

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO
(Bacharelado)

**EXPERIMENTOS COM A TECNOLOGIA WAP ATRAVÉS DE UM
PROTÓTIPO DE SISTEMA DE APOIO COMERCIAL**

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À UNIVERSIDADE
REGIONAL DE BLUMENAU PARA A OBTENÇÃO DOS CRÉDITOS NA
DISCIPLINA COM NOME EQUIVALENTE NO CURSO DE CIÊNCIAS DA
COMPUTAÇÃO — BACHARELADO

EVALDO CEVINSCKI JÚNIOR

BLUMENAU, NOVEMBRO/2001.

2001/2-20

**EXPERIMENTOS COM A TECNOLOGIA WAP ATRAVÉS DE UM
PROTÓTIPO DE SISTEMA DE APOIO COMERCIAL**

EVALDO CEVINSCKI JÚNIOR

ESTE TRABALHO DE CONCLUSÃO DE CURSO FOI JULGADO ADEQUADO
PARA OBTENÇÃO DOS CRÉDITOS NA DISCIPLINA DE TRABALHO DE
CONCLUSÃO DE CURSO OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO DE:

BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO

Prof. Francisco Adell Péricas — Orientador na FURB

Prof. José Roque Voltolini da Silva — Coordenador do TCC

BANCA EXAMINADORA

Prof. Francisco Adell Péricas

Prof. Sérgio Stringari

Prof. Dalton Solano dos Reis

DEDICATÓRIA

Dedico este trabalho aos meus pais, Evaldo Cevinski e Maria Ana Cevinski, por todo o amor, apoio, incentivo, compreensão e carinho que tiveram comigo durante mais esta etapa da minha vida.

AGRADECIMENTOS

Agradeço à Deus, por tudo que me proporcionou durante a minha vida.

Aos meus pais, Evaldo e Maria Ana, por acreditarem na minha capacidade e apostarem no meu potencial, e sempre me apoiando para chegar à condição de formando.

Aos meus irmãos, Márcia, Jorge, Ana, Marta, Adriana, Valter e demais familiares, por sempre me incentivarem a não desistir no meio do caminho.

Ao professor Francisco Adell Péricas, por me orientar para a realização deste trabalho, que sempre esteve presente para esclarecer dúvidas e elogiando cada passo dos meus estudos.

Aos amigos Jean Carlo Corrêa Gomes e Francis Pierre de França, que dividem apartamento comigo, e ao amigo Dante O. Machado, por sempre estarem presentes nos momentos bons e difíceis.

A todos os professores e colegas que tive durante o curso, dos quais irão permanecer na minha memória pelo resto da minha vida.

E a todos aqueles que de alguma forma contribuíram para a realização deste trabalho.

*“Jamais repita as mesmas
táticas que lhe deram uma
vitória, procure variá-las
segundo as circunstâncias.”*

Sun Tzu

SUMÁRIO

SUMÁRIO	VI
LISTA DE FIGURAS	VIII
LISTA DE QUADROS.....	IX
LISTA DE ABREVIATURAS.....	X
RESUMO.....	XII
ABSTRACT.....	XIII
1 INTRODUÇÃO	1
1.1 OBJETIVOS	2
1.2 ESTRUTURA DO TRABALHO.....	2
2 WAP	4
2.1 COMPONENTES DA ARQUITETURA WAP.....	7
2.1.1 WAE – <i>Wireless Application Environment</i> (CAMADA DE APLICAÇÃO).....	8
2.1.2 WSP – <i>Wireless Session Protocol</i> (CAMADA DE SESSÃO)	8
2.1.3 WTP – <i>Wireless Transaction Protocol</i> (CAMADA DE TRANSAÇÃO)	9
2.1.4 WTLS – <i>Wireless Transport Layer Security</i> (CAMADA DE SEGURANÇA)	12
2.1.5 WDP – <i>Wireless Datagram Protocol</i> (CAMADA DE TRANSPORTE)	13
2.2 SEGURANÇA UTILIZADA NO WAP.....	14
2.2.1 FUNCIONAMENTO DO SSL – SECURE SOCKET LAYER	15
2.3 LINGUAGENS WAP	16
2.3.1 WML.....	16
2.3.2 WMLSCRIPT	17
2.4 ASP E WML	17
3 DESENVOLVIMENTO DO PROTÓTIPO	19
3.1 FERRAMENTAS UTILIZADAS	19

3.1.1 MICROSOFT ACCESS.....	19
3.1.2 POWER DESIGNER	19
3.1.3 M3GATE	19
3.1.4 NOKIA WAP TOOLKIT	20
3.2 ESPECIFICAÇÃO DO PROTÓTIPO	21
3.3 IMPLEMENTAÇÃO E APRESENTAÇÃO DO PROTÓTIPO	24
4 CONSIDERAÇÕES FINAIS	32
4.1 CONCLUSÕES.....	32
4.2 SUGESTÕES	33
ANEXO 1.....	34
REFERÊNCIAS BIBLIOGRÁFICAS	42

LISTA DE FIGURAS

Figura 1 - Acesso a Internet na forma convencional.....	6
Figura 2 - Acesso à Internet através da tecnologia WAP	6
Figura 3 - Camadas da Arquitetura WAP	7
Figura 4 – Classe 0.....	10
Figura 5 – Classe 1.....	11
Figura 6 – Classe 2.....	11
Figura 7 – Segurança WAP	14
Figura 8 – Emulador M3Gate	20
Figura 9 – <i>Nokia Wap ToolKit</i>.....	21
Figura 10 – Diagrama de Contexto	22
Figura 11 – MER – Modelo Entidade Relacionamento	22
Figura 12 – Tabelas do Protótipo.....	23
Figura 13 – Macrofluxo do Menu	24
Figura 14 – Tela de Apresentação.....	25
Figura 15 – Menu Principal.....	26
Figura 16 – Cadastrar Pedido	27
Figura 17 – Tela Resultado do Pedido.....	29
Figura 18 – Tela Resultado da Consulta Produto	29
Figura 19 – Página Internet.....	31

LISTA DE QUADROS

Quadro 1 – Card <i>inicial</i>	26
Quadro 2 – Card <i>Menu</i>	26
Quadro 3 – Card <i>Inserir</i>	27
Quadro 4 – <i>Script</i> ASP	28
Quadro 5 – <i>Script</i> Consulta.....	30

LISTA DE ABREVIATURAS

ASP – *Active Server Pages*

HTML – *Hipertext Markup Language*

HTTP – *Hipertext Markup Language*

HTTP – *Hipertext Transport Protocol*

PDA - *Personal Digital Assitent*

RAM – *Random Access Memory*

ROM – *Random Only Memory*

SSL – *Secure Sockets Layer*

TCP – *Transmission Control Protocol*

UI – *User Interface*

WAE – *Application Layer*

WAP – *Wireless Application Protocol*

WDP – *Wireless Transport Protocol*

WML – *Wireless Markup Language*

WMLS – *Wireless Markup Language Script*

WSP – *Wireless Session Protocol*

WTA – *Wireless Telephony Application*

WTLS – *Wireless Transport Layer Security*

WTP – *Wireless Transaction Protocol*

WWW – *Word Wide Web*

XML – *eXtensible Markup Language*

RESUMO

Este trabalho apresenta experimentos com a tecnologia WAP (*Wireless Application Protocol*), da qual serão abordados seus aspectos teóricos, práticos, considerações sobre as linguagens de programação WML (*Wireless Markup Language*) e WMLS (*Wireless Markup Language Script*) que são linguagens específicas para sistemas que utilizam a tecnologia WAP, bem como a linguagem ASP (*Active Server Pages*) para acesso a bancos de dados. O trabalho inclui a especificação e a implementação de um protótipo de um sistema de apoio comercial para dispositivos sem fio baseado nesta tecnologia.

ABSTRACT

This work presents experiments with technology WAP (Wireless Application Protocol), of which aspects will be boarded its theoretical, practical, considerations on programming languages WML (Wireless Markup Language) and WMLS (Wireless Markup Language Script) that they are specific languages for systems that use technology WAP, as well as language ASP (Active Server Pages) for access the data bases. The work includes the specification and the implementation of an archetype of a system of commercial bracket for devices without wire based on this technology.

1 INTRODUÇÃO

A internet vem crescendo cada vez mais, sendo uma área de atualização com constantes novidades, e de muito investimento por grandes empresas (Dias, 2000).

É fato que os usuários tornam-se cada vez mais dependentes dos serviços oferecidos via Internet, mas para acessá-los se faz necessário que eles estejam conectados à rede através de um fio. Entretanto, milhões de usuários passam muito tempo em trânsito e o fato de necessitarem de um cabo para a conexão torna-se um empecilho (Silva, 2000).

Segundo Henkel (2001), num tempo não muito além dos dias atuais, possuir um aparelho de telefone celular era um fator de status para as pessoas. Somente uma fatia muito pequena da população, a qual fosse caracterizada por seu poder aquisitivo elevado, teria condições de exibir esses “aparelhinhos” móveis. Aliado a essa situação, surgia o advento das redes de computadores livres dos fios que os interconectavam. Era a chegada de uma inovação na maneira de se comunicar, deixando mais livre a localização desses terminais de acesso à rede.

Conforme Silva (2000), nos últimos anos, as tentativas de acabar com esta barreira e transformar a Internet em uma plataforma de serviços sem fio não foram bem sucedidas, pois o leque de padrões era muito extenso. O WAP (*Wireless Application Protocol*, ou Protocolo para Aplicações Sem Fio) surgiu como uma promessa de ser um protocolo largamente aceito, com capacidade de reconhecer os serviços WWW (*World Wide Web*), além de oferecer serviços diversos como telefônicos e mensagens eletrônicas.

Mas agora, além de se disseminar, a telefonia móvel quer comemorar o rito de passagem para a sua maioria por meio da oferta de serviços que até algum tempo poucos suspeitavam que um dia seria transportado para a tela de aparelhos compactos. De certa forma, um dos primeiros passos nesse sentido já vem sendo dado por meio de tecnologias como o WAP, que capacitam esses terminais a entrar no mundo da transmissão de dados (Henkel, 2001).

Em 26 de junho de 1997, a Ericsson, a Motorola, a Nokia e a Unwired Planet (hoje Phone.com) iniciaram a criação de um padrão para tornar os serviços avançados dentro do mercado sem fio uma realidade. Este novo padrão foi chamado WAP (*Wireless Application Protocol* – Protocolo para Aplicação Sem Fio). Em dezembro do mesmo ano, o WAPForum foi oficialmente criado e a primeira versão das especificações do protocolo foi disponibilizada em abril de 1998 (Silva, 2000).

O presente trabalho visa o desenvolvimento de um protótipo utilizando a tecnologia WAP, que permite que os representantes comerciais de uma empresa transmitam os pedidos para um servidor, através de um celular com acesso a WAP. O protótipo permitirá que os representantes de uma empresa acessem o seu *WapSite* e entrem com os dados para o pedido, que será transmitido para um servidor, ao qual a empresa terá acesso. Para tanto, foi utilizada no desenvolvimento do trabalho a linguagem WML (*Wireless Markup Language*) para o desenvolvimento do *WapSite*, a linguagem ASP (*Active Server Pages*) para fazer o intercâmbio com a base de dados, o *Microsoft Access* como ferramenta de banco de dados, e também a linguagem HTML (*Hipertext Markup Language*) para o desenvolvimento da *homepage*, onde a empresa poderá acessar os pedidos.

1.1 OBJETIVOS

O trabalho tem por objetivo realizar um estudo sobre a tecnologia WAP, resultando na especificação e implementação de um protótipo de um sistema de apoio comercial, que disponibilizará aos representantes comerciais de uma empresa uma forma mais eficiente de enviar os pedidos.

1.2 ESTRUTURA DO TRABALHO

Este trabalho foi estruturado em quatro capítulos de maneira a apresentar a introdução, os objetivos do trabalho e a sua estrutura no primeiro capítulo.

O segundo capítulo traz as fundamentações e conceituações sobre as tecnologias que foram utilizadas no desenvolvimento deste trabalho, tais como WAP e sua arquitetura, WML, WMLScript e ASP.

No terceiro capítulo são apresentados a especificação do protótipo, seu desenvolvimento e a demonstração da sua funcionalidade, e também as respectivas ferramentas utilizadas.

No quarto capítulo são feitas as considerações finais sobre o trabalho incluindo as conclusões e as sugestões para trabalhos futuros.

Finalmente no anexo 1 estão todos os códigos fontes (WML e ASP) do protótipo desenvolvido neste trabalho.

2 WAP

A possibilidade de se acessar a Internet através de um telefone celular ou de um PDA (*Personal Digital Assitent*) já está presente. Vários são os recursos disponíveis, ou seja, aparelhos celulares capazes de exibir páginas Web. Essa “Internet de bolso” vem revolucionando o mercado ocasionando a união das empresas da área comercial, tecnológica e de telecomunicações para criação de uma rede que alcance tanto usuários já acostumados com a Internet tradicional como aqueles que ainda nem possuem computador ou nunca acessaram a rede (Henkel, 2001).

Segundo Demétrio (2000), WAP é uma especificação para um conjunto de protocolos de comunicação com o intuito de normalizar a forma como os dispositivos sem fio (tais como: telefones celulares, palm tops, emissores/receptores de rádio, etc.) acessam a Internet.

Sob o ambiente evolutivo que está acontecendo, tem-se a tecnologia WAP que, segundo Henkel (2001) é definida como: “WAP provê um método para comunicação através de redes sem fio rapidamente, com segurança e eficiência. A comunicação pode tomar lugar utilizando os dispositivos de telefones celulares, *paggers*, rádios e PDA’s. Estas comunicações não estão limitadas a páginas estáticas como Internet fora antes, mas sim, oferece a oportunidade de integrar bancos de dados, conteúdo dinâmico e comércio eletrônico trafegando via dispositivo WAP”.

Atualmente a Internet é amplamente utilizada por usuários de computadores que possuem alto poder de processamento, grande quantidade de memória, alta largura de banda e transmissão de dados em redes geralmente confiáveis. Até então todas as tecnologias criadas para acesso à Internet eram destinadas a esse tipo de plataforma, mas no caso de aparelhos menos “robustos” como handhelds, telefones celulares e outros era necessário criar uma forma de acesso que considerasse algumas características particulares desses equipamentos, como limitado poder de processamento, pouca memória (RAM e ROM), consumo restrito de energia, pequenas telas, diferentes dispositivos de entrada de dados (como o teclado de um telefone celular ou de um handheld), estreita largura de

banda, maior latência na transmissão de dados e pouca estabilidade durante a conexão, foi pensando em um ambiente desse tipo que surgiu o WAP (Cani, 2000).

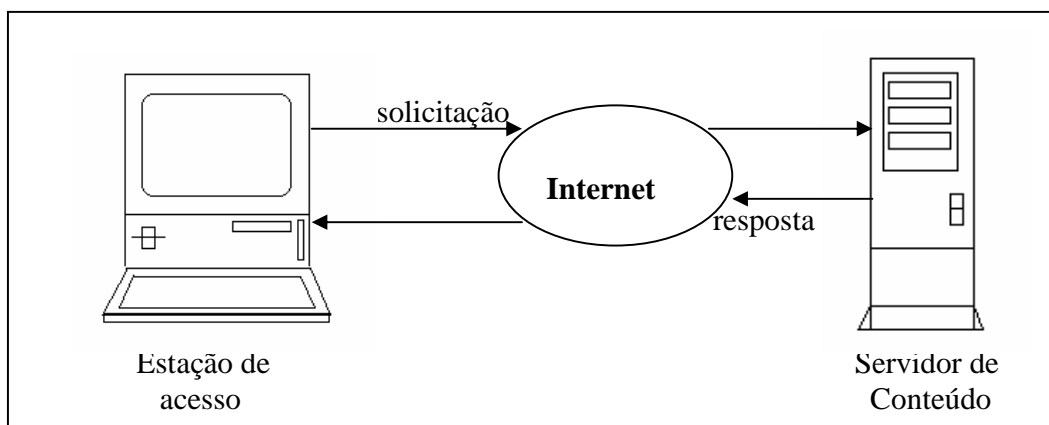
O modelo WAP está muito próximo do modelo da Internet, uma vez que ele oferece um número de características úteis para a comunicação sem fio. As pessoas que desenvolvem aplicativos WAP baseiam seus projetos nas idéias e padrões existentes, acarretando a implementação de serviços e produtos visando à transferência de dados. Por isso, muitas situações existentes estão sendo modificadas para fazer uso dessa tecnologia (Henkel, 2001).

A comunicação entre dispositivos sem fio é bastante lenta (algo em torno de 14,4 Kbps), forçando os desenvolvedores WAP a otimizarem os protocolos existentes tais como HTTP (*HiperText Transport Protocol*) e o TCP (*Transmission Control Protocol*) de maneira a melhorar a transmissão dos dados utilizando pouca largura de banda. Entretanto, a nova geração de telefones celulares padroniza velocidade de até 2 Mbps (Carvalho, 2001).

Outro aspecto a ser considerado é que o WAP pode ser utilizado em qualquer sistema operacional, incluindo plataformas populares de handhelds como Palm OS, Windows CE e Java OS. Possui também compatibilidade com o Microsoft Windows 95/98/NT, Linux, Solaris, etc. Isso ocorre porque o protocolo é independente de padrões de comunicação, em vez de ser baseado nas plataformas existentes. Dessa forma, qualquer plataforma que é capaz de implementar os padrões de comunicação é compatível com o WAP (WapBrasil, 2001).

As figuras 1 e 2 abaixo mostram respectivamente um comparativo da forma básica entre o modo de acesso dos computadores convencionais (computadores de mesa e *notebooks*) e o acesso dos dispositivos sem fio através da tecnologia WAP.

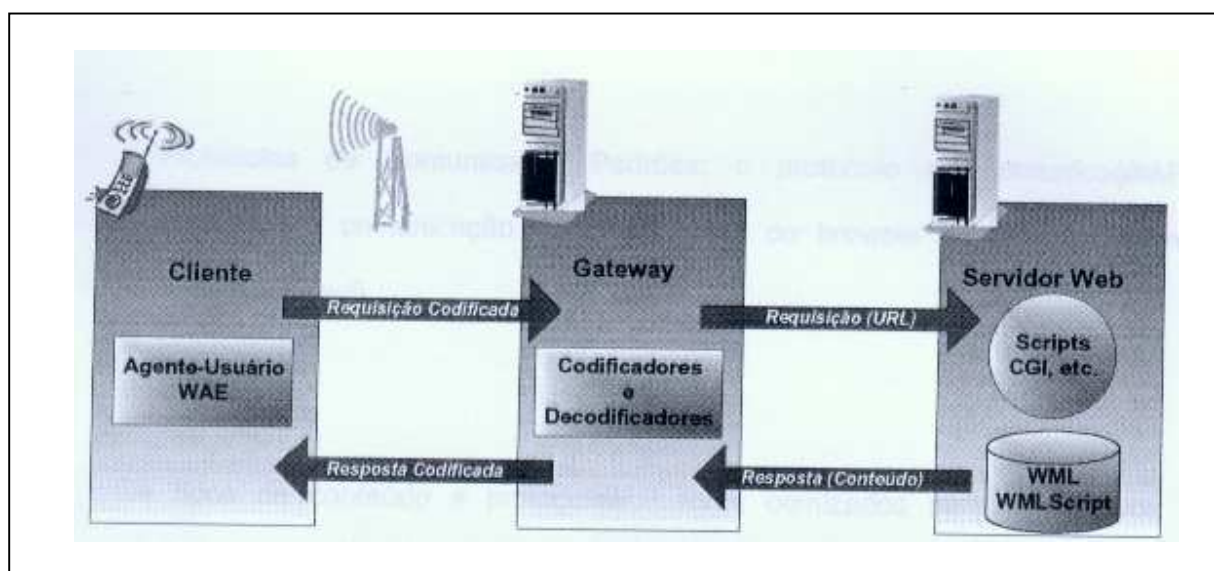
Figura 1 - Acesso a Internet na forma convencional



Fonte: adaptado de (Cani, 2000).

Na forma convencional uma solicitação é enviada de uma estação de acesso até um servidor de conteúdo através da Internet. O servidor interpreta a solicitação e envia a resposta à estação que a solicitou, onde a mesma será recebida (Cani, 2000).

Figura 2 - Acesso à Internet através da tecnologia WAP



Fonte: (Silva, 2000).

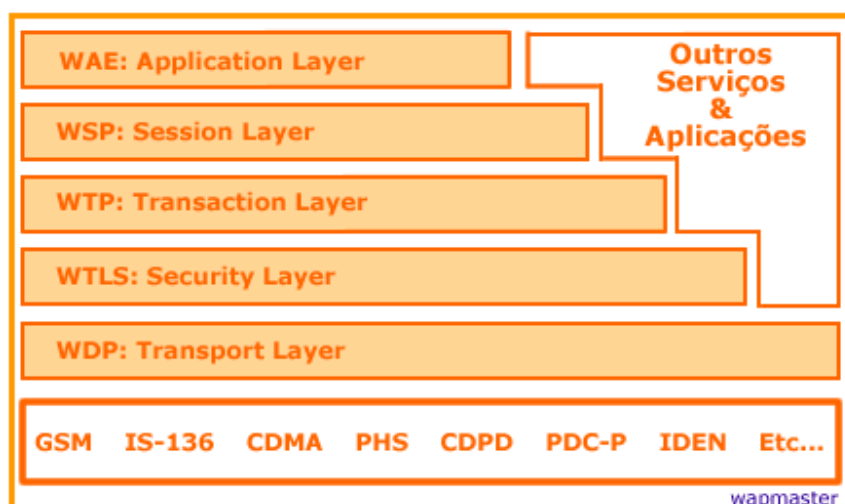
No caso do WAP, o dispositivo envia uma solicitação através da rede sem fio para um *gateway* WAP. O *gateway* converte a requisição para o protocolo da Internet e a envia para o servidor de conteúdo. O servidor de conteúdo envia uma resposta para o *gateway* onde a mesma é convertida e codificada, sendo enviada através da rede sem fio até o dispositivo que originou a solicitação, onde será exibida (Cani, 2000).

Devido ao WAP ter sido projetado em cima de uma banda de comunicação menor do que os demais protocolos Internet (TCP/IP), é feita uma codificação no *gateway* para compactação das informações que são decodificadas apenas no dispositivo que requisitou a informação ou serviço, dessa maneira a transmissão alcança uma melhor desempenho (Cani, 2000).

2.1 COMPONENTES DA ARQUITETURA WAP

A arquitetura é formada por um ambiente escalável e extensivo para o desenvolvimento de aplicações para dispositivos de comunicação móveis. Isto se realiza por meio do protocolo WAP, que é composto por várias camadas de subprotocolos (figura 3), em que cada camada pode ser acessada pela camada superior e por outros serviços e aplicações (Demétrio, 2000).

Figura 3 - Camadas da Arquitetura WAP



Fonte: (WAPMAster2001)

A arquitetura de camadas do WAP permite a outros serviços e aplicações utilizar as funções da pilha de camadas WAP por meio de interfaces bem definidas. Ainda, as aplicações externas podem acessar diretamente as camadas de sessão, transação, segurança e transporte (Demétrio, 2000).

2.1.1 WAE – *Wireless Application Environment* (CAMADA DE APLICAÇÃO)

O WAE é a camada de aplicação de propósito geral baseada na combinação das tecnologias WWW e de telefonia móvel. Seu principal objetivo é estabelecer uma interação entre os operadores e os provedores de serviços, para permitir a construção de aplicações de uma forma eficiente, independente do dispositivo em que vão ser executadas (celulares, palms, etc.). O WAE inclui um micro-navegador com as seguintes funções (Demétrio, 2000):

- a) WML (*Wireless Markup Language*), uma linguagem leve, otimizada, semelhante a HTML, porém específica para uso em dispositivos móveis;
- b) WMLScript, uma linguagem similar ao JavaScript que otimiza a WML;
- c) serviços de telefonia e interfaces de programação WTA (*Wireless Telephony Application*);
- d) suporte a imagens, calendários, agenda, etc.

O WAE não especifica formalmente nenhum agente-usuário, ou seja, suas características e capacidades são deixadas para os implementadores. O WAE somente define serviços e formatos fundamentais que são necessários para assegurar a interoperabilidade entre diferentes implementações.

2.1.2 WSP – *Wireless Session Protocol* (CAMADA DE SESSÃO)

O WSP permite a aplicação de uma interface para o estabelecimento de sessões. Para isso foram criados dois tipos de protocolos diferentes (Demétrio, 2000):

- a) o primeiro para a conexão mediante o serviço de transações WTP (*Wireless Transaction Protocol*);

b) e o outro permite acessar diretamente a camada WDP (*Wireless Datagram Protocol*), sem a necessidade de estabelecer uma conexão direta com as outras camadas, o que melhora o rendimento de aplicações que não necessitam de confirmação de envio de dados. Permite a troca eficiente de dados entre as aplicações.

A diferença básica é que nos serviços orientados à conexão são disponibilizadas facilidades de gerenciamento e segurança na transmissão entre o cliente (*browser WAP*) e o servidor (*gateway WAP*). Caso por algum motivo torne impossível a transmissão, a mesma pode ser interrompida e retomada mais tarde partindo do ponto de parada anterior. Existe um rígido controle entre pacotes enviados e pacotes recebidos, esse controle obviamente gera um tráfego um pouco maior do que nos serviços sem conexão. Já no caso dos serviços sem conexão não existe nenhum controle de recebimento dos pacotes enviados pelo dispositivo, ou seja, os serviços não são confirmados. Nesse caso a sessão de transmissão não é considerada segura, sua principal vantagem é que gera menos tráfego na rede (Cani, 2000).

O núcleo do WSP é um formato binário utilizado pelo HTTP, o que permite a inclusão de cabeçalhos e informação. Dessa forma, os métodos utilizados pelo HTTP/1.1 são suportados garantindo compatibilidade com o mesmo. Convém salientar que o WSP, por si só, não interpreta a informação do cabeçalho nas requisições/respostas e, também, o seu ciclo de vida não está ligado ao transporte, uma vez que a sessão pode ser suspensa quando estiver ociosa, para fins de liberação de recursos na rede ou economia de bateria do dispositivo (Henkel, 2001).

2.1.3 WTP – *Wireless Transaction Protocol* (CAMADA DE TRANSAÇÃO)

É um protocolo simplificado, idealizado para situações de baixa largura de banda, caso específico das comunicações sem fio (Demétrio, 2000).

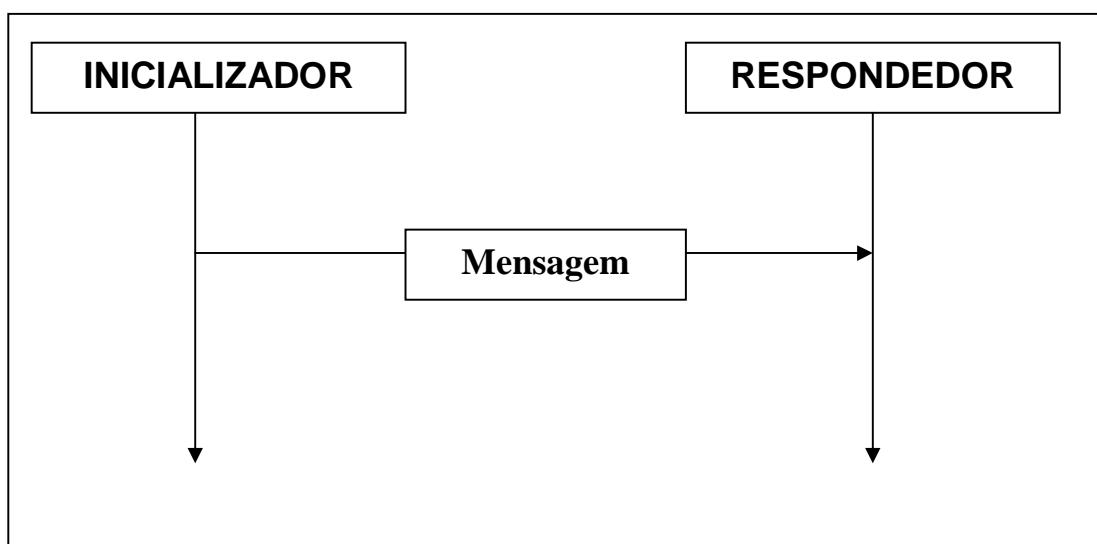
A camada WTP da pilha de protocolos WAP é a responsável por controlar o envio e a recepção das mensagens, oferecendo suporte de transações, o que acrescenta confiança ao

serviço de datagrama provido pela camada WDP, uma vez que libera a camada superior a retransmissões e reconhecimentos necessários quando esse serviço é utilizado (Henkel, 2001).

Quando uma transação WTP se inicia, ela é definida como sendo o Inicializador, enquanto que a resposta à mesma é indicada como sendo do Responder. Uma classe de transação é então criada pelo Inicializador e indicada na mensagem de execução a ser enviada ao Responder. Existem três classes de transação que são (Henkel, 2001):

- a) classe 0: o Inicializador envia uma mensagem para o Responder. A mensagem de execução não é confiável, portanto não possui mensagem de resultado (figura 4);

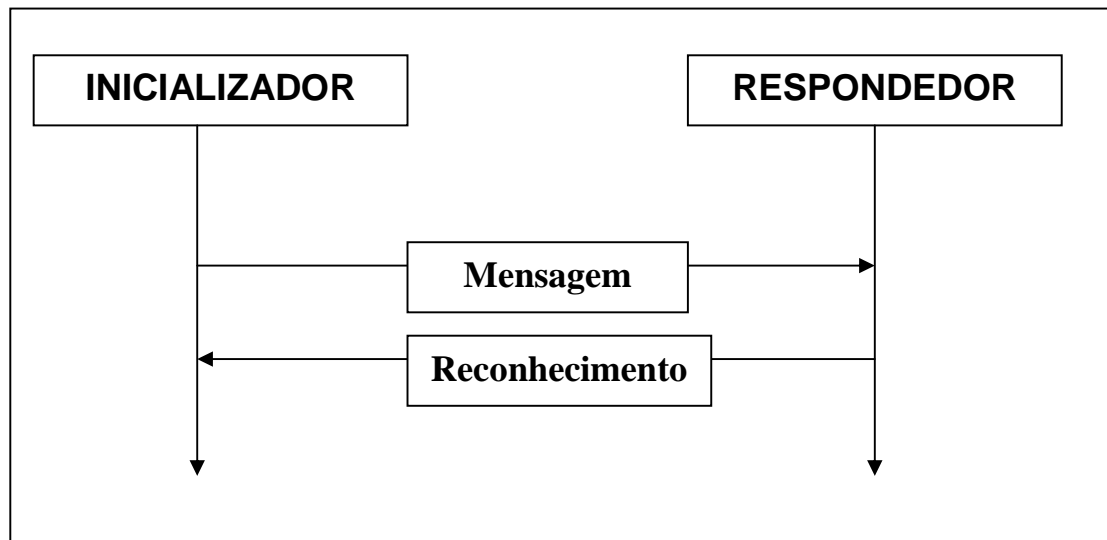
Figura 4 – Classe 0



Fonte: Adaptado de (Cani, 2000).

- b) classe 1: a mensagem de execução é confiável, porém, não possui mensagem de resultado. O Inicializador envia uma mensagem para o Responder, após o recebimento da mensagem pelo Responder o mesmo retorna para o Inicializador um sinal de reconhecimento da mensagem conforme segue abaixo (figura 5);

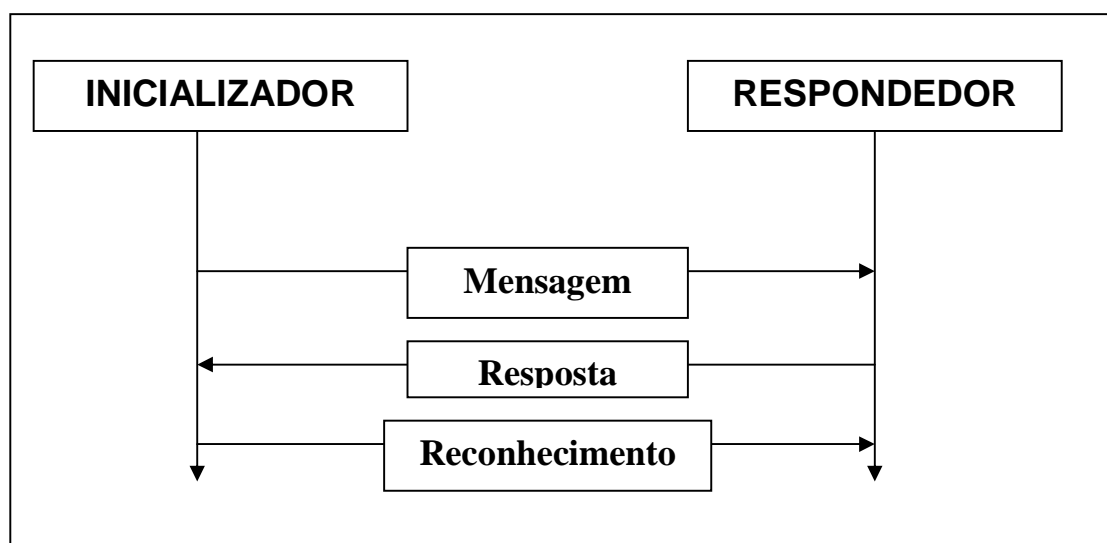
Figura 5 – Classe 1



Fonte: Adaptado de (Cani, 2000).

- c) classe 2: a mensagem de execução é confiável, mas com somente uma mensagem de resultado confiável. O Inicializador manda uma mensagem para o Respondedor e após o recebimento da mensagem pelo Respondedor, o mesmo faz o processamento da mensagem e envia o resultado para o Inicializador. Ao receber o resultado o Inicializador envia ao Respondedor um sinal de reconhecimento (figura 6).

Figura 6 – Classe 2



O protocolo possui várias características, tais como (Silva, 2000):

- a) o alcance da confiabilidade é feito através do uso de identificadores únicos de transação, reconhecimentos, remoção de duplicação e retransmissões.
- b) não possui fase de criação e destruição de conexão explícita, uma vez que este procedimento de abertura e/ou fechamento de conexão impõe uma excessiva sobrecarga na rede de comunicação.
- c) possui segurança de usuário-a-usuário opcional, isto é, o usuário do WTP pode confirmar cada mensagem recebida.
- d) faculta ao último reconhecimento da transação a contenção de informação fora da banda relacionada à transação, como medidas de desempenho.
- e) usa a concatenação para transportar múltiplas unidades de dados do protocolo em uma única unidade de dados de serviço, que será transportada pelo WDP.
- f) baseia-se na mensagem, isto é, a unidade básica de intercâmbio é uma mensagem inteira e não uma cadeia de bytes.
- g) usa mecanismos para minimizar o número de transações sendo retornadas como resultado de pacotes duplicados.
- h) possibilita o cancelamento de transações atual, incluindo o descarte dos dados não enviados tanto no cliente quanto no servidor. O cancelamento pode ser disparado pelo usuário para cancelar um serviço requisitado.
- i) dá segurança através de mensagem de execução confiável. Tanto o sucesso quanto a falha são reportados, isto é, se uma execução não pode ser manipulada pelo Responder, então uma mensagem de cancelamento será retornada para o Inicializador ao invés do resultado.
- j) permite transações assíncronas, isto é, o Responder envia o resultado assim que os dados se tornarem disponíveis.

2.1.4 WTLS – *Wireless Transport Layer Security* (CAMADA DE SEGURANÇA)

O protocolo da camada de segurança na arquitetura WAP provê para as camadas superiores um serviço de interface de transporte seguro, preservando o serviço de interface

de transporte abaixo dele. A camada WTLS é modular, isto é, o seu uso não depende do nível de segurança requerido em uma dada aplicação, ou seja, é uma camada opcional da pilha WAP. Além da segurança, o WTLS fornece uma interface para gerenciamento de conexões seguras (Silva, 2000).

O WTLS disponibiliza os seguintes serviços:

- a) integridade de dados: garante que não haverá alteração do conteúdo nos dados enviados entre o cliente e o servidor, e vice-versa;
- b) privacidade: assegura que não haverá como um interceptador descriptar a mensagem, a não ser o transmissor ou receptor;
- c) autenticação do cliente: limita o acesso através do servidor de conteúdo, ou seja, apenas quem tem permissão pode acessar determinado conteúdo;
- d) autenticação do servidor: certifica que o servidor acessado, é realmente o desejado.

Aplicações são capazes de habilitar ou desabilitar características do WTLS dependendo das suas necessidades de segurança ou das características da rede. Por exemplo, privacidade pode ser desabilitada em redes que já fornecem esse serviço em uma camada de mais baixo nível (Jesus, 2000).

2.1.5 WDP – *Wireless Datagram Protocol* (CAMADA DE TRANSPORTE)

É a camada que transporta os dados, responsável pelo envio e recebimento de mensagens pelos vários tipos de redes. Como o protocolo WDP proporciona uma interface entre os protocolos superiores WAE, WSP e WTLS, é capaz de funcionar de forma independente da rede telefônica em que esteja trabalhando, ou de se adaptar às especificações desta rede (Demétrio, 2000 e Dias, 2000).

Os serviços oferecidos pelo WDP incluem endereçamento de aplicação por número de porta e segmentação opcional, além de reunião e detecção de erros opcionais. Estes serviços permitem às aplicações operarem transparentemente sobre diferentes transportadores disponíveis. O número da porta identifica a entidade mais acima do WDP

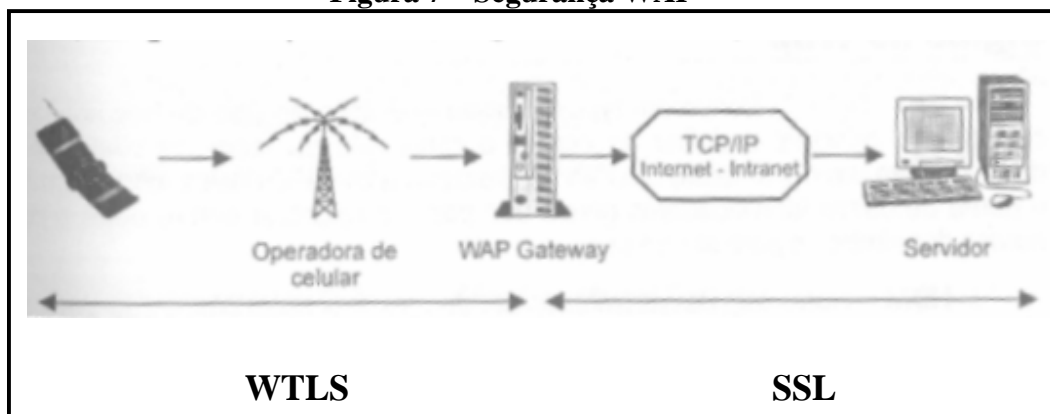
que pode ser o WTP ou WSP ou mesmo uma aplicação como o correio eletrônico (Silva, 2000).

2.2 SEGURANÇA UTILIZADA NO WAP

Com o aumento significativo de dispositivos, aplicações e serviços *wireless* surge a preocupação com a questão da segurança, ou seja, até que ponto o acesso é seguro e livre de intrusos. Nas aplicações de comércio eletrônico via telefones celulares, PDA's ou *Pagers* com acesso à Internet, essa questão é muito mais importante do que no modelo convencional via computador *desktop*. Isso acontece porque se deve garantir a privacidade, a confiabilidade e a qualidade nas operações realizadas (Henkel, 2001).

Como se pode observar na figura 7 abaixo, existem duas partes diferenciadas no modelo de segurança WAP. Na parte direita da figura, o *Gateway* WAP simplesmente usa SSL (*Secure Socket Layer*) para estabelecer uma comunicação segura com o servidor Web, assegurando a privacidade, integridade e autenticação do servidor.

Figura 7 – Segurança WAP



Fonte: (Demétrio, 2000)

Na parte esquerda, o *Gateway* apanha os dados codificados com SSL do servidor Web e os transforma para poder transmitir, usando WAP e o protocolo de segurança WTLS. As solicitações desde o telefone até o servidor Web refazem o caminho inverso. Em resumo, o *Gateway* faz o papel de ponte entre os protocolos WTLS e SSL (Demétrio, 2000).

2.2.1 FUNCIONAMENTO DO SSL – SECURE SOCKET LAYER

Segundo Demétrio (2000), o elemento-chave do protocolo SSL é um código de encriptação público. Para isto usa-se um par de códigos e algoritmos matemáticos que convertem o texto em texto codificado e vice-versa. Esse par de códigos consta de um código registrado público e outro privado, que é mantido em segredo pelo proprietário. Uma mensagem codificada com o código público só pode ser decodificada com o código privado. O inverso é verdadeiro. Assim como uma porta que só pode ser aberta com duas chaves.

A encriptação de código público é muito útil para pequenas quantidades de dados, mas torna-se lenta quando se utilizam grandes quantidades. Para o último caso usa-se a encriptação privada ou simétrica.

O protocolo SSL combina ambas as técnicas para realizar as transações. Em um primeiro contato, conhecido como negociação, o remetente e o destinatário vão se comunicar utilizando o código público, o que assegurará a privacidade durante o resto da transação.

Para proporcionar a integridade, o protocolo SSL usa algoritmos que criam uma assinatura digital matemática de cada mensagem. Se a mensagem modificar-se pelo caminho, o destinatário comprovará que a assinatura digital não coincide com a mensagem e a rejeitará.

A autenticidade de ambas as partes é confirmada por intermédio de certificados digitais, que evitam que terceiros interceptem as transmissões. Os certificados asseguram que remetente e destinatário são quem dizem ser.

Quando o navegador solicita uma conexão com o servidor, ele mostra o seu certificado. O navegador comprova se o certificado é válido, e caso positivo, prossegue com a transação, seguindo então para codificação/decodificação dos dados (Demétrio, 2000).

2.3 LINGUAGENS WAP

Foram criadas duas linguagens de programação específicas para o WAP parecidas com as linguagens atuais para a Internet fixa.

A primeira é a WML (*Wireless Mark-up Language*), baseada na linguagem XML (*Extensible Mark-up Language*), constituída por um sub-conjunto de funcionalidades do HTML.

A outra foi desenvolvida para a criação de páginas dinâmicas: é o WMLScript, parecido com o JavaScript (Jesus, 2000).

2.3.1 WML

A geração de conteúdo para a Internet está, fundamentalmente, baseada na construção de páginas Web através da linguagem HTML. Já na tecnologia WAP não poderia ser diferente, porém é utilizada uma linguagem com sintaxe bastante semelhante à do HTML, ou seja, a linguagem XML. As páginas WAP podem ser exibidas num *microbrowser* incorporado aos dispositivos de telefonia celular (Henkel, 2001).

Segundo Henkel (2001), ao contrário da estrutura plana de documentos HTML, documentos em WML são divididos em um conjunto de unidades bem definidas para interações com o usuário. Uma unidade de interação é chamada de cartão (*card*), e os serviços são criados deixando o usuário navegar de um lado para outro entre cartões (*cards*) de um ou vários documentos WML (*decks*).

Dentre as funções disponibilizadas pelo WML, existem quatro principais (Demétrio, 2000):

- a) apresentação de texto e layout: WML inclui texto e suporte a imagens (wbmp);
- b) organização em cartões e *decks*: toda a informação em WML é organizada em uma coleção de cartões e *decks*. Os cartões especificam uma ou mais unidades de interação do usuário (um menu de seleção, uma tela de texto ou um campo de entrada de texto). Logicamente o usuário navega por uma série de cartões WML, examinando os conteúdos, entrando dados, fazendo escolhas e movimentando-se entre cartões;

- c) agrupamento de cartões dentro de *decks*. Um *deck* é semelhante a uma página HTML, que é identificado por uma URL;
- d) navegação e links entre cartões: inclui um gerenciamento de navegação entre os cartões e *decks*. Também suporta links ancorados na página, similar ao HTML.

Uma outra questão a ser observada é que o WML é uma linguagem sensível há caracteres digitados em letra maiúscula ou minúscula, isso significa que, por exemplo, o comando “input” é diferente de “INPUT”. Por este motivo é aconselhado utilizar apenas caracteres em minúsculo para desenvolvimento de aplicações (Jesus, 2000).

2.3.2 WMLSCRIPT

A linguagem WMLScript (*Wireless Mark-up Language Script*) vem suprir as deficiências de processamento do WML, como ocorre com o JavaScript e o HTML, tornando as páginas em WML mais robustas (Demétrio, 2000).

O WMLScript é uma linguagem de *script* leve e procedural, que realça as facilidades de navegação e apresentação do WML, além de possuir capacidades de suporte ao comportamento da UI (*User Interface* – Interface do Usuário) de modo mais avançado, adicionar inteligência ao cliente, prover um mecanismo conveniente de acesso aos dispositivos e seus periféricos e reduzir a necessidade de idas e voltas ao servidor (Silva, 2000).

2.4 ASP E WML

O ASP (*Active Server Pages*) é um dos recursos da programação que viabiliza para o desenvolvedor criar páginas dinâmicas, ou seja, páginas com conteúdo variável (Weissinger, 1999).

Segundo Raduenz (2000) o ASP é um ambiente de programação interpretado por *scripts* no servidor, que se pode usar para criar páginas dinâmicas, interativas e de alta performance. Com as páginas ASP, os *scripts* são executados no servidor e não no cliente. Quando o cliente solicita a página, o servidor interpreta seu conteúdo e devolve somente o resultado.

Especificamente para o caso do WML, também se pode usar o ASP para criar páginas dinâmicas.

3 DESENVOLVIMENTO DO PROTÓTIPO

Neste capítulo são apresentados a especificação do protótipo, seu desenvolvimento e a demonstração da sua funcionalidade, e também as respectivas ferramentas utilizadas.

3.1 FERRAMENTAS UTILIZADAS

Para o desenvolvimento desse trabalho foram utilizadas várias ferramentas, descritas abaixo.

3.1.1 MICROSOFT ACCESS

O ACCESS é um sistema de gerenciamento de banco de dados relacional produzido pela Microsoft, que oferece facilidades na organização e armazenamento de dados que se relacionam entre si e que vem seguindo a tendência mundial de aplicativos, segundo Galante (1994), em apresentar flexibilidade e amigabilidade ao usuário.

3.1.2 POWER DESIGNER

É uma ferramenta Case utilizada para especificação de projetos de software. Tem-se dois módulos: um que permite desenvolver diagramas de fluxo de dados e o outro desenvolver o modelo entidade relacionamento do banco de dados. Pode-se utiliza-lo para criar novas estruturas de dados ou fazer engenharia reversa das estruturas de dados já existentes.

3.1.3 M3GATE

Este emulador foi desenvolvido pela empresa alemã *Mobile Media Mode Gate*, daí o nome M3Gate. É um programa *freeware*, de fácil utilização e ocupa pouco espaço em disco, em torno de 1,4 Mb.

O M3Gate (figura 8) é um ótimo emulador *off-line* (não precisa estar conectado). No desenvolvimento do protótipo mostrou-se bastante eficiente.

Figura 8 – Emulador M3Gate

3.1.4 NOKIA WAP TOOLKIT

O *Nokia Wap ToolKit v.3.0*, é uma ferramenta integrada de desenvolvimento, simulação e demonstração, do fabricante *Nokia*, para o Windows 98. Na figura 9 é apresentado o seu ambiente de interação com o desenvolvedor: os códigos são escritos no editor que se encontra a esquerda e a direita está o dispositivo WAP, onde é feita toda simulação e testes após o código ter sido compilado.

Figura 9 – Nokia Wap Toolkit

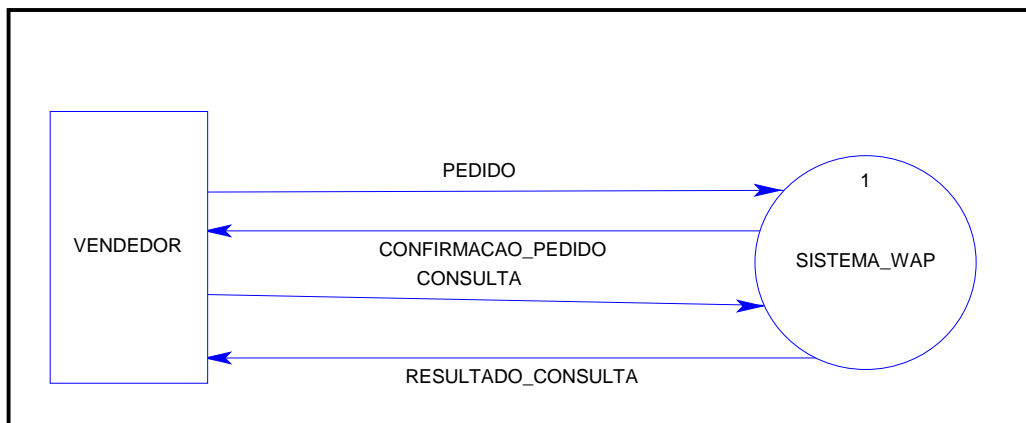
```
<?xml version="1.0"?>  
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">  
  
<wml>  
  <template>  
    <do type="prev">  
      <noop/>  
    </do>  
  </template>  
  
  <card id="init" newcontext="true">  
    <do type="options" label="Copyright">  
      <go href="#copyright"/>  
    </do>  
    <p align="center">  
        
      <br/>  
      <b>Mobile Internet Toolkit 3.0</b>  
    </p>  
  </card>  
  
  <card id="copyright" title="Copyright">  
    <onenter type="ontimer">  
      <prev/>  
    </onenter>  
    <timer value="25"/>  
    <p align="center">&nbsp;<br/>
```

3.2 ESPECIFICAÇÃO DO PROTÓTIPO

A metodologia utilizada para o desenvolvimento da especificação do protótipo foi à análise estruturada (Yourdon, 1990). Para tanto foi utilizada a ferramenta *Power Design*, citada no item 3.1.2.

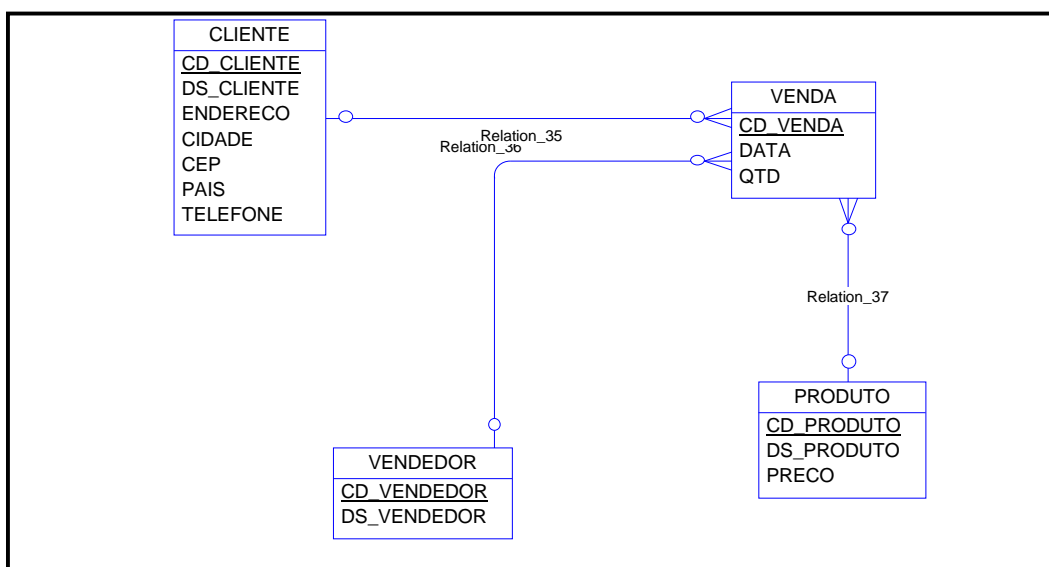
Primeiro tem-se o diagrama de contexto, representado na figura 10 abaixo, que tem como entidade externa somente o vendedor, pois o protótipo tem como objetivo oferecer acesso somente a usuários pré-cadastrados à base de dados da empresa, na qual foi utilizado o banco de dados *Microsoft Access 2000*.

Figura 10 – Diagrama de Contexto



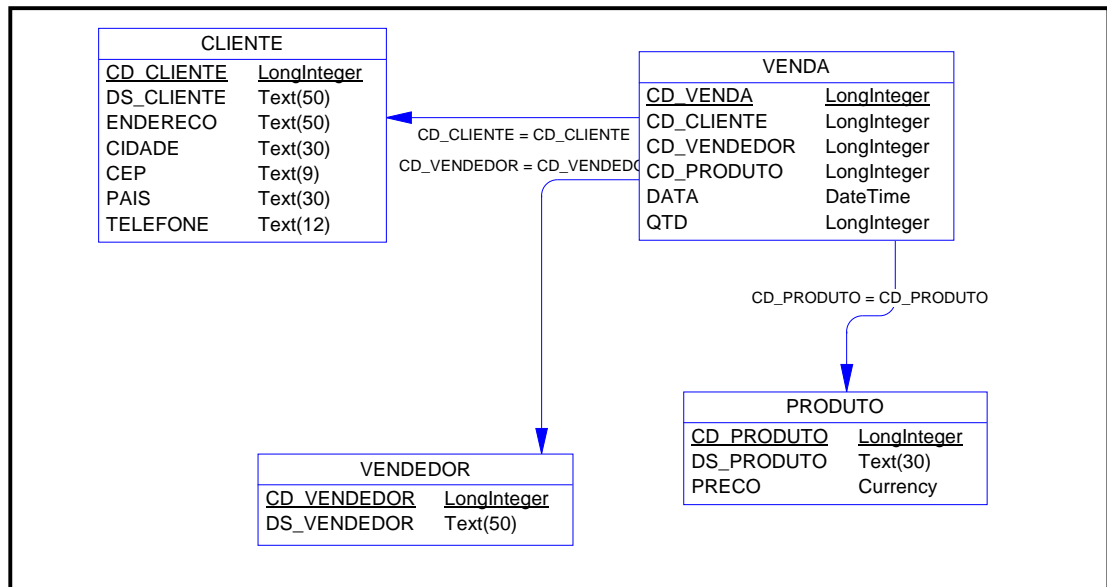
Na figura 11 são apresentadas as entidades e relacionamentos (modelo lógico) que serviram como base para a construção das tabelas do protótipo.

Figura 11 – MER – Modelo Entidade Relacionamento



Na figura 12 são apresentadas as tabelas e seus respectivos atributos utilizados pelo protótipo WAP.

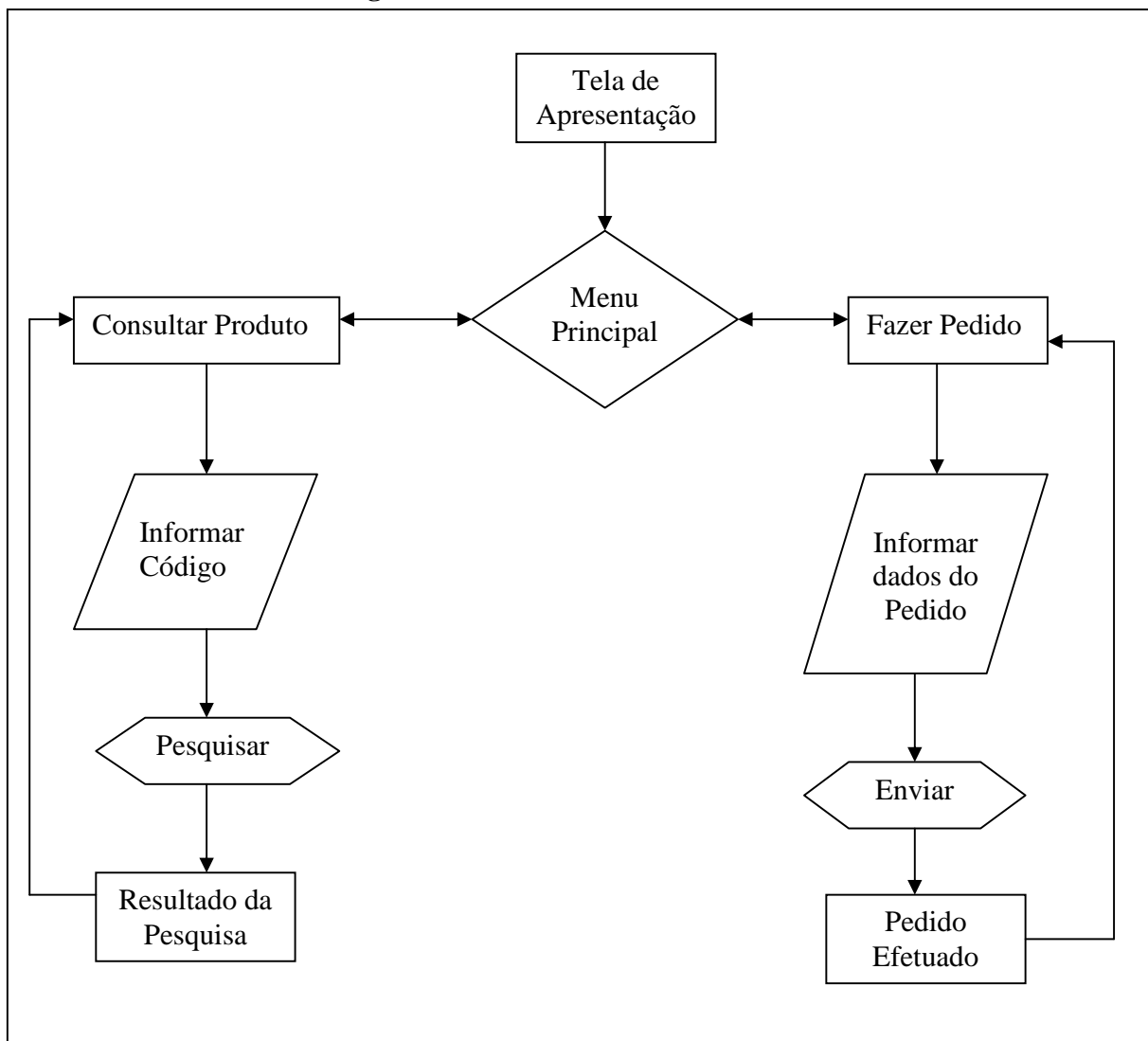
Figura 12 – Tabelas do Protótipo



Na figura 13 abaixo é apresentado o macrofluxo de navegação no protótipo. Onde primeiro há uma breve tela de apresentação, depois há o menu principal no qual oferece duas opções:

- Fazer Pedido**, onde o vendedor vai entrar com os dados do pedido que são o seu código, o código do cliente, o código do produto e por último a quantidade, e após selecionar a opção “Enviar” aparece uma tela “Resultado” contendo os respectivos dados enviados;
- Consultar Produto**, onde o vendedor entra com o código de um produto para consulta, e após selecionar “Pesquisar” aparece uma tela “Resultado” contendo a descrição e o valor do produto.

Figura 13 – Macrofluxo do Menu



3.3 IMPLEMENTAÇÃO E APRESENTAÇÃO DO PROTÓTIPO

Este item apresenta a implementação e a funcionalidade do protótipo. Foram utilizadas as linguagens de programação WML, WMLScript e ASP, e o ambiente de desenvolvimento foi o Nokia WAP ToolKit v.3.0, e também foi utilizado o emulador M3Gate para demonstração do protótipo.

O restante deste capítulo visa apresentar detalhadamente cada uma das páginas constituintes do protótipo, apresentando detalhes da sua implementação e da forma de interação com o usuário.

A tela de apresentação (figura 14) é apresentada por 3 segundos, e em seguida entra automaticamente no “Menu Principal”.

Figura 14 – Tela de Apresentação



A tela inicial da figura 14 é gerada pelo card *inicial* como mostrado no quadro 1.

Quadro 1 – Card inicial

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
<card id="Evaldo" ontimer="menu.wml" title="">
  <timer value="30"/>
  <p align="center">
    Tcc Evaldo
  <br/>
  <small>Sistema de apoio comercial baseado na tecnologia WAP</small>
  </p>
</card>

</wml>
```

Na tela menu principal do protótipo (figura 15), o usuário escolhe uma das duas opções “Fazer Pedido” ou “Consultar Produto”, conforme o card *menu* do quadro 2.

Figura 15 – Menu Principal



Quadro 2 – Card Menu

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
"http://www.wapforum.org/DTD/wml13.dtd">

<wml>
<card id="Menu" title="Menu Principal">
  <p align="left">
    <a href="inserir.asp">Fazer Pedido</a>
  <br/>
    <a href="cons.wml">Consultar Produto</a>
  <br/>
  </p>
</card>

</wml>
```

Se a opção escolhida for fazer cadastro, esta serve para entrar com os dados do pedido (figura 16), gerado pelo card *inserir* do quadro 3.

Figura 16 – Cadastrar Pedido



Quadro 3 – Card *Inserir*

```
<wml>
<card id="Cadastrar" title="Cadastrar">
  <p align="left">
    Codigo do Vendedor:
    <input name="vendedor" type="text" format="*N"/>
    <br/>
    Codigo do Cliente:
    <input name="cliente" type="text" format="*N"/>
    <br/>
    Codigo do Produto:
    <input name="produto" type="text" format="*N"/>
    <br/>
    Qtd:
    <input name="qtd" type="text" format="*N"/>
    <br/>
    <do type="accept" label="Enviar"><go href="qrygeral1.asp" method="post">
<postfield name="venda" value="$(venda:noesc)"/>
    <postfield name="vendedor" value="$(vendedor:noesc)"/>
    <postfield name="cliente" value="$(cliente:noesc)"/>
    <postfield name="produto" value="$(produto:noesc)"/>
    <postfield name="qtd" value="$(qtd:noesc)"/>
    </go></do>
  </p>
  <do label="Menu Pricipal" type="options">
    <go href="menu.wml"/>
  </do>
</card>

</wml>
```

O vendedor deve digitar o seu código, o código do cliente, o código do produto e por último a quantidade do produto desejada. Clicando em *options* selecionará a opção “Enviar”, então os dados serão transmitidos para a empresa. O envio dos dados é feito pelo

código ASP apresentado no quadro 4, o qual ativa a conexão com o banco de dados *Microsoft Access* para receber os dados.

Quadro 4 – Script ASP

```

<%
  Dim conn,rs,rs1
  Set conn = Server.CreateObject("ADODB.Connection")
  Set rs = Server.CreateObject("ADODB.Recordset")
  Set rs1 = Server.CreateObject("ADODB.Recordset")
  conn.open "DRIVER={Microsoft Access Driver (*.mdb)};DBQ=" &
  Server.MapPath("pedido.mdb") & ";"
%>
<% Response.ContentType = "text/vnd.wap.wml" %>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <template>
    <do type="options" name="prev" label="Voltar">
      <prev/>
    </do>
  </template>
  <card id="Login" title="Resultado">
    <p align="center"><b>Pedido Efetuado!</b></p>
    <p>
      <%
        dim data,a,b,c,dia
        cont=0
        a =request.form("vendedor")
        b =request.form("cliente")
        c =request.form("produto")
        d =request.form("qtd")
        dia = date
        data=(now)
        if ((a <> "") and (b <> "") and (c <> "") and (d <> "")) then

          sqlquery = "INSERT INTO venda (cd_vendedor,cd_cliente,data,cd_produto,qtd)
          Values ('&a&','&b&', ' & (data) & ','&c&', '&d&')
          'em caso de erro
          On Error Resume Next
          'executa a instrução
          set rs = conn.Execute(sqlQuery)
          set data = nothing
          rs.Close
          response.write "Vendedor:" & a & "<br/>"
          response.write "Cliente:" & b & "<br/>"
          response.write "Produto:" & c & "<br/>"
          response.write "Qtd:" & d

        %>
        <br/><br/>
        <a href="inserir.asp">Voltar</a>
      <%
        else response.write ("Todos os campos devem ser preenchidos")
        end if %>
    </p>
  </card>
</wml>

```

O *script* ASP do quadro 4 gera uma tela de resultado, ilustrada na figura 17, mostrando os campos e os seus respectivos valores inseridos.

Figura 17 – Tela Resultado do Pedido



Para a opção do menu principal de “Consultar Produto”, pode-se fazer a consulta de um determinado produto apenas digitando o código do mesmo, e como resultado é gerada uma tela (figura 18) através de um *script* ASP (quadro 5) com a descrição deste produto e seu respectivo preço.

Figura 18 – Tela Resultado da Consulta Produto



Quadro 5 – Script Consulta

```

%>
<% Response.ContentType = "text/vnd.wap.wml" %>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="Login" title="">
<p align="center"><b>Resultado</b></p>
<p>
<%
    dim res,cont
    cont=0
    res =Request.Form("Codigo")
if res <> "" then
    sqlQuery = "SELECT * FROM produto WHERE cd_produto = "&res&""
    set rs = conn.Execute(SQLquery)
    While not rs.EOF
        Response.write rs("cd_produto") & " - "
        Response.write rs("Ds_produto") & "<br/>"
        Response.write "Preco : R " &rs("Preco")
        cont=cont+1
        Response.write "<br/>"
        rs.MoveNext

    Wend
    rs.close
end if
if cont = 0 then
Response.write "Nenhuma ocorrencia encontrada"
end if
%>
</p>
<do label="Voltar" type="options">
<go href="consulta.asp"/>
</do>
</card>
</wml>

```

E finalmente na figura 19 está a página da Internet onde a empresa poderá visualizar os pedidos enviados por seus representantes.

Figura 19 – Página Internet

The screenshot shows a Microsoft Internet Explorer browser window displaying a web page titled "Relação de Vendas do dia - 19/11/01". The browser's address bar shows "http://evaldo/INDEX.ASP". The page content is organized into two main sections, each representing a sales transaction for the date 19/11/01.

Transaction 1:
 Date/Time: 19/11/01 17:29:04
 Client: Evaldo
 Vendor: Andrea Silva

Código	Descrição do Produto	Quantidade
5	Cracha	15

TOTAL: R\$ 15

Transaction 2:
 Date/Time: 19/11/01 17:30:28
 Client: Almoarifado Materiais LTDA
 Vendor: Gisleine Costa

Código	Descrição do Produto	Quantidade
7	Relógio de Parede WATCH	30

TOTAL: R\$ 300

Valor total diário: R\$ 315

The browser's status bar at the bottom indicates "Concluído" and "Intranet local".

4 CONSIDERAÇÕES FINAIS

Este capítulo irá apresentar as conclusões com relação ao trabalho desenvolvido como também sugestões para trabalhos futuros nesta área.

4.1 CONCLUSÕES

O número de pessoas que acessam a Internet hoje em dia é relativamente grande. E, com a telefonia celular se disseminando dia a dia, oferecer acesso à Internet, através de dispositivos sem fio baseados na tecnologia WAP, é um potencial mercado para desenvolvimento de serviços.

No decorrer do desenvolvimento deste trabalho, percebeu-se como a atual tecnologia relativa ao WAP ainda carece de evolução. As operadoras terão que oferecer conexões melhores e também reduzir tarifas para atrair clientes, ou seja, o crescimento do WAP depende diretamente da oferta de bons serviços.

Pode-se observar que para desenvolver aplicações WAP tem que ser levada em consideração algumas restrições (com relação a Internet fixa) como: as redes sem fio têm menor largura de banda, os dispositivos tem pouca memória, menor poder de processamento, tela reduzida e recursos de entrada de dados limitados.

Foi de grande importância toda a pesquisa e estudo realizados sobre a tecnologia WAP, podendo verificar seu histórico, seus conceitos e as suas principais características.

Em relação às ferramentas utilizadas, o *Microsoft Access* mostrou-se bastante eficiente na construção do banco de dados. O *Power Design* foi muito útil para o desenvolvimento da especificação do protótipo. O *M3Gate* mostrou-se um excelente emulador on-line, e o *Nokia WAP ToolKit* também foi uma ferramenta muito importante para o desenvolvimento, simulação e demonstração do protótipo.

Quanto à linguagem ASP, mostrou-se muito eficiente para o acesso a um banco de dados. Enquanto a linguagem WML mostrou-se bastante funcional e similar ao HTML.

4.2 SUGESTÕES

Como sugestões para trabalhos futuros, poderiam ser incorporadas mais funções ao protótipo, tais como:

- a) possibilitar pedir mais de um item de cada vez;
- b) habilitar o cadastro de novos clientes pelo celular;
- c) consultar o estoque de produtos da empresa;
- d) consultar o histórico de pedidos de um determinado cliente.

ANEXO 1

Código fonte do protótipo:

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
"http://www.wapforum.org/DTD/wml13.dtd">
<wml>
<card id="Menu" title="Menu Principal">
<p align="left">
  <a href="inserir.asp">Fazer Pedido</a><br/>
  <a href="cons.wml">Consultar Produto</a><br/>
</p>
</card>
</wml>

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<head>
  <meta http-equiv="Cache-Control" content="max-age=0"/>
</head>
<card id="teste" title="Cadastrar">
<p align="left">
  <a href="inserir.asp">Pedido</a><br/>
</p>
  <do label="Menu Principal" type="options">
    <go href="menu.wml"/>
  </do>
</card>
</wml>

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<head>
  <meta http-equiv="Cache-Control" content="max-age=0"/>
</head>
<card id="teste" title="Consultar">
<p align="left">
  <a href="consulta.asp">Produtos</a><br/>
</p>
  <do label="Menu Principal" type="options">
```

```

        <go href="menu.wml" />
    </do>
</card>
</wml>

<%acceptString = Request.ServerVariables("HTTP_ACCEPT")
    If Instr(LCase(acceptString), "wap") > 0 Then
        Response.ContentType = "text/vnd.wap.wml"
        %>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
    "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="Evaldo" ontimer="menu.wml" title="">
<timer value="30"/>
<p align="center">
Tcc Evaldo<br/>
<small>Sistema de apoio comercial baseado na tecnologia
WAP</small>
</p>
</card>
</wml>
<%
    Else
        Response.ContentType = "text/html"
        %>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<% dataatual = date %>
<html>
<head>
    <title></title>
</head>
<body topmargin="0" leftmargin="0" bgcolor="white"
text="black">
<table border="0" cellpadding="1" cellspacing="0"
width="640">
<tr>
<td><font face="Verdana" size="5" color="#000000">
<b>Relação de Vendas do dia - <%response.write dataatual
%></b></font></td>
</tr>
</table>
<br>
<table border="2" cellspacing="2" cellpadding="2"
width="640">

```



```

<tr><td align="center" valign="top">
<br>

<%
    set con=Server.CreateObject("ADODB.Connection")
    Set rs= Server.CreateObject("ADODB.Recordset")
    con.Open("DRIVER={Microsoft Access Driver (*.mdb)};
DBQ=" & Server.MapPath("pedido.mdb"))
    SQL= "Select produto.preco, produto.ds_produto,
produto.cd_produto, venda.cd_venda, venda.data, venda.qtd,
venda.cd_produto, venda.cd_vendedor, venda.cd_cliente,
cliente.cnpj, cliente.ds_cliente, vendedor.cd_vendedor,
vendedor.ds_vendedor From venda, vendedor, cliente,produto
"&_
    "Where venda.data like '%"&dataatual&%"' and
venda.cd_produto = produto.cd_produto and cliente.cnpj =
venda.cd_cliente and vendedor.cd_vendedor =venda.cd_vendedor"
'response.write sql
total_final=0
    set rs=con.execute(SQL)
    on error resume next
    if NOT RS.EOF then
    do until RS.EOF
        Response.write "<br>"
        Response.write "<table border=0 cellpadding=1
cellspacing=1 width=640><tr>"
        Response.write "<td width=50% align=right><font
color=#000000 face=Arial size=2>Data/Hora: " & rs("data") &
"</font></td></tr>"

        Response.write "<tr>"
        Response.write "<td width=50% ><font color=#000000
face=Arial size=2>Cliente: " & rs("ds_cliente") &
"</font></td>"
        Response.write "<td width=50% align=right><font
color=#000000 face=Arial size=2>Vendedor: " &
rs("ds_vendedor") & "</font></td>"
        Response.write "</tr>"
        Response.write "</table>"
        subtot=0
        Response.write "<table border=0 cellpadding=1
cellspacing=1 width=640><tr bgcolor=#808080>"
        Response.write "<td align=center width=25%
><b><font color=#ffffff face=Verdana
size=1>Código</font></b></td>"

```

```

                Response.write "<td align=center width=50%
><b><font color=#ffffff face=Verdana size=1>Descrição do
Produto</font></b></td>"
                Response.write "<td align=center width=25%
><b><font color=#ffffff face=Verdana
size=1>Quantidade</font></b></td>"
                Response.write "</tr>"
                subtot = rs("qtd") * rs("preco")
                Response.write "<tr>"
                Response.write "<td align=center><font face=Verdana
size=1 color=#000000>" & rs("cd_produto") & "</font></td>"
                Response.write "<td align=center><font face=Verdana
size=1 color=#000000>" & rs("ds_produto") & "</font></td>"
                Response.write "<td align=center><font face=Verdana
size=1 color=#000000>" & rs("qtd") & "</font></td>"
                Response.write "</tr>"
                total_final = total_final + subtot
                Response.write "<tr>"
                Response.write "<td>&nbsp;</td>"
                Response.write "<td>&nbsp;</td>"
                Response.write "<td>&nbsp;</td>"
                Response.write "</tr>"
                Response.write "<tr>"
                Response.write "<td>&nbsp;</td>"
                Response.write "<td>&nbsp;</td>"
                Response.write "<td align=right><font face=Verdana
size=1 color=#000000><b>TOTAL</b>: R$ " & subtot &
"</font></td>"
                subtot=0
                Response.write "</tr>"
                Response.write "<tr bgcolor=#cdcdcd><td colspan=3
height=1></td></tr>"
                Response.write "</table>"
                Response.write "<br>"
                Response.write "<br>"
                rs.movenext
                loop
                else response.write "<br><HR><BR>Nenhuma venda
realizada hoje"
                end if
                Response.write "<table border=0 cellpadding=1
cellspacing=1 width=640>"
                Response.write "<tr bgcolor=#cdcdcd>"
                Response.write "<td align=right><font face=Verdana
size=1 color=#000000><b>Valor total diário</b>: R$ " &
total_final & "</font></td>"
                Response.write "</tr>"

```

```

        Response.write "</table>"
        Response.write "<br>"
        Response.write "<br>"
    %>
</center>
<br>
</center>
</body>
</html>
    <%End If%>

<% Response.ContentType = "text/vnd.wap.wml" %>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
    <card id="Cadastrar" title="Cadastrar">
        <p align="left">
            Codigo do Vendedor:
            <input name="vendedor" type="text" format="*N"
/><br/>
            Codigo do Cliente:
            <input name="cliente" type="text" format="*N"/><br/>
            Codigo do Produto:
            <input name="produto" type="text" format="*N"/><br/>
            Qtd:
            <input name="qtd" type="text" format="*N"/><br/>
            <do type="accept" label="Enviar">
                <go href="qrygerall.asp" method="post">
                    <postfield name="venda" value="$venda" />
                    <postfield name="vendedor" value="$vendedor"
/>
                    <postfield name="cliente" value="$cliente" />
                    <postfield name="produto" value="$produto" />
                    <postfield name="qtd" value="$qtd" />
                </go>
            </do>
            <do label="Menu Pricipal" type="options">
                <go href="menu.wml"/>
            </do>
        </p>
    </card>
</wml>

<% Response.ContentType = "text/vnd.wap.wml" %>
<?xml version="1.0"?>

```

```

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="principal" title="Pesquisa">
  <p align="center">
    Pesquisar Produto: <br/>
  </p>
  <p align="left">
   Codigo:<input size="10" name="codigo"/>
  </p>
<do type="accept" label="Pesquisar">
  <go href="consulta.asp" method="post">
    <postfield name="codigo" value="$codigo"/>
  </go>
</do>
<do label="Menu Principal" type="options">
  <go href="menu.wml"/>
</do>
</card>
</wml>

<%
  Dim conn, rs
  Set conn = Server.CreateObject("ADODB.Connection")
  Set rs    = Server.CreateObject("ADODB.Recordset")
conn.open "DRIVER={Microsoft Access Driver (*.mdb)};DBQ=" &
Server.MapPath("pedido.mdb") & ";"
%>
<% Response.ContentType = "text/vnd.wap.wml" %>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="Login" title="">
  <p align="center"><b>Resultado</b></p>
  <p>
  <%
    dim res,cont
    cont=0
    res =Request.Form("Codigo")
if res <> "" then
  sqlQuery = "SELECT * FROM produto WHERE cd_produto =
"&res&" "
  set rs = conn.Execute(SQLquery)
  While not rs.EOF
    Response.write rs("cd_produto") & " - "

```

```

        Response.write rs("Ds_produto") & "<br/>"
Response.write "Preco : R " &rs("Preco")
cont=cont+1
        Response.write "<br/>"
        rs.MoveNext
    Wend
    rs.close
end if
if cont = 0 then
Response.write "Nenhuma ocorrencia encontrada"
end if
%>
    </p>
        <do label="Voltar" type="options">
            <go href="consulta.asp"/>
        </do>
</card>
</wml>

<%
    Dim conn, rs,rs1
    Set conn = Server.CreateObject("ADODB.Connection")
    Set rs = Server.CreateObject("ADODB.Recordset")
    Set rs1 = Server.CreateObject("ADODB.Recordset")
conn.open "DRIVER={Microsoft Access Driver (*.mdb)};DBQ=" &
Server.MapPath("pedido.mdb") & ";"
%>
<% Response.ContentType = "text/vnd.wap.wml" %>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
    <template>
        <do type="options" name="prev" label="Voltar">
            <prev/>
        </do>
    </template>
<card id="Login" title="Resultado">
    <p align="center"><b>Pedido Efetuado!</b></p>
    <p>
<%
    dim data,a,b,c,dia
    cont=0
    a =request.form("vendedor")
    b =request.form("cliente")
    c =request.form("produto")

```

```

d =request.form("qtd")
dia = date
data=(now)
if ((a <> "") and (b <> "") and (c <> "") and (d <> "")) then
    sqlquery = "INSERT INTO venda
(cd_vendedor,cd_cliente,data,cd_produto,qtd) Values
("&a&","&b&"," & (data) & "','"&c&","&d&")"
    'em caso de erro
    On Error Resume Next
    'executa a instrução
    set rs = conn.Execute(sqlQuery)
    set data = nothing
    rs.Close
    response.write "Vendedor:" & a & "<br/>"
    response.write "Cliente:" & b & "<br/>"
    response.write "Produto:" & c & "<br/>"
    response.write "Qtd:" & d
%>
    <br/><br/>
    <a href="inserir.asp">Voltar</a>
<%
else response.write ("Todos os campos devem ser preenchidos")
end if %>
</p>
</card>
</wml>

```

REFERÊNCIAS BIBLIOGRÁFICAS

CANI, Giovani Endrigo. **Protótipo de software para acesso a informações baseado na tecnologia WAP**. 2000. 88 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

CARVALHO, Allan; ROCHA Leandro M. **Aprenda WAP**. Endereço Eletrônico: <http://www.aprendawap.com.br>. Data da consulta: 21/09/2001.

DEMÉTRIO, Rinaldo. **A Tecnologia WAP : aprenda a criar sites para celulares com a linguagem WML**. São Paulo: Érica, 2000.

DIAS, Adilson de Souza. **WAP - wireless application protocol : a internet sem fios**. Rio de Janeiro: Ciência Moderna, 2000.

GALANTE, Luiz Cláudio. **Construindo aplicações com ACCESS 2.0 para WINDOWS**. São Paulo: Makron Books, 1994.

HENKEL, César Augusto. **WAP : wireless application protocol**. 2001. 103 f. Monografia (Curso Especialista em Redes de Computadores e Internet) – Centro de Ciências Exatas e Tecnológicas, Universidade do Vale do Rio dos Sinos, São Leopoldo.

JESUS, Alberto Pereira de. **Protótipo de um sistema de serviços WAP para a biblioteca central da FURB**. 2000. 65 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

RADUENZ, Gilmar Vereano. **Protótipo de aplicativo para escritório de advocacia com acesso a Internet baseado em ASP**. 2000. 73 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

SILVA, Genivaldo Francisco da Junior. **WAP : wireless application protocol**. 2000. 89 f. Monografia (Curso de Tecnologia em Processamento de dados) – Centro de Ciências Formais e Tecnológicas, Universidade Tiradentes, Aracajú.

WAPBRASIL. **WAP Brasil**. Endereço eletrônico: <http://www.wapbr.org>. Data da consulta: 05/10/2001.

WAPMASTER. **Conheça o WAP**. Endereço eletrônico: <http://www.wapmaster.com.br>. Data da consulta: 25/09/2001.

WEISSINGER, A. Keyton. **ASP – guia completo**. Rio de Janeiro: Ciência Moderna, 1999.

YOURDON, E. **Análise estruturada moderna**. Rio de Janeiro: Campus, 1990.