão dos dados da rede móvel para a rede IP no *Gateway WAP*, como o E-Móvel é protótipo de *M-commerce*, assumimos que o servidor *Gateway* está seguro na companhia telefônica. Se fosse uma implementação comercial, o indicado é a instalação de um *Gateway* próprio para garantir a segurança.

Para garantir um nível aceitável de segurança do *Gateway*, siga os procedimentos padrão de segurança (FORTA, 2001) para o *Gateway WAP*:

- utilizar um *firewall*;
- limitar o acesso administrativo ao equipamento somente às pessoas competentes;
- limitar o acesso físico ao servidor;
- utilizar um software de *Gateway* que não armazene as mensagens de texto simples; monitorar o servidor quanto à criação de novos.

Na aplicação, o modo de autenticação de usuário foi à autenticação manual através da criação de uma tabela no banco de dados para controle de acesso de usuários ao site. Esse método é bastante simples, e se baseia na entrada do nome de usuário e senha e a verificação numa tabela de um banco de dados ou em arquivo do servidor. Para garantir que a senha dos usuários não seja visualizada, foi utilizada a função *mcrypt_cbc* disponível na biblioteca *mcrypt.* Com essa função, é possível utilizar diversos algoritmos da chave privada. Alguns desses algoritmos são o *DES*, o *DES Triplo*, o *Twofish* e três implementações do *Rijndael* (chaves de 128, 192 e 256 *bits*). Foi utilizado o algoritmo *DES Triplo* e o modo de cifra CBC *(cipher block chaining)* , por ser um algoritmo muito poderoso e seguro, segundo Schneier (1996).

O problema da utilização dessa biblioteca é que ela deve ser compilada juntamente com a compilação do PHP e o servidor Apache, o que torna complicado para realizar em um servidor que está em produção. E também ela roda somente em ambiente UNIX. Mas com certeza é uma ótima biblioteca para ser testada, utilizada e para pesquisas, pois todos os códigos fontes estão completos em linguagem C.

Também poderia ser a autenticação básica incorporada no servidor Web ou também o método de autenticação que obtém o ID do telefone celular utilizando a variável *HTTP_X_UP_SUBNO*. A autenticação realizada pelo servidor *Web* é uma alternativa a autenticação manual por uma tabela de banco de dados. Esse recurso é chamado de Autenticação Básica (*Authentication Basic*) e foi implementado no *HTTP* através da RFC 2617. O servidor e o navegador implementam um esquema de autenticação básica abaixo da

camada de aplicativo. O servidor *Web* gera um código de resposta para cada solicitação *HTTP* recebida. Normalmente, o código de resposta é "*OK*", mas quando a solicitação se refere a uma página protegida, o código de resposta orienta o navegador a solicitar a identidade do usuário. Essa solicitação de identidade orienta o navegador WML ou HTML a solicitar. Em seguida, o navegador envia informações de volta para o servidor Web e o servidor valida as informações. Se as informações forem autenticadas com sucesso, o servidor Web retorna o código "*OK*" e envia a página solicitada. Mas para implementar o recurso de autenticação, o servidor *Web* necessita de um diretório de usuários, podendo esse diretório de usuários ser somente para a aplicação ou ser corporativo. Se a autenticação for realizada com sucesso, o servidor *Web* inclui o nome e a senha do usuário para cada solicitação que o navegador realizar. Depois, o servidor *Web* cria uma variável de aplicação, para que o sistema que estiver sendo executado no servidor possa acessar essa variável. Ela se chama *REMOTE_USER*.

A outra alternativa de fazer a autenticação do usuário seria utilizar informações de identificação específicas da plataforma. O *Gateway WAP* da Phone.com envia uma variável HTTP que é exclusiva do dispositivo que fez a solicitação. O problema que essa variável que identifica o usuário, chamada de *HTTP_X_P_SUBNO*, está disponível somente para dispositivos de usuários em que o agente de usuário seja da Phone.com. Essa restrição torna o uso dessa possibilidade inviável, por causa dos diversos agentes de usuários e o sistema tem que atender a todos os usuários sem restrições por causa do software ou hardware utilizado.

6.2.1 PHP

O *PHP* foi criado no Estados Unidos em 1994 por Rasmus Lerdorf, sendo que as primeiras versões foram usadas na sua *homepage* para saber quem estava consultando o seu currículo on-line. A primeira versão, utilizada por outras pessoas, foi disponibilizada em meados de 1995, e era conhecida como *Personal Home Page Tools* (Ferramentas para *Homepages* Pessoais), mas hoje o PHP significa *Hypertext Preprocessor*. Consistia num *engine* de interpretação bem simples que entendia alguns macros especiais e alguns utilitários de uso comum nas *homepages* daquela época. O sufixo FI veio de um outro pacote escrito por Ramus, que interpretava dados de formulário HTML. Ele combinou os scripts das Ferramentas para *Homepages* Pessoais com o Interpretador de Formulário e adicionou o suporte ao *MySQL*; o *PHP/FI* estava criado. O *PHP/FI* cresceu num ritmo incrível, e as pessoas começaram a adicionar-lhe código, e hoje o número de domínios na Internet que utilizam o PHP passa de dois milhões, segundo Converse (2001).

Segundo Dias (2000), o PHP é uma linguagem de programação de *scripts* que roda no lado servidor para criar sites *WML* ou *HTML* dinâmicos. Sites dinâmicos são aqueles que retornam para o cliente uma página criada em tempo real. Utilizando a linguagem *PHP*, ocorre a interação direta do usuário com o *site*; aplicações em *PHP* são geradas com excelente performance e automaticamente pelo servidor. O usuário não vê o código *PHP*, somente o *WML* ou *HTML*; isto é muito importante quando se está lidando com senhas.

Enquanto o *ASP* da Microsoft é baseado em um sistema proprietário, o *PHP* é distribuído sobre *GNU GPL (General Public License*), ou seja, não se precisa pagar nem um centavo para usar o *PHP*, e roda em um grande número de plataformas, entre elas Windows, UNIX e LINUX.

O interpretador reconhece automaticamente *scripts PHP* delimitados da seguinte maneira: <? // código em php ?>

Para trabalhar com o *PHP* utilizando o WML, deve-se declarar o cabeçalho indicando o tipo de conteúdo usado. Exemplo:

```
<?
header (“Content-type: text/vnd.wap.wml”);
echo “<?xml version=\“1.0\” ?>”;
? >
```

Segundo Converse (2001), algumas vantagens de utilizar o PHP são:

- Linguagem de script de fácil aprendizado e com grande poder interatividade;
- Velocidade de execução excelente;
- Acesso nativo a diversos bancos de dados, entre eles: Sybase, SQL Server, Oracle, MySQL, mSQL, Informix e qualquer outro banco através de ODBC.

O código PHP fica embutido dentro das páginas WML ou HTML. Sendo que o PHP não é baseado em *tags.* Segundo Meloni (2000), o PHP é muito estável, ou seja, o servidor não precisa ser reinicializado constantemente e o software não sofre alterações e incompatibilidades radicais entre uma versão e a outra. Principalmente quando o PHP é utilizado no ambiente UNIX, rodando como um módulo do servidor HTTP Apache. Quando compilado como um módulo do Apache, o PHP fica além disso muito mais rápido, tendo desempenho melhor que os scripts CGI na maioria das funcionalidades e ficando no mesmo nível ou mais rápido que o ASP da Microsoft.

O PHP torna fácil a comunicação com outros programas e protocolos. Os protocolos mais importantes também são suportados, como o POP3, IMAP e LDAP. Na versão 4 do PHP, existe um novo suporte para o Java e arquiteturas de Objeto distribuídas (COM e CORBA), tornando possível o desenvolvimento em *n camadas*. O PHP4 tem suporte a controle de sessões utilizando *cookies* (que possui problemas de segurança e compatibilidade*)*, ou utilizando funções de sessão que são implementadas no PHP4. Existe o suporte à criptografia de dados, utilizando bibliotecas com algoritmos livres de criptografia. Também existe o suporte para trabalhar com a linguagem XML e suporte para algumas funcionalidades da OOP (*Objetct Oriented Programming*), sendo que muitas funcionalidades ainda estão em desenvolvimento nessa área.

Segundo Converse (2001), as sessões são utilizadas para monitorar o comportamento de um usuário ao longo de interações que duram mais de uma execução de script. Quando o programador apresenta um conteúdo aos usuários que depende de quais páginas anteriores eles vieram ou com quais eles tiveram interação, o código precisa armazenar ou propagar essas informações em uma maneira que identifique as informações de um usuário do outro. A implementação de sessões do PHP encapsula os problemas de utilizar variáveis ocultas, *cookies* ou passar os IDs de sessão transparentemente ao programador nos argumentos *GET/POST*.

A versão Unix do PHP fornece uma biblioteca publicada livremente que disponibiliza diversas funções que implementam a criptografia com chave privada. O nome dessa biblioteca é *mcrypt.* Para utilizar essas funções com o PHP, é necessário fazer o *download* e instalar o *mcrypt* e depois compilar o PHP com a opção de configuração *–enable-mcrypt.* Nessa biblioteca, são implementados vários algoritmos de chave privada, sendo os mais conhecidos o *DES, 3DES, Twofish e Blowfish.* A escolha pela utilização desses algoritmos deve ser feita analisando a velocidade e a força de cada algoritmo. A biblioteca *mcrypt* permite que o programador escolha também o modo de cifra a ser utilizado (Converse, 2001), conforme mostra o Quadro 1.

Quadro 1 – Modos de Cifra Disponível na Biblioteca *Mcrypt*

Modo	Descrição
ECB *(eletronic code block)*	Traduz apenas o bloco de dados fornecido. Não deve ser utilizado para criptografar textos, pois a alta freqüência de caracteres pode ser utilizada para quebrar a criptografia (Schneier, 1996).
CBC *(cipher block chaining)*	Modo mais seguro e o mais indicado para utilização (Schneier, 1996).
CFB *(cipher feedback)*	Como o ECB, o CFB é útil para blocos curtos de dados.
OFB *(output feedback)*	Semelhante ao CFB mas projetado para se comportar melhor quando encontra erros em sua entrada

O PHP suporta a integração do XML (*eXtensible Markup Language)* no desenvolvimento para Internet. Com o XML, é possível manipular e armazenar dados, distribuir dados entre organizações e entre programas aplicativos e exibir páginas XML em um *browser* (WML ou HTML) ou aplicativo utilizando folhas de estilo para definir a exibição. Mas atualmente, o XML não é utilizando para armazenar dados em banco de dados. O XML é utilizado para a troca de informações entre aplicativos e organizações diferentes. Por exemplo, um programa escrito em C realiza diversas operações com dados de um bando de dados e lhe fornece a saída desse processamento em XML, que o PHP pode transformar em WML ou HTML para exibição em um *browser* ou em um aplicativo.

Existem duas APIs para tratar documentos XML: o *Document Object Model* e a *Simple API* para XML. O PHP4 tem um módulo para cada API. O módulo SAX é o padrão, e o módulo DOMXML é opcional. O SAX é mais leve e mais fácil de aprender, mas trata XML como um fluxo contínuo de dados de *string*. Ela é utilizada para analisar sintaticamente documentos de XML, sendo uma API baseada em eventos, ou seja, o analisador de sintaxe chama as funções designadas quando reconhece um certo gatilho no fluxo de eventos. Já o DOMXML é uma API orientada a objetos, e ela não é somente um analisador de sintaxe, embora as implementações tenham um analisador de sintaxe. O DOMXML lê um arquivo XML e cria uma árvore de objetos na memória. Iniciando em um documento ou um elemento de um documento, é possível recuperar ou configurar os filhos, pais e conteúdo de texto de cada nó da árvore. O DOMXML pode consumir mais recursos e tempo quando constrói uma árvore na memória e o documento for muito grande. O W3C recomenda a utilização do *Document Object Model* como está descrito no site http://www.w3.org/DOM/ .

Segundo Converse (2001), O PHP foi criado como uma linguagem de script não orientada a objetos. As classes e objetos foram incluídos somente depois do primeiro lançamento, sendo considerado um complemento tardio por muitos programadores. O próprio PHP4 foi escrito em C e não em C++ e o seu desenvolvimento contínua sendo procedural. Mas com o crescente número de usuários de PHP e com a tendência de orientação a objetos é muito forte, o PHP tende a ser cada vez mais orientado a objetos. O PHP4 suporta a maioria das funcionalidades básicas para definir classes e objetos para instanciar uma classe. As funcionalidades que o PHP ainda não suporta estão listadas aqui: herança múltipla (o PHP4 suporta somente herança de uma única classe), suporte a interfaces, suporte a classes abstratas, suporte a sobrecarga de funções, não tem suporte a modificadores privados ou protegidos em classes PHP, pois tudo é público. O PHP4 têm módulos que suportam a conexão e a manipulação de objetos COM, DCOM e Java.

Segundo Converse (2001) o PHP é muito semelhante à linguagem de programação C. Sendo esse um importante fator na crescente utilização do PHP para o desenvolvimento em Internet, pois os programadores C não sentirão nenhuma dificuldade para trabalhar com o PHP. A sintaxe PHP herdou muita coisa da linguagem C, como listado a seguir:

- Blocos: são seqüências de comandos definidos com a utilização de chaves. Exemplo: if (3 == 2+1) { print(“OK”); }
- Comentários: igual à linguagem C. Exemplos: /* isso é um comentário no PHP */ ; // Isso é um comentário de uma linha ;
- Variáveis: no PHP as variáveis são semelhantes ao PERL. Todas as variáveis são precedidas de cifrão ($), e a atribuição é feita com o símbolo “=”. Exemplo: $pi = 3.014159 ;
- Funções de Saída: são duas funções bem simples: *echo* e *print*. Exemplo: *echo* “Teste do PHP”; *print* (“Teste do PHP”). A diferença é que o *echo* aceita mais de um argumento e o *print* somente um argumento.
- Tipos de dados: o PHP não obriga que o programador declare uma variável com o seu tipo antes de utilizá-la. O programador pode fazer a atribuição diretamente e o PHP faz então a seleção do tipo pela expressão atribuída à variável. Exemplo: $num_conta = 5535.6; $dono_conta = “Rodrigo Lessa”. Os tipos de dados suportados são: inteiros, números de precisão dupla (ponto flutuante), booleanos, *strings, arrays* e *class*.
- Estruturas de controle: são bem semelhantes às estruturas utilizadas na linguagem C. Para realizar desvio no código no PHP utiliza-se *if* (condição) { bloco } *else* { } para fazer uma estrutura condicional ou utiliza-se *switch* (expressão) { case valor1: instrução1; instução2; .... *break*; case valor2: instrução1; instução2; .... *break*; }. Também existem as estruturas para realizar *loops* que são: *while* (condição) instrução; se a instrução for verdadeira continua no laço. O comando *while* (expressão); A instrução é executada pelo menos uma vez, e sai do laço quando a expressão for falsa. O comando for (expressão inicial; verificação-de-término;expressão-fim-do-loop) instrução. Funciona exatamente como na linguagem C. Exemplo: for ($cont = 0; $cont < $limit; $cont = $cont +1) { print (“$cont”)};
- Funções: O PHP também permite a criação de funções para melhorar a codificação de um sistema, tornando o código mais legível e mais fácil de manter. Para criar uma função no PHP a sintaxe é a seguinte: *function* nome_da_função ($argumento1, $argumento2, ...) {instrução1 ; instrução2; ... ; return ($valor)}. Um detalhe importante é que, se na chamada da função o número de argumentos for diferente da definição da função não ocorre um erro; se o número de argumentos for menor, o PHP considera os parâmetros não preenchidos como variáveis não vinculadas. Se o número de argumentos for maior que a definição da função, o PHP desconsidera os argumentos passados em excesso.
- Escopo de Variáveis: as variáveis atribuídas dentro de uma função serão locais a essa função somente, a não ser que sejam declaradas como global. As variáveis locais podem ser declaradas como estáticas, significando que elas continuam com o valor entre as chamadas da função.
- Chamadas de Função: o comportamento padrão para funções definidas pelo programador é a chamada por valor, onde as funções trabalham com cópias de seus argumentos e assim não podem alterar as variáveis originais na chamada da função. Esse comportamento pode ser alterado utilizando a chamada por referência precedendo os parâmetros com o “&”, tanto na definição da função

como na chamada da função. O PHP também permite que as funções que serão chamadas sejam determinadas em tempo de execução, substituindo uma variável alfanumérica pelo nome da função definido pelo programador, permitindo que as funções sejam tratadas como dados e passadas como parâmetros para outras funções.

6.2.2 SGBD MYSQL

Segundo Anselmo (2000), o bando de dados MySQL é um DBMS *(Relational Database Management System)* chamado em português de SGBD (Sistema Gerenciador de Banco de Dados), baseado no padrão SQL92 que define algumas características que os SGBD devem possuir. Ele utiliza a linguagem SQL *(Structured Query Language)* para realizar as operações no banco de dados.

Outra característica importante é que o MySQL é *Open Source Software* sendo possível utilizar e modificar o seu código-fonte no ambiente Unix. O MySQL usa o GPL *(GNU General Public License)* para estabelecer o que pode e o que não pode ser feito com o software em diversas situações diferentes.

As principais vantagens de utilizar o MySQL são:

- Compatível com a grande maioria de sistemas operacionais que existem. Entre eles: AIX, HP-UX, Solaris, OS/2, Linux, SCO UNIX, TRU64 UNIX, FreeBSD, MacOS X, Windows 9.X, WINNT, WIN2000;
- APIs para as seguintes linguagens: C, C++, Eiffel, Java, Perl, PHP, Python e Tcl;
- *Multithread* utilizando *threads* de *kernel*;
- Um banco de dados pode ter mais de 50.000.000 de registros;
- Velocidade de execução dos comandos é excelente, sendo um dos mais rápidos SGBD existentes;
- Controle de usuários flexível e muito simples;

Como todos os programas novos, a primeira versão do MySQL foi implementada em 1995, ele ainda não suporta as seguintes funcionalidades:

- Relacionamento entre tabelas (*Foreign Key*) (está em desenvolvimento nesse momento);
- *Store procedures e triggers* (existe a previsão para a implementação);
- *View* e o comando *Select* com *Sub-Select*

O *MySQL* foi implementada usando o padrão SQL92 como base, como foi visto anteriormente. Segundo Stoco (2000), os principais comandos SQL implementados no *MySQL* são:

- *Alter Table*: que modifica os atributos de uma coluna ou adiciona um novo campo em uma determinada tabela em banco de dados. As ações desse comando são: *add column* para adicionar uma coluna; *add index* para adicionar um índice à tabela; *add primary key* para adicionar uma chave primária à tabela; *add unique* para adicionar um índice sem duplicidade à tabela; *alter column* para setar ou eliminar (*Set*|*Drop*) o valor padrão de uma coluna; *change column* para alterar o nome ou a declaração de uma coluna; *drop column* para excluir uma coluna de uma tabela, drop index para excluir um índice de uma tabela; *drop primary key* para excluir a chave primária de uma tabela; *rename* para alterar o nome de uma tabela;

- *Create Index*: cria um índice em uma tabela específica de um banco de dados selecionado. Pode ser um índice normal ou um índice sem duplicidade (*Unique*);
- *Create Table*: cria uma tabela em um banco de dados selecionado;
- *Delete*: utilizado para excluir um ou mais registros de uma determinada tabela em um banco de dados selecionado. São excluídos todos os registros que satisfizerem a expressão de pesquisa de registro. A clausula *Where* especifica a condição que as linhas devem ter para serem excluídas;
- *Drop Index*: exclui um determinado índice de uma tabela em um banco de dados selecionado. Equivale ao comando *Alter Table Drop Index*;
- *Drop Table*: exclui uma determinada tabela do banco de dados;
- *Insert:* utilizado para fazer a inserção de dados dentro de uma tabela em um banco de dados selecionado. Os parâmetros são: *Into* é o nome da tabela que vai ser inserida informação, *Values* inserir as linhas baseadas nos valores específicos, *Select* insere as linhas retornadas pelo comando *Select*;
- *Lock Tables*: bloqueia uma ou mais tabelas. Espera até conseguir bloquear todas as tabelas especificadas;
- *Optimize Table*: otimiza o espaço ocupado por uma tabela. Uma tabela fica fragmentada quando ocorrem muitas exclusões de registros. Durante o processo de otimização, o acesso à tabela é feito normalmente, exceto que as gravações são feitas em uma tabela temporária;
- *Repair Table*: repara tabelas danificadas;
- *Select*: o *select* é o comando mais utilizado em um banco de dados SQL, pois ele permite pesquisar dados armazenados em uma determinada tabela em um banco de dados. Os principais parâmetros do comando *Select* são: *From* especifica o nome das tabelas dos campos a serem utilizados no comando *Select*, *Where* especifica a condição que determina quais linhas serão retornadas, *Group By* retorna um valor para cada grupo de registros, *Order By* especifica as colunas a serem utilizadas como chave para ordenar os registros retornados, *Join* retorna todas as combinações possíveis de linhas das tabelas, sendo que a condição da junção é especificada na cláusula *Where*, *Left Join* retorna todas as linhas da tabela da esquerda mesmo que não exista equivalência na tabela da direita;
- *Set*: define opções para a sessão corrente do MySQL. As principais opções são: *Insert_Id* especifica o valor a ser usado no próximo comando *Insert* quando estiver inserindo em uma coluna *Auto_increment*, *Sql_Log_Update* determina se o arquivo de *update-log* deve ser atualizado (1) ou não (0), esse arquivo de *log* registra todos os comandos que alteram as tabelas sendo utilizado no backup, *Sql_Log_Off* determina se o arquivo de *log* padrão deve ser atualizado (0) ou não (1);
- *Show Columns*: exibe as informações sobre as colunas de uma tabela. O parâmetro *From* define a tabela em que vão ser mostrados os campos;
- *Show Databases*: lista todos os banco de dados de um servidor MySQL;
- *Show Grants*: exibe as informações sobre as permissões de um usuário. O parâmetro *For* identifica o usuário.
- *Show Index*: exibe as informações sobre os índices de uma tabela. O parâmetro FROM identifica a tabela;
- *Show ProcessList*: exibe as informações sobre os threads que estão sendo executados no servidor;
- *Show Status*: mostra as variáveis de status do servidor e seus valores;

- *Show Table Status*: exibe as informações sobre as tabelas em um banco de dados;
- *Show Tables*: lista todas as tabelas em um banco de dados;
- *Show Variables*: exibe uma lista das variáveis do servidor e seus valores;
- *Unlock Tables*: desbloqueia as tabelas que foram bloqueadas pelo usuário corrente;
- *Update:* utilizado para modificar campos em um registro em uma determinada tabela em um banco de dados. Os parâmetros principais são: *Set* especifica as colunas que serão atualizadas e os respectivos valores a serem atribuídos, *Where* especifica a condição que determina qual as linhas que serão atualizadas, senão, especifica todas as linhas que são modificadas.

O *MySQL* é um software muito estável, leve, *open source*, simples de instalar, fácil de utilizar e muito poderoso. Por isso, sua utilização tem crescido muito rapidamente, tanto que o site oficial do *MySQL* (http://www.mysql.com/) recebe em média 10 milhões de *page views* por mês, sendo a grande maioria usuários e programadores da comunidade Linux. Com a implementação de recursos como *foreign key* (chave estrangeira), *store procedures* e *triggers*, esse software será mais reconhecido pelos profissionais de informática, acadêmicos e pelas empresas.

6.2.3 Ferramentas para Desenvolvimento WML

No protótipo E-Móvel, foram utilizadas duas ferramentas para o desenvolvimento da interface e os testes do aplicativo. As ferramentas são o Easy Pad WAPTor 2.3 para o desenvolvimento das páginas WML, e o software UP.SDK 4.0 da Phone.com para testar o aplicativo através dos diversos emuladores que essa ferramenta disponibiliza.

O WAPTor é uma ferramenta que traz botões e menus com as principais *tags* da linguagem WML, facilitando, assim, a criação de uma página para dispositivo móvel. Ele também traz uma tela que simula como ficará a página criada na tela do dispositivo móvel, sendo que na verdade deve-se testar com emuladores reais dos telefones celulares, pois o WAPTor não utiliza o software de *microbrowser* utilizados pelas empresas de telefonia móvel.

O UP.SDK da Phone é um software que traz diversos emuladores de telefones celulares, para que sejam realizados testes reais do funcionamento de uma aplicação. Os emuladores utilizam o código do microbrowser da Phone.com que está presente em diversos dispositivos móveis utilizados no mundo inteiro. Outra vantagem desse software é que ele traz uma janela DOS que traz as informações de cada *deck* e dos *cards* carregados no emulador, sendo que essas informações são muito úteis para resolver problemas que acontecem durante o desenvolvimento de uma aplicação WML.

Outra ferramenta muito útil para o desenvolvimento de aplicação que acompanha o UP.SDK é o CertMaker 4.0 que é um software para criar um certificado digital para ser utilizado no aplicativo móvel quando esse acessa um servidor WAP da Phone.com. Para criar o certificado digital, o desenvolvedor precisa da chave de um *Certicate Authorith* (CA) que é a empresa confiável pela autenticação do certificado digital.

6.3 ANÁLISE E PROJETO DO E-MÓVEL

A ferramenta *Use Case* do *UML* facilita o entendimento de um sistema mostrando a sua "visão externa". Ela é utilizada para modelar o contexto de um sistema, subsistema ou classe. É uma das maneiras mais comuns de documentar os requisitos do sistema, de delimitar o Sistema, e de definir a sua funcionalidade.

6.3.1 Use Cases

Use Cases descrevem o que acontece dentro do sistema. O modelo conceitual apresentado na Figura 17, foi desenvolvido com a ferramenta *Use Case* do *UML*, por permitir, através da utilização de suas ferramentas (diagramas), o funcionamento estático e dinâmico do sistema. No TCCI, foi representada a modelagem estática do Sistema. A modelagem dinâmica do sistema foi demonstrada no TCCII, juntamente com o restante dos modelos UML.

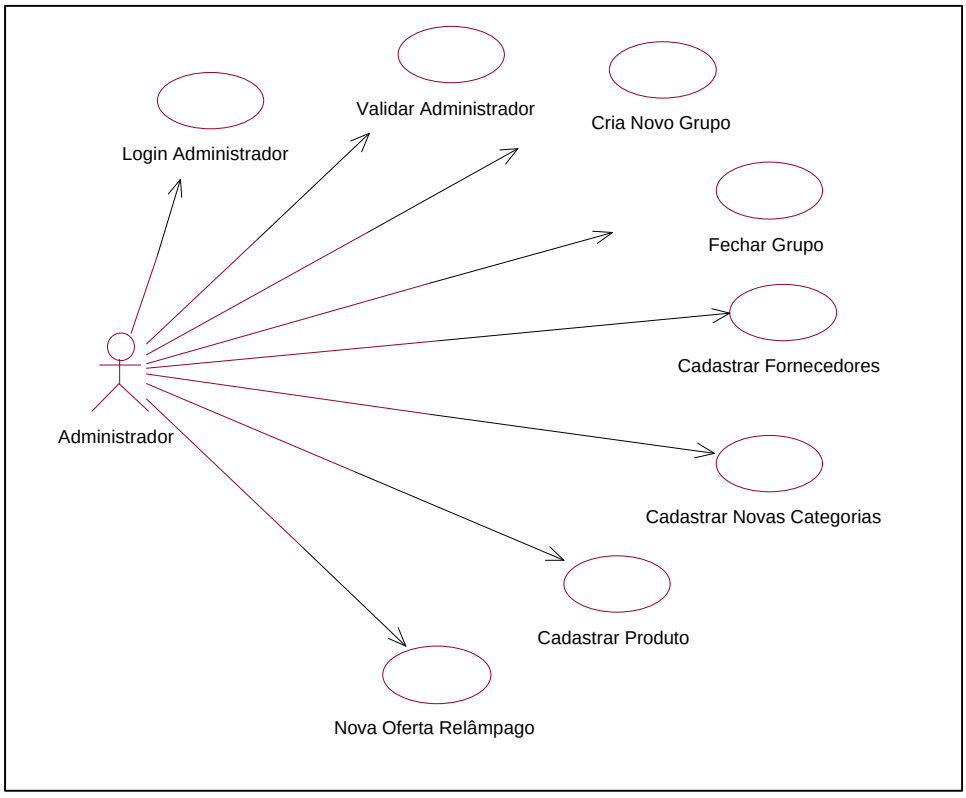


Figura 17 – Use Case – Visão Administrador

Quadro 2 – Funcionalidades do Use Case Administrador

Use Case	Descrição
Login Administrador	Faz o login no Sistema
Validar Administrador	Valida o usuário administrador no sistema
Novo Grupo	Cria um Novo Grupo de Compra
Cadastra Fornecedores	Adiciona um novo Fornecedor
Cadastra Produto	Adiciona um novo Produto
Cadastra Categoria	Adiciona uma nova Categoria
Nova Oferta Relâmpago	Adiciona uma nova oferta
Fecha Grupo	Contabiliza um grupo de compra, onde os valores que devem ser restituídos para os clientes são calculados

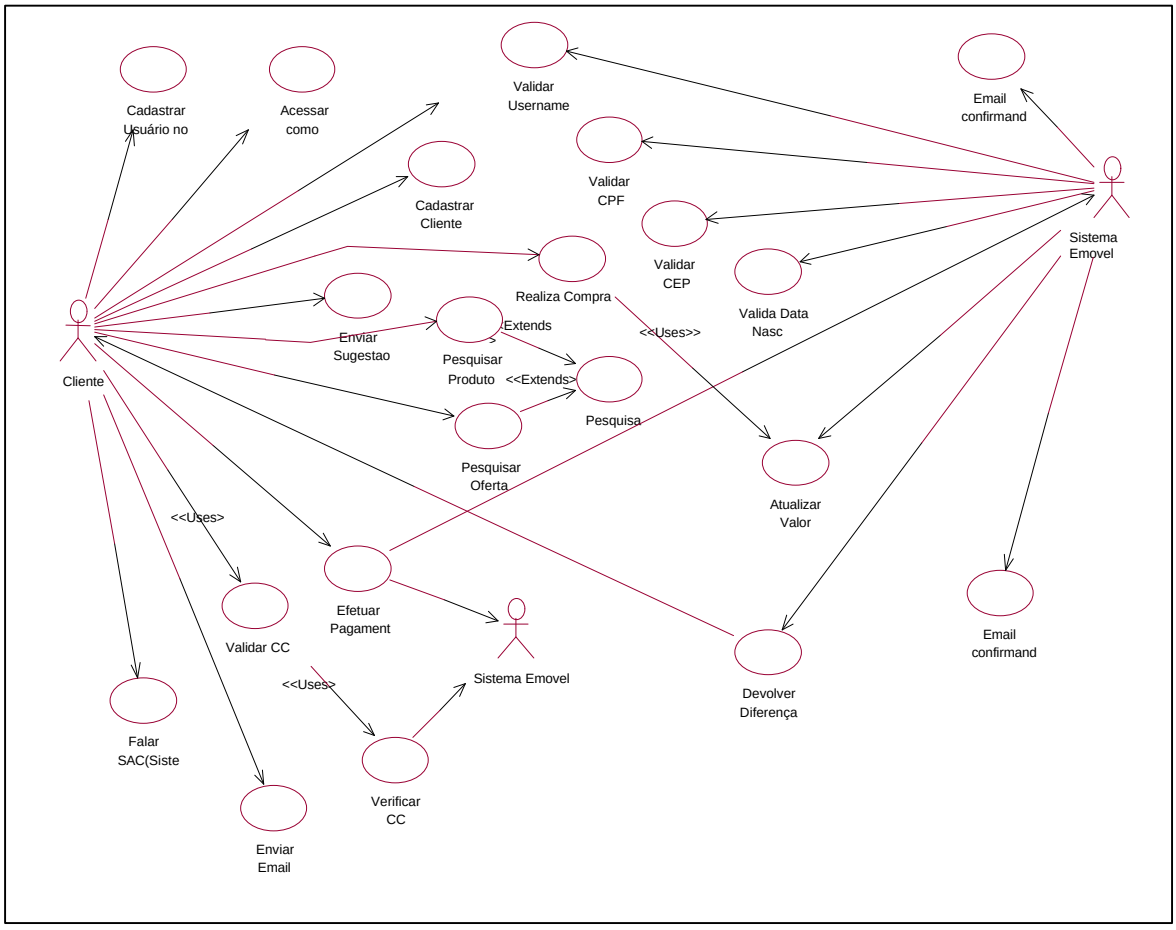


Figura 18 – Use Case – Visão Cliente

Quadro 3 – Funcionalidades do Use Case Cliente

Use Case	Descrição
Login	Faz o login no Sistema
Validar CC	Valida o Cartão de Crédito
Pesquisar	Pesquisa Produto
Enviar Email	Cliente envia um e-mail
Cadastra Cliente	Cadastra um novo cliente no sistema
Falar com SAC (Atendimento ao Cliente)	Disca automaticamente para o SAC de dentro do sistema E-Móvel
Realiza Compra	Cliente realiza a compra de um produto
E-mail Confirmando Compra	Sistema envia um e-mail confirmando compra
Devolver Diferença	Sistema Cálculo o valor a restituir aos clientes
E-mail confirmando cadastro	Envia e-mail confirmando o cadastro do cliente
Atualizar Valor	Atualiza Valor de um produto
Atualiza Estoque	Atualiza Estoque do produto

6.3.2 Diagrama de Classes

Uma classe é uma descrição de um conjunto de objetos com os mesmos atributos, relacionamentos, operações e semântica. Classes são usadas para capturar o vocabulário de um sistema. Classes podem ser abstrações do domínio do problema, como “Cliente”, “Banco”, “Conta”. Classes podem também ser usadas em nível de implementação (listas, filas, eventos).

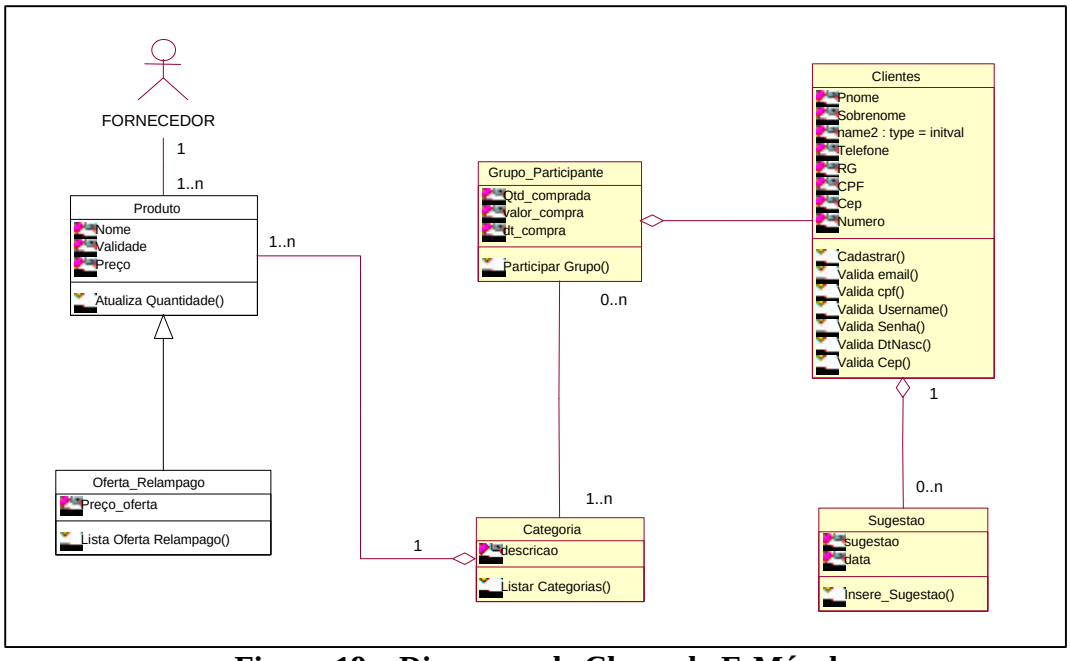


Figura 19 – Diagrama de Classe do E-Móvel

6.3.3 Diagramas de Seqüências (Cenários)

Interações mostram os aspectos dinâmicos de um sistema. Interações são usadas para modelar o fluxo de controle para uma operação, uma classe, um componente, um subsistema, ou para um sistema inteiro. Dois diagramas podem ser usados para modelar as interações: diagramas de seqüência e diagramas de colaboração.

No projeto, foram utilizados os diagramas de seqüência que enfatizam a ordenação das mensagens trocadas entre os objetos. Um cenário é uma seqüência específica de ações que ilustra um comportamento. Diagramas de seqüência podem modelar apenas um cenário ou um conjunto de cenários. Diagramas de seqüência podem mostrar decisões simples e interações. Foram demonstrados aqui somente os principais diagramas de seqüências implementados.

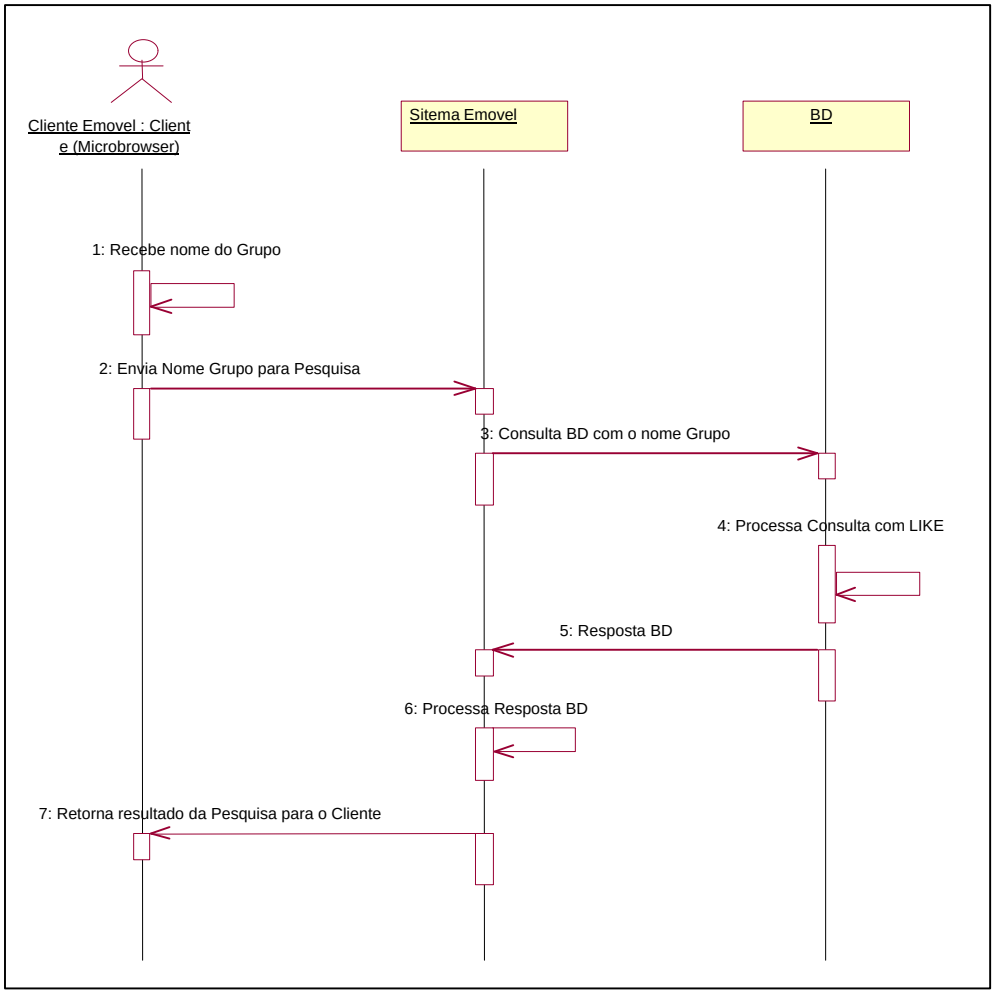


Figura 20 – Diagrama de Seqüência – Pesquisa Produto

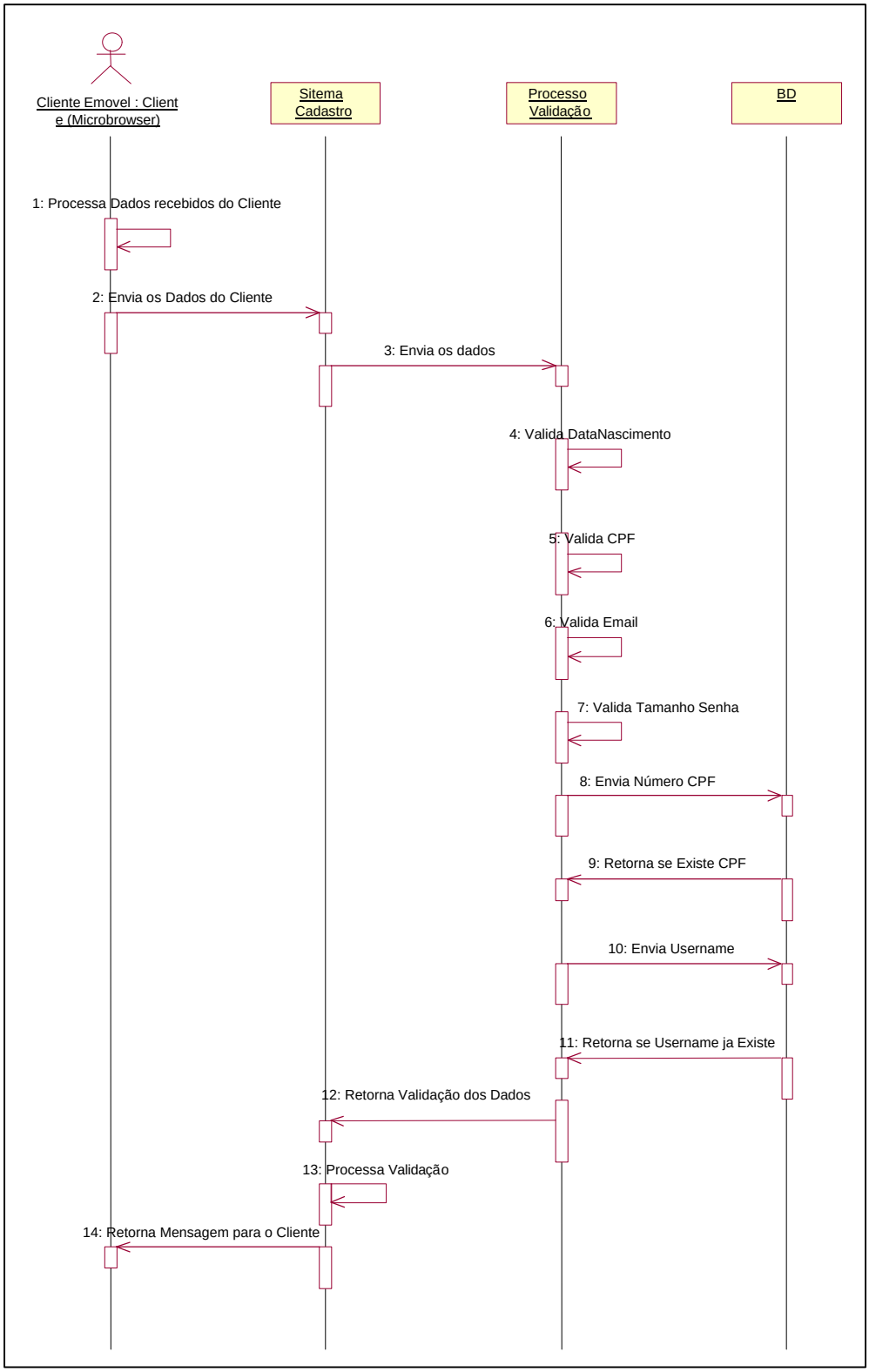


Figura 21 – Diagrama de Seqüência – Cadastro Novo Cliente

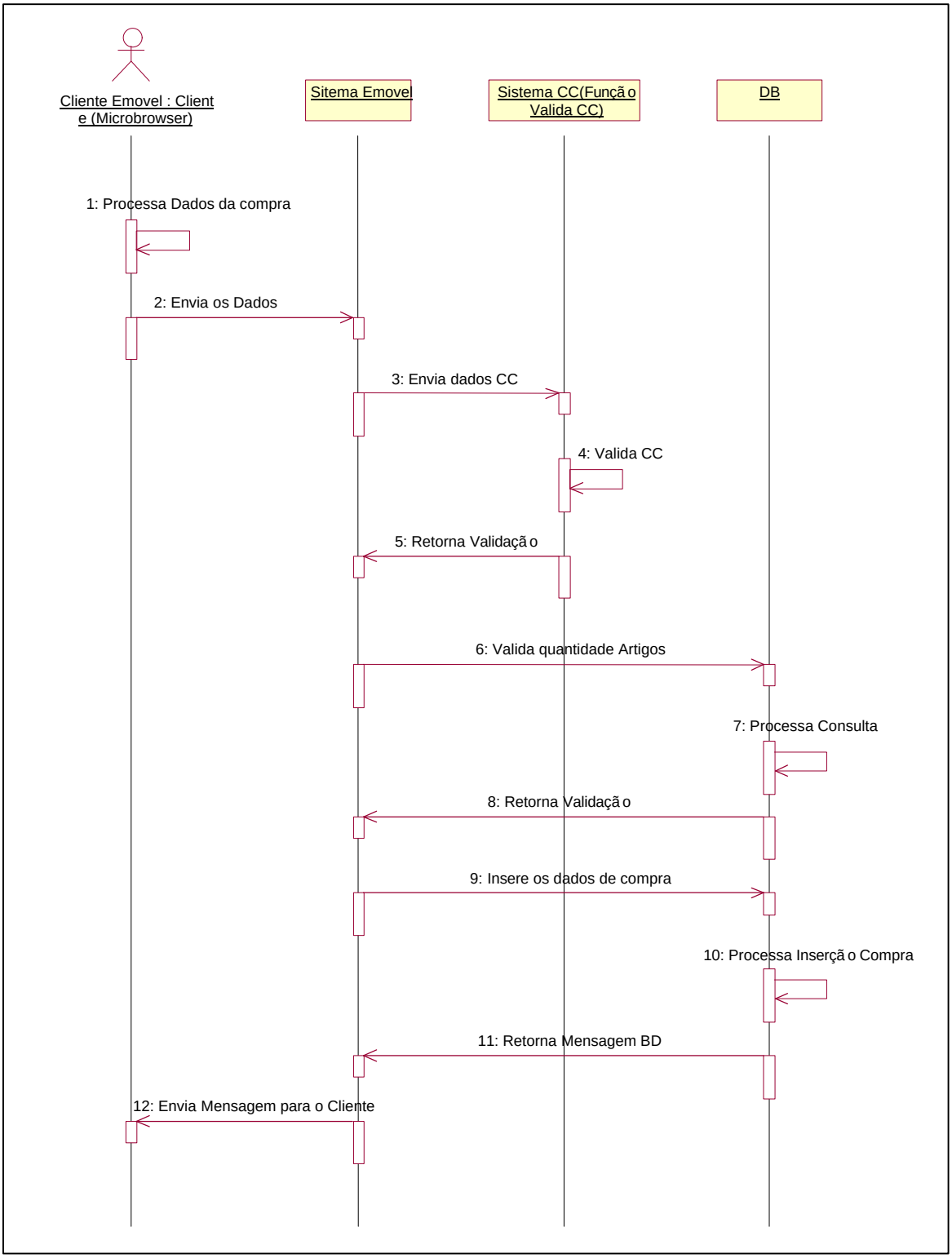


Figura 22 – Diagrama de Seqüência – Realiza Compra

6.4 DESENVOLVIMENTO DO E-MÓVEL

O desenvolvimento do E-Móvel foi baseado na linguagem de marcação do WAP chamada WML versão 1.2, sendo essa a única opção atual para a apresentação de conteúdo

para dispositivos WAP. A linguagem de programação para rodar no lado servidor e implementar a lógica de negócio do E-Móvel utilizada foi o PHP versão 4. Essa linguagem foi utilizada, porque nas pesquisas realizadas, foram identificados vários sites móveis em produção que foram criados com o PHP, entre eles sites como o Selig que foi criado em PHP tanto na versão Web, com na versão móvel. Outro site muito acessado, que foi construído com PHP, MySQL e Apache rodando em Linux, é o da revista Info Exame, tendo mais de 50.000 páginas no site. Os exemplos de aplicações móveis em PHP que auxiliaram no desenvolvimento desse protótipo, além de essa linguagem ser independente de ambiente computacional, ser *open source* e possui acesso nativo a grande maioria dos bancos de dados padrão SQL92.

O banco de dados utilizado foi o MySQL versão 3.22.32, suporta a maioria dos recursos necessários para implementar um protótipo de uma aplicação Internet, utilizado muitas vezes em ambiente de produção, como sites de comércio eletrônico. O MySQL também é *open source,* o que garante o seu aprimoramento constante e a resolução de problemas muito mais rápida do que muitos programas comerciais.

O sistema E-Móvel foi criado utilizando os recursos mais avançados para tornar o sistema o mais dinâmico possível. Outra característica desejada e recomendada pelos desenvolvedores mais experientes em sistema móveis foi evitar ao máximo a entrada de dados pelo cliente, porque a digitação de dados nos dispositivos móveis é muito complicada e o cliente pode desistir de utilizar o sistema. Segundo Forta (2001), outra funcionalidade que deve ser utilizada são os botões de opção que na maioria dos telefones celulares são apenas dois. Esses botões podem ser programados com as *tags* <do> e <go> e também a *tag* <*template*> que define o comportamento de um botão em todos os *cards* de um *deck.*

6.4.1 Página Inicial do E-Móvel

Na Figura 23 está representada a tela de entrada do sistema E-Móvel. Nela, as opções que o cliente dispõe são: *Login*, onde o usuário faz a autenticação no site, sendo necessário ele se cadastrar antes de acessar o sistema para realizar as compras. A opção Novo Usuário, o usuário pode realizar o seu cadastro via sistema móvel. A opção Visitante disponibiliza o acesso dos clientes sem que esses tenham feito um cadastro, mas nessa opção só é possível visualizar os produtos oferecidos ou pesquisar por um produto.

A última opção é o Atendimento ao Cliente, onde o cliente acessa uma página que tem uma função da biblioteca *WTAI* para discar para o serviço 0800 do E-Móvel. Para essa função funcionar, o *browser* do usuário tem que suportar a linguagem *WML* 1.2. Muitos sites utilizam essa função de discagem para realizar a compra nos seus sites de *M-commerce* para realizar as transações, em vez de criar um sistema totalmente móvel.

```
<do type="accept" label="Chamar">
<go href="wtai://wp/mc;+0800907030">
</go>
</do>
```

Figura 23 – Tela Inicial do Sistema E-Móvel

Um dos problemas enfrentados no desenvolvimento do sistema foi na utilização de imagens no site. Foi criado um pequeno símbolo para a loja E-Móvel, e o arquivo convertido para o formato WBMP que ficou com o tamanho de 2 Kb. O problema é que o tamanho máximo que um *deck*, arquivo enviado para o dispositivo móvel, não pode ser maior que do que 1,5 Kb (Forta, 2001). Essa limitação impossibilita qualquer inclusão de imagens para apresentar um produto ou utilizar como marketing no site. Por essa razão, não foram utilizadas imagens no desenvolvimento do sistema, sendo identificado nessa limitação um dos grandes problemas na tecnologia WAP disponível.

Quando o cliente seleciona a opção *Login*, a página de autenticação é apresentada. Nessa página, o cliente entra com o nome de usuário e a sua senha de acesso. A autenticação é realizada com a consulta dos dados do cliente no banco de dados, sendo que a senha foi armazenada criptografada, no banco de dados. O script que realiza a autenticação, consulta o banco de dados para conferir a validade dos dados digitados pelo cliente, criptografando a senha para comparar com a senha armazenada no banco de dados. A biblioteca *Mcrypt* foi utilizada, ela roda nos sistemas operacionais UNIX, e implementa os algoritmos de chave privada mais consagrados. Foram utilizadas as funções base64_encode e base64_decode para transformar o resultado da função mcrypt_cbc, que é um formato binário, para uma string de caracter, e vice-versa. Na Figura 24 é apresentado o código-fonte da função encript que realiza a criptografia dos dados secretos que são armazenados no banco de dados.

```
function encript($dado){

  $key = "=9DSC#D$e%$+";
  $IV = "$%GDS14#@1";
  $cripto = mcrypt_cbc(MCRYPT_TripleDES,$key,$dado,MCRYPT_ENCRYPT,$IV);
  $base64 = base64_encode ($cripto);
  return $base64;
}

function decript($dado){

  $key = "=9DSC#D$e%$+";
  $IV = "$%GDS14#@1";
  $cripto = mcrypt_cbc(MCRYPT_TripleDES,$key,$dado,MCRYPT_ENCRYPT,$IV);
  $base64 = base64_encode ($cripto);
  return $base64;
}
```

Figura 24 – Código das Funções encript e decript

O controle de sessão do E-Móvel foi realizado com a utilização das funções de sessão do PHP 4, sendo que com essas funções é possível controlar os dados dos clientes durante a sua navegação nas páginas WML, e também garante que nenhuma pessoa faça uma adulteração no site. As outras soluções que poderiam ser utilizadas seriam a utilização de *cookies*, mas segundo Forta (2001), a implementação do controle de sessão com *cookies* no ambiente móvel não é confiável e muitos erros ocorrem com a sua utilização, além do problema de segurança que a utilização de *cookies* proporciona. Ou utilizar as variáveis de sessão, que consiste em passar entre os *scripts* chamados as variáveis de sessão como parâmetro via *GET HTTP*, o problema é que as variáveis de sessão podem ser visualizadas por emuladores que rodam nos *browsers Web*, como o *WINAMP*, possibilitando que pessoas não autorizadas adulterem o sistema (Converse, 2001).

Muitas aplicações móveis são acessadas por telefones celulares, e como esses dispositivos muitas vezes são perdidos ou roubados, uma preocupação que deve ser solucionada é com a segurança dos dados que ficam no *cache* do telefone celular. Nas páginas onde existem dados que não podem ser armazenados por causa da segurança, como a senha, o número do cartão de crédito, a utilização do elemento *meta* na *tag* <*head*> faz o controle do *cache*. A utilização do comando abaixo faz com que o *deck* não seja armazenado no *cache* do dispositivo móvel:

```
<head>
<meta http-equiv="Cache-control" content="no-cache"/>
</head>
```

Outra programação que deve ser utilizada para garantir que os dados digitados pelo usuário não fiquem disponíveis entre as páginas, é utilizar a inicialização das variáveis dos campos de <*input*>. Campos como a senha, o número de cartão de crédito e outros dados devem ser inicializados no início do cartão (*card)* no evento *onenterforward*. Como no exemplo a seguir:

```
<onevent type="onenterforward">
<refresh>
      <setvar name="usuario" value="" />
      <setvar name="senha" value="" />
</refresh>
</onevent>
```

O WML 1.2 também fornece uma *tag* para restringir o acesso de outros aplicativos através de *links* no nos *decks* do sistema. A restrição pode ser feita no nível de domínio e também no nível de diretório do servidor Web. A *tag* utilizada para isso é a <*access*> dentro da *tag* <*head*>, e os atributos são *domain* para restringir o acesso para um determinado domínio da rede, e *path* que é utilizado para restringir o acesso somente para um determinado diretório (*folder)* do servidor Web. Essa técnica também foi utilizada no E-Móvel para negar o acesso do sistema através de outros aplicativos. Por exemplo:

```
<head>
  <access domain="www.emovel.com.br" path="/wap" />
</head>
```

Outro problema no desenvolvimento WML é o reconhecimento de caracteres especiais, como os caracteres com acento. Alguns telefones celulares reconhecem esses caracteres com acento corretamente, porém outros não reconhecem. Para garantir a apresentação correta do caracter com acentuação, é recomendada a utilização da codificação WML para os caracteres ASCII (Forta, 2001). O Quadro 4 apresenta os principais caracteres acentuados utilizados na língua portuguesa.

Quadro 4 – Código WML para Representar Caracteres com Acentuação

Código WML	Caracter ASCII
&#225;	Á
&#226;	Â
&#227;	Ã
&#231;	Ç
&#233;	É
&#234;	Ê
&#237;	Í
&#243;	Ó
&#244;	Ô
&#245;	Õ
&#250;	Ú
&#252;	Ü

6.4.2 Cadastro do E-Móvel

A opção Novo Usuário possibilita ao cliente se cadastrar no site do E-Móvel via dispositivo móvel. Muitos sites em produção não disponibilizam essa opção para seus clientes, sendo possível realizar o cadastro somente via a Web tradicional. Essa estratégia foi utilizada por muitos desenvolvedores de sistemas para dispositivos móveis, forçando os clientes a acessarem o site Web para realizar o cadastro. O objetivo deles é evitar que o cliente fique chateado pelas restrições de entrada de dados e não realize o cadastro. Mas se o cliente não possuir um acesso a Web, ele não tem como realizar o cadastro. No E-Móvel, foram disponibilizadas as duas formas de realizar o cadastro. Quando o cliente acessa o *link* para realizar o cadastro no dispositivo móvel, ele recebe um aviso mostrando que a forma de cadastro mais indicada para realizar o cadastro é na página da Web do E-Móvel. Mas se o cliente desejar, ele pode continuar o cadastro com o dispositivo móvel.

Se o cliente desejar alterar os seus dados cadastrais, a única opção é via Web tradicional, pois a alteração dos dados no dispositivo móvel é ainda mais complicada do que a entrada dos dados.

Os dados que são informados para realizar o cadastro do cliente são: o primeiro nome, o sobrenome, o nome de usuário para acessar o sistema, a senha para acessar o sistema, o email, o sexo, se o cliente deseja receber ofertas por email, o CEP, o número da casa ou apartamento, o CPF, a data de nascimento, sendo que esses campos são obrigatórios para realizar o cadastro. Ainda tem os campos renda familiar e telefone para o cliente realizar o cadastro. Os campos para realizar o cadastro são os mesmos no sistema móvel e na Web, sendo que as funções para validação dos dados são as mesmas. Conforme Forta (2001), não se deve utilizar a máscara de *password* na *tag* <*input*> da senha fornecida pelo usuário, porque, como a entrada de dados no dispositivo móvel é muito limitada, às vezes são necessários três toques em uma tecla para selecionar um único caracter e ficando impossível o usuário saber os caracteres digitados se ele não conseguir enxergar o que foi digitado. O problema é que o usuário não pode deixar ninguém observar, os dados enquanto ele se cadastra. Esse problema reforça a utilização da página de cadastro na Web tradicional, pois, acessando essa página, não existe a limitação de entrada de dados no microcomputador.

Como o MySQL ainda não suporta *store procedures*, as restrições no cadastro foram implementadas com funções criadas no PHP. Essas restrições basicamente foram: os campos obrigatórios não podem sem inseridos sem valor, para isso foram implementadas funções que verificam se os campos foram digitados. Essas funções são: num_casa_informado, pnome_informado, sobrenome_informado e valida_senha. Essas funções simplesmente verificam se o número da casa, o primeiro nome, o sobrenome e a senha foram informados. A função valida_senha ainda verifica se o tamanho mínimo de quatro caracteres foi informado. As funções que validam os outros dados informados pelo cliente são: valida_data, existe_username, valida_cpf, existe_cpf, valida_cep e valida_email. A função valida_data verifica se o formato da data de nascimento informada está correta. O formato de entrada da data é “dd/mm/aaaa”, sendo que as datas reconhecidas pelo sistema vão do ano 1800 até o ano 9999. Se a data estiver correta, a data informada é transformada para o formato “aaaa/mm/dd”, que é formato para gravação de dados no MySQL. Na função valida_data, foi utilizada a função do PHP que reconhece expressões regulares *ereg*, sendo que essa função do padrão POSIX serve para reconhecer uma determinada expressão. Outra função do PHP utilizada foi a *explode*, ela é utilizada para quebrar uma *string* em vetores, sendo que o caracter informado como parâmetro na chamada da função identifica os pontos onde a *string* é quebrada.

Outra característica implementada no sistema foi criar as funções de negócio do sistema em funções armazenadas em arquivos separados do código de apresentação. O PHP permite, utilizando o comando *include*, que se faça à inclusão de código de outros scripts em um script PHP, funcionando semelhantemente as bibliotecas da linguagem C. Isso permite a melhor organização do código-fonte, e também a reutilização de código.

A função existe_username verifica se já não existe um usuário cadastrado com o nome de usuário informado pelo cliente. Se existir o nome de usuário no sistema, um erro é retornado para o cliente, e esse deve escolher outro nome para ser cadastrado.

A função valida_email verifica se o formato de e-mail informado pelo cliente está correto, conforme a Figura 25. Ela simplesmente verifica se o formato está correto, e não se o endereço eletrônico existe realmente. Se o formato for inválido, um erro é gerado e uma mensagem é retornada para o cliente informando que o formato do e-mail digitado está

incorreto. Foi utilizada a função de reconhecimento de expressões regulares *eregi* do PHP que segue o padrão POSIX, para reconhecer o endereço informado.

```
<?php
function valida_email($email){
      $regex = '^([._a-z0-9-]+[._a-z0-9-]*)@(([a-z0-9-]+\.)*([a-z0-9-
]+)(\.[a-z]{2,3}))$';
      if (!eregi($regex,$email))
         return FALSE;
      else
         return TRUE;}
?>
```

Figura 25 – Código em PHP da Função valida_email

A função valida_cep demonstrada na Figura 26, foi implementada para verificar se o CEP informado existe nas tabelas referentes ao logradouro do sistema E-Móvel. Essas tabelas foram importadas da base do Correio do Brasil, e nessas tabelas são informados todos os logradouros, as localidades, os bairros e as UFs do Brasil. Sendo que com o CEP é possível identificar o logradouro, a localidade (cidade), o bairro e a UF do cliente. Assim, não é necessário que o cliente informe todos os dados referentes ao endereço do cliente. Se o CEP informado não existir na base de dados, um erro é gerado e uma mensagem é retornada para o cliente informando que um CEP válido deve ser informado.

```
function valida_cep($cep,$connection){
      if (strlen($cep) != 8)   return FALSE;
      else{
         $sql = "SELECT cep8_log FROM logradouros WHERE cep8_log='$cep'";
         $sql_result = mysql_query($sql,$connection);
         $nlinhas=mysql_num_rows($sql_result);
         if($nlinhas==1)
            return TRUE;
         else return FALSE;}
  }
```

Figura 26 – Código Função valida_cep

A função valida_cpf foi implementada para identificar se o número do CPF informado pelo cliente está correto. O algoritmo utilizado para implementar essa função é de domínio público, sendo fácil de implementar em qualquer linguagem de programação. Ela simplesmente verifica se os dois dígitos de verificação conferem com a lógica realizada com os outros números informados. A função existe_cpf verifica se o CPF informado ainda não existe na base do E-Móvel, se o número do CPF já existir, um erro é gerado e uma mensagem é retornada para o cliente, informando que o CPF já foi cadastrado no E-Móvel.

Se nenhuma das validações ocasionar um erro, o cadastro do cliente é realizado, e o código do cliente é gerado automaticamente pelo *AUTOINCREMENT* do MySQL. O código do cliente é utilizado como variável de sessão para controlar os passos do cliente durante a utilização do E-Móvel. Antes de realizar a inserção, é feito um *LOCK* de escrita na tabela de clientes para realizar a gravação sem problemas de sincronismo com outros usuários. O MySQL suporta somente *LOCK* no nível da tabela, e não no nível de registro como no Oracle e outros banco de dados. Nas tabelas consultadas, também é feito um *LOCK*, mas somente para leitura dessa tabela. O MySQL trabalha com *threads,* e a utilização de *LOCKs* trabalha

diretamente com as *threads*. Quando uma *thread* realiza um *LOCK* para escrita em uma tabela, todas as outras *threads* esperam a liberação do *LOCK* para acessar essa tabela. Quando o *LOCK* é para leitura em uma tabela, as outras *threads* podem realizar leituras nessa tabela. O comando do MySQL para realizar o desbloqueio das tabelas é o *UNLOCK*.

6.4.3 Acesso como Visitante no E-Móvel

O cliente pode acessar o sistema E-Móvel como visitante. Com esse acesso, ele pode listar toda a oferta relâmpago disponíveis naquele momento, ou pesquisar pelo nome de um determinado produto, conforme a Figura 27. Se o cliente selecionar a opção que lista as ofertas relâmpagos disponíveis, o sistema consulta todos os grupos que estão em oferta relâmpagos naquele instante e que não estejam fechados. Quando o cliente escolhe a opção pesquisar, uma página solicitando a palavra chave para a pesquisa é apresentada. O cliente então entra com o nome do produto desejado e é realizada uma pesquisa na base de dados do E-Móvel. Essa pesquisa procura pelo nome do produto na tabela artigos, comparando a palavra digitada com a descrição dos produtos, utilizando o atributo de comparação *Like* na consulta ao banco de dados.

Figura 27 – Acesso como Visitante no E-Móvel

6.4.4 Acesso como Usuário no E-Móvel

Quando o cliente faz o *login* no sistema E-Móvel, o código desse cliente é utilizado como variável de sessão, sendo passado entre as páginas WML para ser possível recuperar as informações do cliente quando necessário. As opções que o cliente pode selecionar depois do *login* são mostradas na Figura 28.

Se o cliente selecionar a opção “Sugestão E-Móvel”, uma página será exibida e um campo de entrada de dados (*tag* <*input*>), para que o cliente digita a sua sugestão de produto ou alguma melhora no site do E-Móvel. Quando o cliente clicar em *enviar*, a sugestão será gravada na base dados do E-Móvel.

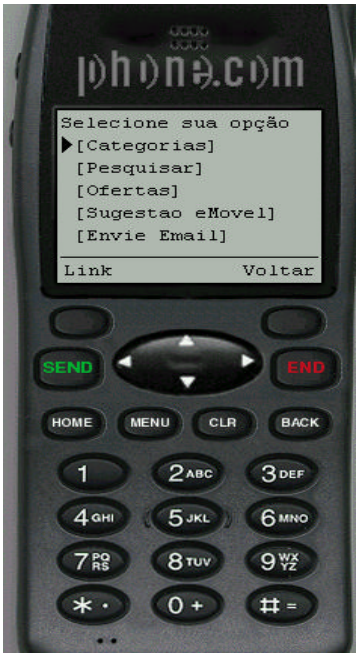


Figura 28 – Opções para o Cliente depois do *Login*

Se o cliente selecionar a opção “Envie Email”, uma página será exibida para o cliente, como os campos para a entrada dos dados para o envio de um email. Os dados que o cliente precisa informar são: o email do destinatário, o título da mensagem, o corpo da mensagem. O email do destinatário, o E-Móvel busca nos dados cadastrais do cliente. Esse é um serviço gratuito que o E-Móvel disponibiliza para seus clientes, para que ele utilize o site E-Móvel sempre que precise enviar um email, mas não possa utilizar um PC. Assim, o cliente sempre retornará ao site, e a probabilidade de ele realizar uma compra fica muito maior.

Quando o cliente seleciona a opção “Categorias”, o sistema consulta a base de dados e mostra todas as categorias que possuem pelo menos um produto que esteja sendo oferecido em um grupo aberto. Se não existir nenhum produto oferecido no momento, uma mensagem é retornada ao cliente indicando que o site não está com nenhum grupo de compra sendo oferecido no momento. Quando o cliente seleciona uma categoria, o sistema E-Móvel cria uma página dinâmica mostrando todos os grupos oferecidos com um produto dessa categoria, conforme a Figura 29.

Deve-se notar aqui a escalabilidade e o poder da utilização do banco de dados para criar sites dinâmicos, pois quando uma nova categoria for criada e um cliente acessar o site, vai visualizar essa nova categoria.

```
$sql = "SELECT cat.codigo,cat.descricao
                FROM categorias cat, artigos art, grupo_prod_venda grupv
                WHERE cat.codigo = art.cod_cat
                AND art.codigo = grupv.cod_artigo
                AND grupv.grupo_fechado = 'N'
                AND now() < grupv.dt_final_grupo
                GROUP BY cat.codigo,cat.descricao
                ORDER BY cat.descricao;";
      $sql_result = mysql_query($sql,$connection);
      $nlinhas=mysql_num_rows($sql_result);
      if($nlinhas > 0 ){
            echo "<p align=\"center\">";
            echo "Selecione a Categoria:" ;
            echo "</p>";
            while ($row = mysql_fetch_array($sql_result)) {
                  $codigo = $row[0];
                  $descricao = $row[1];
                  $parametro = $codigo;
                  echo "<p>";
echo "<a href=\"http://127.0.0.1/wap/ecommerce/E-
Móvel/lista_artigos.php?parametro=$parametro\">$descricao</a>";
                  echo "<br/>";
                  echo "</p>";
            }
       }
```

Figura 29 – Código PHP para Criar a Página Dinâmica das Categorias

Conforme Forta (2001), quando se desenvolve um sistema para dispositivo móvel, deve-se sempre que possível possibilitar que o usuário volte para a página anterior. Assim, no E-Móvel, sempre que foi possível foi colocado um *link* denominado “Voltar” para retornar à página anterior.

Quando o cliente escolhe a opção “Pesquisar”, uma página solicitando a palavra chave para a pesquisa é apresentada. O cliente entra com o nome do produto desejado, é realizada uma pesquisa na base de dados do E-Móvel. Essa pesquisa procura pelo nome do produto na tabela artigos, comparando a palavra digitada com a descrição dos produtos, utilizando o atributo de comparação *Like* na consulta ao banco de dados. Todos os produtos encontrados no sistema E-Móvel, que estejam sendo oferecidos em um grupo e que esse grupo não esteja fechado, serão mostrados para o cliente. A descrição do produto foi utilizada como *link* para abrir uma página com os detalhes do produto oferecido, onde dados como a descrição detalhada do produto, a quantidade vendida, o valor atual do produto, a data de fechamento do grupo, o valor do produto para cada faixa de quantidades vendidas, são mostrados ao cliente. Um *link* para que seja realizada a compra é disponibilizado para o cliente realizar a compra do produto. Aqui, um dos problemas é que o cliente precisa rolar a tela para verificar todas as informações e acessar o *link* “Comprar” para efetuar a compra, mas infelizmente não existe outra maneira de criar uma página com um pouco mais de informações sem ter que rolar a tela, ou quebrar essas informações em mais páginas forçando o usuário além de rolar um pouco a tela, ainda ter que clicar no botão de opção ou em um *link* para acessar o restante das informações em outra página.

A opção “Oferta”, o sistema faz uma seleção de todos os grupos que estão em oferta relâmpago, ou seja, os grupos que estão com um preço muito baixo e têm a data de fechamento muito pequena, às vezes somente horas. Esse tipo de oferta é utilizado para fidelizar o cliente móvel, fazendo com que ele compre de qualquer lugar e na hora desejada

uma oferta. Os produtos são listados como *link* para a página que lista os detalhes do grupo. A página que lista os detalhes de um grupo, que está no script artigo_detalhe.php, é a mesma tanto para a opção inicial “Categoria”, como para a opção “Pesquisar” e para a opção “Ofertas”, pois ela é dinâmica, e recebe como parâmetros o código do cliente e o código do grupo desejado. A Figura 30 mostra a tela com os detalhes do grupo de compras.

Figura 30 – Página com os Detalhes do Grupo de Compra

Quando o cliente seleciona o grupo de compra que deseja participar e clica no *link* “Comprar”, o sistema mostra a página que o informará à quantidade que ele deseja comprar, o tipo de cartão de crédito que o cliente possui e o número do cartão de crédito. A informação da quantidade foi colocada em um *tag* <select> dinâmica, ou seja, ela é criada a partir da quantidade ainda disponível para venda. Se um produto “X” só tem quatro itens restantes no grupo de compras, o sistema disponibiliza a seleção de um até quatro itens para o cliente comprar. Essa técnica foi utilizada porque o E-Móvel tem como um dos objetivos vender somente os produtos que existam em estoque, e não fazer como a grande maioria dos sites de comércio eletrônico, que vendem o produto e depois enviam um email para o cliente avisando que o produto não existe em estoque e vai demorar trinta dias para ser entregue, ou que a compra foi cancelada. Esse tipo de acontecimento provoca uma revolta e uma desilusão no cliente, sendo que as possibilidades do cliente voltar a realizar compras nesse site são mínimas.

Figura 31 – Dados da Compra Enviados pelo Método *POST* do HTTP

Quando o cliente clica no botão de opções do celular, as duas opções disponíveis são “Voltar” para a tela anterior onde ocorreu à entrada dos dados, ou “Finalizar Compra” para confirmar os dados para realizar a compra do grupo de compra. Caso o cliente escolha a opção para realizar a compra, o código do cliente, o código do grupo de compra, a quantidade desejada para a compra, o tipo do cartão de crédito e o número do cartão de crédito são passados pelo método *POST*, sendo que com esse método os parâmetros com os valores das variáveis são passados sem aparecerem na *URL* de destino, evitando que um cliente utilizando um *browser* como o WINAMP visualize esses valores. A Figura 31 mostra as opções que o cliente tem no momento de realizar a compra.

A primeira validação que o sistema E-Móvel faz é verificar a validade do cartão de crédito. No protótipo, foi utilizada uma função que válida o número do cartão de crédito das maiores operadoras de cartão de crédito do mercado. Se o sistema E-Móvel estivesse em produção, essa função seria substituída pela validação digital diretamente no sistema da operadora de cartão de crédito.

Se o cartão for inválido, o sistema gera um erro e retorna uma mensagem para o cliente informando que o cartão de crédito não é válido. Mas se o cartão de crédito for válido, o sistema verifica se a quantidade informada na página dos detalhes do produto pode ser atendida, conforme o código da Figura 32. Esse teste se deve ao fato que pode ter ocorrido uma venda para outro cliente em outra sessão, enquanto o cliente da sessão atual realiza a sua entrada de dados para a compra.

```
function    verifica_quantidade($connection,$codigo_grupo,$quantidade){
      $codigo_artigo = busca_codartigo($connection,$codigo_grupo);
      $sql = "SELECT MAX(grupqtd.qtd_fim) - grupv.qtd_vendida
              FROM artigos art, grupo_prod_compra grupv, grupo_valor_qtd
grupqtd
             WHERE art.codigo = grupv.cod_artigo
             AND grupv.codigo = grupqtd.cod_grupo_compra
             AND grupv.codigo = 1
             AND grupv.grupo_fechado = 'N'
             AND now() < grupv.dt_final_grupo;";
     $sql_result = mysql_query($sql,$connection) or die ("A consulta do
artigo Falhou!");
     while ($row = mysql_fetch_array($sql_result)) {
             $estoque = $row[0];
     }
     if($estoque >= $quantidade)
      return TRUE;
     else
      return FALSE;
}
}
?>
```

Figura 32 – Código da Função verifica_quantidade

Se existir em estoque a quantidade que o cliente deseja comprar, então o sistema realiza o *Lock* das tabelas onde serão gravados os dados da compra e as tabelas que serão atualizadas com os valores utilizados na compra.

O *Lock* evita que outras *threads* de outras sessões realizem operações nessas tabelas, enquanto as transações da compra não são efetivadas no banco de dados. Segundo Converse (2001) essa técnica é indicada para simular o controle de transações com o banco de dados *MySQL*, pois esse SGBD não tem implementado o controle de transações como existe em banco de dados como o Oracle. Com a utilização do comando *Lock* e do comando *Unlock*, fica garantido o controle de acesso concorrente nas tabelas, e assim problemas como leitura de dados já alterados não ocorre.

Depois da realização do *Lock* nas tabelas, o sistema verifica o valor atual do grupo de compra que o cliente deseja comprar. Esse valor é utilizado para armazenar o valor que o cliente pagou pelo produto, para criar a nota fiscal e enviar as informações por e-mail para o cliente. A Figura 33 mostra a função valor_artigo que retorna o valor por unidade a ser pago pelo cliente.

```
function valor_artigo($codigo_grupo,$quantidade,$connection)
{

      $sql_vlatual = "SELECT  grupqtd.valor
                              FROM artigos art, grupo_prod_compra grupv,
grupo_valor_qtd grupqtd
                              WHERE art.codigo = grupv.cod_artigo
                              AND grupv.codigo = grupqtd.cod_grupo_compra
                              AND grupv.codigo = $codigo_grupo
                              AND grupv.grupo_fechado = 'N'
                              AND now() < grupv.dt_final_grupo
                              AND grupv.qtd_vendida+$quantidade between
grupqtd.qtd_ini and grupqtd.qtd_fim
                              GROUP BY grupqtd.valor;";
      $result = mysql_query($sql_vlatual,$connection);
      $nlinhas_vlatual=mysql_num_rows($result);
      while ($rowvl = mysql_fetch_array($result)) {
            $vlatual = $rowvl[0];
      }
      return $vlatual;
}
```

Figura 33 – Código da Função valor_artigo

Essa função soma a quantidade que já foi vendida no grupo de venda mais à quantidade que o cliente deseja comprar e testa em qual faixa de preço essa quantidade está na tabela *grupo_valor_qtd* que armazena os valores para cada faixa de quantidade vendida. Para identificar o valor total da compra, é só multiplicar a quantidade que o cliente deseja comprar com o valor atual retornado pela função valor_artigo.

Após, a função insere_participante é chamada. Essa função faz a inserção dos dados do cliente, do grupo, e da compra realizada pelo cliente na tabela grupo_participantes. Essa tabela armazena todas as compras realizadas pelos clientes dos grupos de compra. O número do cartão de crédito é criptografado com o algoritmo *DES Triplo* antes de ser armazenado, para garantir o sigilo dessa informação do cliente.

Depois é chamada a função atualiza_estoque que atualiza a quantidade em estoque do produto comprado pelo cliente.

A próxima função chamada é a atualiza_grupo_compra. Essa função atualiza a quantidade vendida do grupo de compra. Também verifica a quantidade em estoque do produto. Se o estoque chegou a zero, então o grupo de compra deve ser fechado, para que esse grupo não seja mais oferecido aos clientes do E-Móvel.

A última função chamada, que fecha a transação de compra é a envia_email, conforme o código na Figura 34. Essa função é responsável por criar o email com os dados da compra realizada para o cliente e enviar esse email para o cliente. Como o WML ainda não dispõe de nenhum serviço de email nativo, foi utilizado o serviço de email do PHP. Como no desenvolvimento do sistema não estava disponível um servidor de email na rede local, foi utilizada uma classe PHP com o código-fonte aberto que realiza a conexão com um servidor de email na Internet e envia o email do cliente por ela. Essa classe se chama POP, e ela utiliza a função *fsockopen* que abre uma conexão via *socket* com um servidor de SMTP para utilizar os serviços desse servidor. Ela teve que ser revisada, porque continha alguns erros em relação à utilização da data do sistema. A configuração que deve ser realizada nessa classe é o nome do servidor de email e o nome de um usuário cadastrado nesse sistema. Para descobrir o nome

verdadeiro de um servidor de email deve-se utilizar o comando *telnet* na porta 25, que é a porta de email. No protótipo E-Móvel foi utilizado o servidor do Ulbra Gravataí, sendo que o nome verdadeiro do servidor é *gravatai.ulbra.tche.br*. O usuário utilizado para realizar a conexão foi "rodrigol". Os dados enviados por email para o cliente são o código do pedido (número da nota fiscal), o código do cliente, o nome do cliente, o código do produto, a descrição do produto, a quantidade comprada, o valor total da compra, a data da compra e o telefone do E-Móvel em caso de problemas na entrega do pedido.

```
...
  $smtp_sock = fsockopen($server, 25, &$errno, &$errstr, 30);
        if (!$smtp_sock) {
                        return FALSE;
        } else {
            while (!feof($smtp_sock)){
                fputs($smtp_sock, "HELO ".$this->smtp_server."\r\n");
                $this->echos($smtp_sock,4096,$this->verbel);
                fputs($smtp_sock, "MAIL FROM:<".$this->mailFrom.">\r\n");
                $this->echos($smtp_sock,4096,$this->verbel);
                fputs($smtp_sock, "RCPT TO:<".$this->mailTO.">\r\n");
                $this->echos($smtp_sock,4096,$this->verbel);
                fputs($smtp_sock, "DATA\n");
                $this->echos($smtp_sock,4096,$this->verbel);
                fputs($smtp_sock, "From: ".$this->mailFrom."\nSubject:
".$this->mailSubject. "\nDate:". $this->MailDate. "\nTo: ".$this-
>mailTO."\n\n".$this->mailText."\n");
                $this->echos($smtp_sock,4096,$this->verbel);
                fputs($smtp_sock, "\r\n.\r\n");
                $this->echos($smtp_sock,4096,$this->verbel);
                fputs($smtp_sock, "quit\n");
             }...
```

Figura 34 – Código da Função email_sem_server

6.4.5 Cuidado com Páginas Dinâmicas

Uma característica importante no desenvolvimento de um sistema móvel dinâmico, é o cuidado em relação ao limite máximo do tamanho do *deck*. Simplesmente porque se o tamanho de uma página ultrapassar 1,5 *Kbytes* ocorrerá um erro no momento da carga e o sistema será abortado imediatamente. Segundo Forta (2001), o ideal é gerar páginas de no máximo 1 *Kbyte*, para garantir que não ocorram problemas de travamento do sistema. O desenvolvedor deve tomar um cuidado especial com essa limitação, pois não tem como evitar que o sistema trave se o limite de memória for ultrapassado.

No E-Móvel, todas as páginas geradas dinamicamente acessando o banco de dados tiveram que ser limitadas para mostrar no máximo nove registros retornados de uma consulta por vez. Se a consulta ao banco de dados retornar mais do que nove linhas, o sistema gera um *link* dinâmico chamado "Mais...", e cria uma página nova dinamicamente com os próximos registros a serem exibidos.

6.4.6 Acesso como Administrador

Para administrar o site foi criado um subsistema de administração do E-Móvel. O acesso à área de administração foi colocado em um diretório chamado *Admin* no servidor Web, onde existe uma opção de *Login* para os administradores.

As opções que os administradores tem a disposição são: Criar uma nova categoria de produtos, criar um novo produto, cadastrar um novo fornecedor, criar um novo grupo de compra e contabilizar um grupo de compra que já terminou o período de compra. Não existe a opção de criar novos administradores por questões de segurança.

A opção criar uma nova categoria, adiciona uma categoria ainda não existente na tabela de categorias do sistema. A opção novo fornecedor adiciona um novo fornecedor na tabela de fornecedores do sistema, aqui existe a validação do número do “cnpj” e do email do fornecedor. A opção para “Criar Novo Produto” adiciona um novo produto na tabela do sistema, aqui as opções de categoria e fornecedor são colocadas na *tag* <*select*> dinamicamente para o administrador realizar o cadastro.

A opção para criar um novo grupo adiciona um novo grupo de compras nas tabelas do grupo de compras e os intervalos de quantidade vendida e valor para esse novo grupo de compras.

A opção contabiliza grupo de compras, possibilita ao administrador realizar o processo de contabilizar um grupo de compras que já está fechado e com o período de validade terminado. O termo contabilizar, nesse caso, consiste em verificar quais os clientes que pagaram o valor maior pelo produto de um grupo de compras que depois fechou com um valor mais baixo, por causa da participação de novos clientes. Esse processo insere na tabela de devoluções os valores que serão creditados no cartão de crédito dos clientes que tem direito a restituição de dinheiro, como pode ser visto na Figura 35.

```
<?php
     header("Content-type: text/vnd.wap.wml");
     echo "<?xml version=\"1.0\" ?>"; ?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.xml">
<wml>
<head>
  <access domain="www.emovel.com.br"  path="/wap" />
</head>
<template>
      <do type="options" label="Voltar"><prev/></do>
</template>
<card id="contabil" title="eMovel">
<?php
      include("db.php");
      conecta_db(&$connection);
      $sql_vlatual = "SELECT  grupqtd.valor
                       FROM artigos art, grupo_prod_compra grupv,
grupo_valor_qtd grupqtd
                      WHERE art.codigo = grupv.cod_artigo
                            AND grupv.codigo = grupqtd.cod_grupo_compra
                            AND grupv.codigo = '$codigogrupo'
                            AND now() >= grupv.dt_final_grupo
                            AND grupv.qtd_vendida between grupqtd.qtd_ini
and grupqtd.qtd_fim
                           GROUP BY grupqtd.valor;";
      $result = mysql_query($sql_vlatual,$connection);
      $nlinhas_vlatual=mysql_num_rows($result);
      while ($rowvl = mysql_fetch_array($result)) {
            $vlfinal = $rowvl[0];
      }
      $sql = "INSERT into devolucao_clientes
               SELECT
cod_grupo,cod_cliente,qtd_comprada,valor_compra,valor_unitario,dt_compra,nu
merocc,
                    '$vlfinal',qtd_comprada * (valor_unitario -
'$vlfinal'),now()
                                FROM grupo_participantes
                                WHERE cod_grupo = '$codigogrupo'
                                AND (qtd_comprada * (valor_unitario -
'$vlfinal')) > 0;";
      $sql_result = mysql_query($sql,$connection);

      $sql_update = "UPDATE grupo_prod_compra SET contabilizado = 'S' WHERE
codigo = '$codigogrupo';";
      $sql_result = mysql_query($sql_update,$connection);
      mysql_close($connection);
      echo "<p>";
      echo "<a
href=\"http://10.1.1.20/wap/ecommerce/admin/escolha_admin.php\">Voltar</a>"
;
      echo "</p>";?>
</card> </wml>
```

Figura 35 – Código da Página que Contabiliza um Grupo de Compra

6.5 MANUAL DO USUÁRIO

A utilização do sistema E-Móvel via telefone celular ou um PDA é bem fácil, sendo um dos objetivos do trabalho criar um site funcional e simples de utilizar.

O cliente deve acessar primeiro a tela inicial do sistema, que se localiza no endereço http://ulbra.gravatai.tche.br/wap/ecommerce/index.php. Na primeira página do sistema aparecem as opções disponíveis para o cliente.

Se o cliente não for cadastrado ainda no sistema, ele pode fazer o cadastro via o dispositivo móvel acessando as páginas WML, ou acessar a página Web de cadastro no site http://ulbra.gravatai.tche.br/wap/ecommerce/emovel.php. Os dados de cadastro que o usuário deve informar obrigatoriamente estão com um "*" no nome do campo de entrada. O sistema verifica os dados informados pelo usuário para consistir as informações, sendo que o nome de usuário no sistema e o CPF são únicos. Se o cadastro for realizado com sucesso o sistema envia um email para o endereço eletrônico do cliente informando que o seu cadastro foi realizado com sucesso no sistema.

O cliente pode acessar o sistema como visitante, onde existem duas opções, uma que é listar todas as ofertas disponíveis no sistema e a outra opção é uma pesquisa pelo nome de um determinado produto, sendo que os produtos semelhantes à chave de pesquisa que estão sendo oferecidos serão mostrados ao cliente.

O cliente também pode falar diretamente com o SAC (Serviço de Atendimento ao Cliente), simplesmente ele acessa o *link* "Falar Conosco", e o sistema liga automaticamente para o serviço 0800 do E-Móvel. Essa funcionalidade é executada somente quando o cliente estiver utilizando um telefone celular com o *microbrowser* que suporta a linguagem WML 1.2.

Para realizar compras e utilizar os outros serviços disponíveis no sistema, o cliente tem que fazer o *Login* no sistema. O cliente informa o seu nome de usuário e a senha para acessar as opções disponíveis. Após realizar o *Login* no sistema, o cliente pode listar todas as categorias existentes no sistema que tenha produtos oferecidos no momento, e depois de selecionar uma categoria o sistema lista todos os produtos daquela categoria que estão sendo oferecidos.

O cliente pode enviar uma sugestão para o site, que pode ser uma reclamação do sistema, uma sugestão de produtos para oferecer em grupos de compra, ou até mesmo um elogio ao site. Essa sugestão fica armazenada no banco de dados do sistema e será utilizada para oferecer novos grupos aos clientes.

Existe a opção do cliente enviar um email pelo sistema. Essa opção é importante pois é um serviço gratuito oferecido aos clientes, e por ser um serviço muito utilizado faz com que os clientes acessem sempre o sistema quando precisarem enviar um email e não tiverem um computador por perto. Acessando o sistema, o cliente tem grande possibilidade de verificar as ofertas e os produtos oferecidos, aumentando as chances de vendas.

O cliente pode pesquisar pelo nome um determinado produto e se encontrar, pode clicar no *link* do produto e verificar todos os detalhes desse produto, como valor atual para compra, data de fechamento do grupo de compra desse produto, valores pela quantidade vendida do produto.

O cliente também pode pesquisar todas as ofertas que estão sendo oferecidas naquele instante. Essas ofertas têm a característica de possuírem um tempo de duração muito pequeno

do grupo no máximo de 24 horas, sendo que os preços praticados são bem atraentes para os clientes.

Se o cliente desejar pode clicar no *link* comprar que a página para realizar a compara será aberta. Nessa página, o cliente pode selecionar a quantidade do produto que ele deseja comprar, o tipo do cartão de crédito que ele vai utilizar para realizar a compra e o número do cartão de crédito. Uma observação importante que o sistema gerencia o número de produtos que podem ser vendidos, por isso a quantidade que o cliente pode escolher para comprar nunca será maior que a quantidade ainda restante no grupo. Para garantir isso, foi implementado no sistema a *tag* <*select*> dinâmica com a lista de 1 até o limite máximo que o cliente pode comprar.

O sistema então faz a validação do número do cartão de crédito e de sua validade, e realiza todas as operações de atualizações na base de dados, finalizando a compra do cliente. O cliente recebe na tela do celular um aviso dizendo que um email foi enviado para o endereço eletrônico do cliente com os detalhes da sua compra.

Quando o grupo de compras estiver completo ou quando terminar o prazo de duração do grupo de compras, os clientes que compraram um produto com o valor maior que valor final desse produto, recebem a restituição do dinheiro no cartão de crédito.

6.6 CONCLUSÃO SOBRE O PROTÓTIPO

Com o desenvolvimento do protótipo de *M-commerce* E-Móvel, foi possível identificar as dificuldades enfrentadas com a tecnologia móvel atual para a construção de uma aplicação utilizável. Desde a fase de análise e projeto até o desenvolvimento da aplicação, as pessoas envolvidas no projeto devem trabalhar direcionadas pelas limitações impostas pela tecnologia WAP.

Um dos principais problemas enfrentados no desenvolvimento foi à parte de correção dos erros no código PHP, pois, diferente do que ocorre com o HTML o WML não mostra para o programador a linha do código onde está o erro, e sim mostra todos os erros causados na comparação feita com o DTD da linguagem WML. Por causa disso, praticamente todas as funcionalidades foram testadas inicialmente em HTML, sendo depois a apresentação convertida para WML.

As limitações também dificultaram muito o desenvolvimento, pois o tamanho reduzido das telas dos telefones celulares, a pouca banda de dados, a limitação de processamento e memória, a falta de conteúdo multimídia e o tamanho máximo de 1500 *bytes* para um arquivo WML fazem o trabalho de programação muito complicado.

A análise e o projeto do sistema foram realizados, tendo em vista o objetivo do sistema, mas sem esquecer a limitação da tecnologia. Sendo que o sistema foi modelado para ser o mais simples possível para o usuário, com o mínimo de entrada de dados do usuário, utilizando os recursos de páginas dinâmicas para facilitar a interface com o usuário.

Outro aspecto na fase da análise foi modelar o sistema para que o processo de compra fosse o mais simples possível, com a opção de busca pelo nome do produto, ou pelas ofertas ou listados por categoria. Esse aspecto é fundamental em um sistema de comércio eletrônico, principalmente no *M-commerce.* A interação com o sistema não estimula o usuário a navegar como na Web. Se o sistema de *M-commerce* não for simples e rápido na compra de um produto, certamente os clientes não realizarão a compra, pois o cliente utilizando um telefone

celular não tem o mesmo comportamento que um cliente utilizando um PC, e nem a mesma paciência.

Um problema detectado no desenvolvimento foi à diferença no comportamento e na apresentação do sistema nos diversos emuladores de telefones celulares utilizados nos testes. Principalmente em relação à apresentação, porque alguns telefones têm o tamanho de quatro linhas com 18 caracteres por linha; outros têm duas linhas com 15 caracteres por linha. Assim, uma página que fica perfeita em um determinado telefone, simplesmente fica problemática em outro. Apesar do WAP ser um padrão para o desenvolvimento de aplicativos móveis, cada fabricante implementa algumas funcionalidades a mais do que os outros, tornando assim o desenvolvimento mais complexo e com menos recursos, pois os desenvolvedores simplesmente não utilizam os recursos proprietários de um fabricante que não funciona com a maioria dos telefones móveis disponíveis.

Mesmo com todas essas limitações, é notório que a Internet Móvel vai revolucionar o comportamento das pessoas, pois o acesso à informação, a realização de transações bancárias, a realização de compras, pode ser realizada em qualquer lugar e a qualquer hora. Com o aumento da banda de dados e o suporte a multimídia como já ocorre no Japão, as aplicações se tornarão muito mais interessantes, e a convergência entre os computadores, PDA e telefones celulares se tornará realidade.

O modelo de programação WAP ainda tem muito a melhorar, principalmente em relação a linguagem WML e o suporte as características da rede de telefonia celular, pois existem somente algumas funcionalidades como discagem automática, manipulação da agenda eletrônica do telefone celular, sendo isso muito pouco em relação ao número de serviços que podem ser oferecidos, como localização do aparelho telefônico, identificação do aparelho e até informações sobre a utilização desses aparelhos quando eles estão acessando a rede de dados.

7 CONCLUSÃO

Com o surgimento da tecnologia para transmitir dados via *wireless* e com o avanço da popularização do acesso às redes de telefonia móvel, o acesso à Internet por meio desses dispositivos vai se tornar o principal meio de acesso a informações, serviços, e negócios na Internet.

O protocolo *WAP*, hoje na versão 1.2, é apenas o início de uma revolução. Mesmo com todas as limitações que ele ainda possui, com o acesso de no máximo 14.4 *Kbps*, os dispositivos pequenos, com problemas de segurança, ele disponibiliza serviços diferenciados, como o acesso a informações em qualquer lugar e a qualquer hora, possibilidades de negócios rápidos, etc. Mesmo sendo baseado no modelo *OSI*, o protocolo *WAP* tem problemas graves relacionados à segurança, porque quando utiliza os protocolos *WLTS* entre o dispositivo móvel e o *Gateway WAP*, e o protocolo *SSL/TLS* entre o *Gateway WAP* e o servidor na Internet, o *Gateway* tem que decodificar os *bytecodes* para converter o conteúdo em binário, existindo a possibilidade que pessoas não autorizadas tenham acesso à informação. Sendo que o *Gateway* converte o conteúdo primeiro para texto, para depois fazer a conversão em binário, e só depois codificar novamente a informação.

Outro problema que ainda precisa ser resolvido é a incompatibilidade de *browsers* no desenvolvimento, pois é muito difícil desenvolver em *WML*, porque cada fabricante possui um tipo de *browser WAP* diferente, ou às vezes, tem mais de um. Determinadas *tags* têm uma aparência em um *browser*, e uma aparência totalmente diferente em outro *browser*. Exemplos desses problemas são as *tags* <*input*> e <*select*>.

As limitações dos dispositivos móveis como telas pequenas, pouca qualidade do visor, teclado inapropriado para entrada de dados, tornam o desenvolvimento mais lento, e muito mais caro. Com isso, muitas aplicações WML simplesmente não são desenvolvidas em razão do seu alto custo.

Como todas as novas tecnologias, ela precisa melhorar ainda muito para se afirmar como o protocolo padrão da Internet Móvel, ou talvez venha a desaparecer, mesmo assim, o objetivo de mostrar para as pessoas um novo conceito de acesso à informação já foi atingido.

O *M-commerce*, um dos assuntos abordados no trabalho, ainda não é uma realidade aqui no Brasil, muito por falta de aplicações de qualidade nessa área, e também por problemas ainda não resolvidos como o custo de acesso à Internet Móvel, que hoje no Brasil praticamente só é utilizado pela elite da sociedade. Essas questões devem ser resolvidas logo, com a utilização do *GPRS* que trabalha com circuito de pacotes, e vai possibilitar ao usuário estar sempre conectado na Internet Móvel.

Os estudos de caso estudado mostram que as principais aplicações mais desenvolvidas são os serviços bancários, sites de informações diversas e poucas aplicações de *m-commerce*.

Em relação às tecnologias possíveis de se utilizar para criar um *site WAP*, segundo pesquisas realizadas, e também sobre o relato de pessoas que já estão desenvolvendo aplicações *WAP* há mais tempo, o ideal é criar aplicações leves, com o mínimo de entrada dos usuários. Para isso, devem ser utilizadas linguagens de script, como o PHP, o JSP, o PERL entre outras opções, que são muito utilizadas para desenvolver sites dinâmicos na Internet, fácil de aprender, não tem custo de licença e rodam em qualquer ambiente de computação.

Um dos principais objetivos desse trabalho é criar tecnologia que possa ser utilizada por outras pessoas para difundir a Internet móvel, e incentivar outras pessoas a criarem sites interessantes, utilizando as tecnologias disponíveis. O projeto E-Móvel tem como objetivo criar uma aplicação de *M-commerce* de qualidade, possibilitando assim ser utilizado como fonte de conhecimento para ser aplicada em outros sistemas. Foram utilizadas as técnicas mais avançadas com a integração de banco de dados, linguagem de script e a linguagem de marcação WML.

Uma das vantagens de utilizar software livre no desenvolvimento do sistema foi a grande quantidade de informação disponível sobre esses programas. Certamente, se fosse utilizada uma tecnologia proprietária, além de não ter a disponibilidade de tantos programas de qualidade, o acesso à informação seria muito mais complicado. Alguns programas, como a biblioteca *Mcrypt*, simplesmente não pode ser utilizada em sistemas Windows.

O E-Móvel foi desenvolvido utilizando as tecnologias *WML*, *PHP* e o bando de dados *MySQL*. Sendo que um dos objetivos foi utilizar as ferramentas que sejam de livre uso no mercado e com o código-fonte aberto, como o *Linux*, *MySQL* e o *PHP*, seguindo assim um novo padrão que está surgindo no mercado que é a independência de fornecedores de software, incentivando o desenvolvimento de software livre com qualidade. Principalmente em países como o Brasil, onde os órgãos públicos e privados gastam milhões em licenças de uso de software, mas grande parte da população nunca assistiu a TV, e mais da metade da população vive na miséria total.

Com a criação do protótipo foi possível identificar as dificuldades para implementar um site de *M-Commerce* com a tecnologia WAP atual. As limitações fazem o desenvolvimento de um sistema ficar mais complexo e causa o aumento no tempo e dinheiro gasto em um projeto. Porém, foram identificadas as grandes possibilidades de novos sistemas que podem ser criados para a Internet Móvel, como a cesso a informações de empresas, consulta a dados de serviços públicos, sistemas para empresas como seguradoras e transportadoras e também sistemas de comércio eletrônico.

O modelo de negócio implementado no protótipo é muito interessante, pois ele traz vantagens tanto para o comprador como para o site vendedor. Um dos grandes problemas identificados nas pesquisas realizadas em relação ao comércio eletrônico foi que, na maioria dos casos, os benefícios como a redução de custos nas transações e nos custos operacionais fica com o site da empresa que vende o produto, e o cliente que compra o produto não recebe parte desses benefícios. Com a compra em grupo, quanto mais clientes comprarem um determinado produto mais acessível ele fica e o site ganha mais porque o seu volume de venda aumenta.

A utilização do software livre no projeto possibilitou conhecer todo o potencial dos programas com licença GNU que tem o código-fonte aberto. Claro que ainda existem limitações, como as limitações encontradas no MySQL onde a função que mais fez falta no

desenvolvimento do E-Móvel foi o comando *Select* com um *Sub-Select* para fazer uma seleção. Mas com a grande colaboração dos programadores do mundo inteiro, os programas livres se tornam cada vez mais estáveis, seguros e fácil de utilizar, como ocorre com o Linux que vem se tornando uma grande alternativa ao sistema operacional dominante no mundo dos PCs. A utilização de software livre permite o aumento de conhecimento das tecnologias de ponta para diversas pessoas, pois atualmente concentrado em grandes empresas de software. Além disso, permite que os profissionais de ciência da computação se especializem mais e aumentem seus conhecimentos, pois o futuro em relação ao software é o valor agregado no serviço, e não a venda de licenças de software.

REFERÊNCIAS BIBLIOGRÁFICAS

1 ALBERTIN, Alberto Luiz. **Comércio Eletrônico**: modelo, aspectos e contribuições de sua aplicação. São Paulo: Editora Atlas, 2001. 276p.

2 ANSELMO, Fernando. **PHP e MySQL para Windows**. Florianópolis: Visual Books, 2000. 144p.

3 AREHART, Charles et al.. **Professional WAP**. United States of America: Editora Wrox Press Ltd, 2000. 813p.

4 CONVERSE, Tim; PARK Joyce. **PHP4**: a Bíblia. Tradução Edson Frumankiewicz, Joana Figueiredo. Rio de Janeiro: Editora Campus, 2001. 697p.

5 DIAS, Adilson de Souza. **WAP (*Wireless Application Protocol*) – A Internet Sem Fios**. Rio de Janeiro: Editora Ciência Moderna, 2000. 252p.

6 DORMAN, Andy. ***Wireless* Comunication**: o guia essencial de comunicação sem fio. Tradução Fábio Freitas. Rio de Janeiro: Editora Campus, 2001. 304p.

7 FORTA, Bem et al.. **Desenvolvendo WAP com WML e WMLScript**. Tradução Daniela Lacerda, Sônia Milione, Valéria Chamon. Rio de Janeiro: Editora Campus, 2000. 559p.

8 FÓRUM, Wireless Application Protocol . **Official Wireless Application Protocol**: The Complete Standart with Searchable CD-ROM. United States of America: Editora John Wiley & Sons, Inc., 1999. 744p.

9 MANN, Steve. **Programming Applications with the Wireless Application Protocol***:* the complete developer's guide. United States of America: Editora John Wiley & Sons, Inc., 2000. 226p.

10 MELONI, Julie C.. **Fundamentos de PHP**. Rio de Janeiro: Editora Ciência Moderna Ltda., 2000. 352p.

11 SCHNEIER, Bruce. **Applied Criptography Second Edition**: protocols, algorithms, and source code in C. United States of America: Editora John Wiley & Sons, Inc., 1996. 784p.

12 SCHNEIER, Bruce. **Segurança.com – Segredos e mentiras sobre a proteção na vida digital**. Tradução Daniel Vieira. Rio de Janeiro: Editora Campus, 2001. 401p.

13 STOCO, Lucio M.. **Integrando PHP com MySQL**. São Paulo: Novatec Editora, 2000. 96p.