

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

**PROTÓTIPO DE UM SISTEMA DE SERVIÇOS WAP PARA A  
BIBLIOTECA CENTRAL DA FURB**

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

**ALBERTO PEREIRA DE JESUS**

BLUMENAU, NOVEMBRO/2000.

2000/2-3

# **PROTÓTIPO DE UM SISTEMA DE SERVIÇOS WAP PARA A BIBLIOTECA CENTRAL DA FURB**

**ALBERTO PEREIRA DE JESUS**

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

---

Mauro Tessari (Esp. Administração Bibliotecas)

# DEDICATÓRIA

Dedico este trabalho a meu pai e minha mãe, que me incentivaram nos momentos difíceis e fizeram com que seguisse a frente.

## **AGRADECIMENTOS**

Agradeço aos meus pais, por estarem ao meu lado em todos os momentos, incentivando e auxiliando em mais esta etapa da minha vida.

À Biblioteca Central da Universidade Regional de Blumenau - FURB, por disponibilizar espaço para a realização deste trabalho.

Aos meus colegas de trabalho Mauro Tessari, Alexander Roberto Valdameri, Marcos Rogério Cardoso, Evanilde Maria Moser e Izildinha Ramos Acceta por me ajudarem e auxiliarem no desenvolvimento.

Ao coordenador do trabalho de conclusão de curso, Prof. José Roque Voltolini da Silva por toda atenção e apoio dispensados.

Ao professor Francisco Adell Péricas, por acreditar e confiar no meu potencial e todo apoio prestado para conclusão deste trabalho.

A todos aqueles que diretamente ou indiretamente contribuíram para a conclusão deste trabalho.

Não basta saber,  
é preciso também aplicar;  
não basta querer,  
é preciso também agir.

Goethe

# SUMÁRIO

<b>SUMÁRIO .....</b>	<b>VI</b>
<b>LISTA DE FIGURAS .....</b>	<b>VIII</b>
<b>LISTA DE ABREVIATURAS .....</b>	<b>IX</b>
<b>RESUMO.....</b>	<b>XI</b>
<b>ABSTRACT .....</b>	<b>XII</b>
<b>1 INTRODUÇÃO .....</b>	<b>1</b>
1.1 MOTIVAÇÃO.....	2
1.2 ÁREA.....	3
1.3 JUSTIFICATIVAS.....	3
1.4 OBJETIVOS .....	3
1.5 ESTRUTURA DO TRABALHO.....	3
<b>2 BIBLIOTECAS.....</b>	<b>4</b>
2.1 BIBLIOTECAS E A INFORMAÇÃO.....	4
2.2 A BIBLIOTECA CENTRAL DA FURB .....	4
<b>3 WAP.....</b>	<b>6</b>
3.1 DEFINIÇÕES.....	6
3.2 WAP FÓRUM .....	6
3.3 AMBIENTE WAP .....	8
3.4 MODELOS DE REDE.....	9
3.4.1 O modelo WWW .....	10
3.4.2 O modelo WAP .....	12
3.5 ARQUITETURA.....	14
3.5.1 Componentes da arquitetura WAP .....	14
3.5.2 Portadoras - bearer services.....	18

3.6	LINGUAGENS WAP .....	18
3.6.1	Wireless Markup Language - WML.....	19
3.6.2	WMLScript.....	20
3.7	PÁGINAS DINÂMICAS EM WML .....	20
3.7.1	ASP e WML .....	21
3.8	APLICAÇÕES.....	21
<b>4</b>	<b>DESENVOLVIMENTO DO PROTÓTIPO .....</b>	<b>23</b>
4.1	ESPECIFICAÇÃO.....	23
4.1.1	Ferramenta utilizada .....	23
4.1.2	Diagrama de contexto.....	23
4.1.3	Modelo entidade Relacionamento .....	24
4.1.4	Diagrams de fluxo de dados .....	26
4.2	IMPLEMENTAÇÃO .....	28
4.2.1	Técnicas e ferramentas utilizadas.....	28
4.2.2	Operacionalidade da Implementação.....	32
4.3	APRESENTAÇÃO DO PROTÓTIPO .....	36
<b>5</b>	<b>CONCLUSÕES E SUGESTÕES .....</b>	<b>41</b>
5.1	CONCLUSÕES.....	41
5.2	LIMITAÇÕES .....	42
5.3	SUGESTÕES .....	42
	<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>43</b>
	<b>ANEXO I.....</b>	<b>45</b>
	<b>ANEXO II .....</b>	<b>59</b>

## LISTA DE FIGURAS

Figura 1 – Codificação conteúdo via WAP .....	9
Figura 2 - Comparativo protocolos Internet e WAP .....	10
Figura 3 - Modelo WEB .....	10
Figura 4 - Modelo WAP .....	12
Figura 5 - Arquitetura WAP .....	14
Figura 6 - Diagrama de Contexto .....	24
Figura 7 - modelo entidade relacionamento (MER).....	25
Figura 8 - Consultar Acervo .....	26
Figura 9 - Consultar detalhes MFN .....	27
Figura 10 - Manter reserva .....	27
Figura 11 - Consultar reserva .....	28
Figura 12 - WAPDrive WAPtor .....	29
Figura 13 - UP Simulador.....	30
Figura 14 - Debug UP Simulador .....	30
Figura 15 – IIS Windows 2000 .....	32
Figura 16 – Macro fluxo sistema.....	33
Figura 17 - Tela de apresentação .....	36
Figura 18 - Menu sistema .....	37
Figura 19 - Tipo de busca.....	37
Figura 20 - Expressão de busca .....	37
Figura 21 - Resultado busca .....	38
Figura 22 - Detalhes do título.....	38
Figura 23 - Reserva de material.....	38
Figura 24 - Reposta reserva .....	39
Figura 25 - Consulta reserva.....	39
Figura 26 - Espera negativa .....	39
Figura 27 - Espera positiva.....	40
Figura 28 - Ajuda.....	40
Figura 29 - Contatos .....	40



## LISTA DE ABREVIATURAS

**FURB** – Fundação Universidade Regional de Blumenau

**BC** – Biblioteca Central da FURB

**WAP** – *Wireless Application Protocol*

**WML** – *Wireless Markup Language*

**IP**- *Internet Protocol*

**WWW**- *World Wide Web*

**HTML**- *Hipertext Markup Language*

**ASP**- *Active Server Page*

**HTML**- *Hypertext Markup Language*

**PDA**- *Personal Digital Assistant*

**TDMA**- *Time Division Multiple Access*

**CDMA**- *Code Division Multiple Access*

**GSM**- *Global System for Mobile Communication*

**WDP**- *Wireless Datagram Protocol*

**ETSI**- *European Telecommunication Standard Institute*

**W3C**- *World Wide Web Consortium*

**TIA**- *Telecommunication Industry Association*

**CTIA**- *Cellular Telecommunication Industry Association*

**RAM**- *Random Access Memory*

**ROM**- *Read-Only Memory*

**SMS-** *Short Message Service*

**WSP-** *Wireless Session Protocol*

**WTP-** *Wireless Transaction Protocol*

**WTLS-** *Wireless Transport Layer Security*

**SQL-** *Structured Query Language*

**IIS-** *Internet Information Service*

**MINE-** *Multipurpose Internet Mail Extensions*

**HSCSD-** *High-Speed Circuit Switched Data*

**GPRS-** *General Packet Radio Service*

**HTTP-** *Hipertext Transport Protocol*

**URL-** *Uniform Resource Locator*

**TCP-** *Transport Control Protocol*

**CGI-** *Common Gateway Interface*

**WAE-** *Wireless Application Enviroment*

**WTA-** *Wireless Telephony Application*

**PDU-** *Protocol Data Unit*

**XML-** *Extensible Mark-up Language*

**HDML-** *Handhel Device Markup Language*

**WBMP-** *Wireless BitMap*

**MER-** *Modelo Entidade Relacionamento*

**MFN-** *Master File Number*

## RESUMO

Consiste no desenvolvimento de uma interface para acesso aos serviços da Biblioteca Central da Universidade Regional de Blumenau - FURB através da tecnologia WAP (*Wireless Application Protocol*) para celulares, utilizando para isso toda a estrutura de dados da biblioteca e gerando páginas dinâmicas na linguagem WML (*Wireless Markup Language*).

## **ABSTRACT**

Consists about developing an access interface to the services provided by Universidade Regional de Blumenau - FURB Central Library through WAP (Wireless Application Protocol) technology for cellular phones, using all the data structure from the library and generating dynamic pages in WML (Wireless Markup Language).

# 1 INTRODUÇÃO

Mudança. Informação. Tecnologia. Sociedade digital. Diariamente essas palavras são ouvidas, lidas ou escritas milhares de vezes. O mundo está perplexo diante das mudanças que estão ocorrendo com a natureza, a velocidade e a aplicação da tecnologia da informação nas empresas, nas escolas, nas universidades, nos lares, na vida de todos. Segundo [TUD2000], um novo mundo está sendo construído a partir da convergência do computador e das tecnologias de telecomunicação.

É uma revolução que supera todos os saltos anteriores da humanidade pelo modo profundo e veloz com que transforma os processos produtivos, as instituições e o estilo de vida das pessoas. Uma mudança tão forte que ninguém mais ousa traçar uma fronteira rígida entre ficção e realidade. Apesar de não se ter conseguido, ainda, expressar com precisão o alcance dessas mudanças e de suas implicações, a inquietação e ansiedade delas decorrentes estão instaladas ([CUN1999] [TUD2000]).

Aqueles que lidam com a informação como atividade finalística encontram-se, mais do que nunca, cercados de desafios: a informação cada vez mais ágil, cada vez mais fácil. As inovações tecnológicas vêm encurtando o tempo e o acesso às informações de forma impossível de se prever ([CUN1999]).

A área de biblioteca encontra-se atingida por esse desafio: inovar ou ser superada pelas novas tecnologias. Aos serviços tradicionais devem ser incorporados novos serviços, compatíveis com novos paradigmas da tecnologia de informação.

Uma evolução dos serviços das bibliotecas é o acesso das informações via Internet, e, através da tecnologia WAP, o acesso das informações via telefonia celular.

Um sistema de serviço WAP é composto por um celular que suporta a tecnologia, uma operadora que disponibiliza o serviço e por um portal WAP que disponibiliza informações e serviços através de WML ([GOW2000] [WAP2000b]).

O celular acessa a Internet da mesma maneira que o seu computador. O aparelho recebe um endereço IP (o seu identificador) da operadora e, a partir daí, já está fazendo parte da grande rede, pois está conectado à Internet da mesma forma que o computador de casa faz ([TEL2000]).

WAP é a sigla de *Wireless Application Protocol* (Protocolo para Aplicações sem Fio), que é um conjunto de especificações que define um ambiente semelhante à Web, mas que funciona em redes de aparelhos sem fio e em velocidades mais baixas, como é o caso dos telefones celulares. Utilizam um micro-navegador que acessa páginas criadas em WML (*Wireless Markup Language*). O WAP é um padrão aberto, desenvolvido e controlado pelo WAP Fórum, entidade que congrega as principais empresas internacionais de telecomunicações sem fio e informática ([TEL2000]).

Os navegadores que se conhecem, como Internet Explorer ou Netscape, interpretam páginas HTML da Web em PCs. Segundo GOWap ([GOW2000]), WAPBrasil ([WAP2000a]), um micro-navegador WAP é um software similar, ainda que muito mais simples, criado para funcionar em celulares e outros aparelhos sem fio, interpretando páginas WML. Os portais WAP criados na linguagem WML são baseados quase que totalmente em texto, com pouquíssimas imagens monocromáticas e sem qualquer tipo de diagramação. Lembram muito os primeiros portais da Web, que apareceram em 1993. Ainda há poucos portais WAP na Internet, mas surgem novos a cada dia.

WML é a sigla de *Wireless Markup Language*. É a linguagem utilizada para construir os mini portais para os celulares (chamados "*decks*"), divididos em telas ou páginas (conhecidas como "*cards*") ([GOW2000] [WAP2000a]).

Sendo assim, este trabalho consiste no desenvolvimento de um protótipo de um portal de serviços da Biblioteca Central da FURB para WAP, através do estudo da tecnologia, da implantação de páginas WML, criadas dinamicamente através de ASP (*Active Server Pages*) acessando dados no banco de dados *Oracle*, da configuração do servidor Web para acesso WAP.

Os usuários da Biblioteca Central da FURB, através do seu celular com tecnologia WAP, poderão consultar livros, verificar reservas ou consultar sua situação junto a Biblioteca Central da FURB.

## 1.1 MOTIVAÇÃO

O acesso à Internet já está difundido e utilizado por grande número de pessoas. Com o crescimento da telefonia celular está se criando um novo mercado de acesso à Internet através dos celulares, onde tem-se acesso a informações em nossos aparelhos independente do lugar onde localizar-se.

## **1.2 ÁREA**

Este trabalho abrange as áreas de disseminação da informação, Internet, comunicação de dados sem fio, telefonia celular, desenvolvimento para Internet com WML e ASP.

## **1.3 JUSTIFICATIVAS**

A importância deste trabalho está na facilidade com que os usuários da Biblioteca Central da FURB passarão a obter informações sobre o acervo e a ter acesso a serviços da biblioteca, através da Internet fixa e móvel, dispensando assim ao usuário que se desloque a biblioteca para fazer uma pesquisa ou consultar se o livro que reservou já está disponível.

## **1.4 OBJETIVOS**

O objetivo principal é a implementação de um protótipo para disponibilizar os serviços da Biblioteca Central da FURB, através da tecnologia WAP.

Como objetivos secundários pretende-se:

- a) estudar e mostrar a tecnologia WAP;
- b) estudar e mostrar linguagem de programação WML;
- c) melhorar e facilitar o atendimento aos usuários da Biblioteca Central da FURB.

## **1.5 ESTRUTURA DO TRABALHO**

O primeiro capítulo apresenta uma introdução sobre o assunto, justificativas, área e objetivos do trabalho.

O segundo capítulo contextualiza as Bibliotecas, sobre disseminação da informação, apresenta a Biblioteca Central da FURB e sua evolução.

O terceiro capítulo é dedicado às fundamentações e conceituações sobre as tecnologias que serão utilizadas no desenvolvimento deste trabalho, como WAP, WML e ASP.

O quarto capítulo descreve a estruturação necessária para o desenvolvimento do protótipo até seu estágio final, destacando a metodologia utilizada, a modelagem do sistema e considerações sobre sua codificação.

O quinto capítulo conclui o trabalho, apresenta limitações e também, sugestões para trabalhos futuros.

## 2 BIBLIOTECAS

Este capítulo apresenta as bibliotecas e os serviços oferecidos por elas. A principal evolução pela qual estão passando é consequência da necessidade de suprir a informação de forma cada vez mais rápida e ágil, dando ênfase à Biblioteca Central da FURB.

### 2.1 BIBLIOTECAS E A INFORMAÇÃO

Há milhares de anos as bibliotecas são o meio de distribuição da informação e do conhecimento, e têm seu espaço garantido e inquestionável em qualquer instituição de ensino e pesquisa. São fundamentais para dar suporte ao aprendizado e à pesquisa.

A tecnologia vem proporcionando profundas transformações nos procedimentos de produção, transmissão e uso da informação. Às necessidades do ser humano – ar, água, alimentação e abrigo – pode-se acrescentar um quinto fator essencial à sua sobrevivência e evolução que é a necessidade de informação. No atual momento, as organizações tem um novo paradigma, sendo a informação e o conhecimento seu bem mais precioso ([ACC1998]).

Com a necessidade de colocar a informação ao alcance de qualquer pessoa, quando e onde for necessário, e por outro lado, com o aumento global do volume de publicações, torna-se difícil gerenciar, disseminar e recuperar a informação em tempo hábil ([TEI1997]).

Considerando a informática como o fator fundamental no processo de recuperação de informação, percebe que a cada momento surgem tecnologias cada vez mais sofisticadas que permitem uma interação direta entre os usuário e os sistemas ([TEI1997]).

Atualmente, a arte de disseminar e recuperar informações bibliográficas dispõe, além dos recursos usuais de recuperação, dos serviços em nível mundial como a Internet, baseado nos conceitos de hipertexto e hipermídia, e mais recentemente, a Internet para telefones celulares (*Wireless Application Protocol* ou WAP), colocando a informação cada vez mais perto do usuário e de forma mais rápida ([TEI1997]).

### 2.2 A BIBLIOTECA CENTRAL DA FURB

A Biblioteca Central Martinho Cardoso da Veiga é um órgão suplementar da Universidade Regional de Blumenau.



Sua missão é desenvolver e colocar a disposição da comunidade universitária um acervo bibliográfico que atenda as necessidades de informação para as atividades de ensino, pesquisa e extensão, adotando modernas tecnologias para o tratamento, recuperação e transferência da informação ([ACC1998]).

Está aberta à comunidade em geral para consultas e permite o empréstimo domiciliar aos usuários vinculados à Instituição ou seja, corpo docente e técnico administrativo da FURB ([ACC1998]).

A Biblioteca Central começou seu processo de automação em 1987, quando através de um convênio com a FGV – Fundação Getúlio Vargas, passou a usar o Sistema Bibliodata Calco. Desde então percorre um caminho de inovações e melhorias no seu processo de automação. O sistema de automação da Biblioteca Central, foi utilizado por outras bibliotecas das universidades do estado. Foi uma das primeiras bibliotecas a disponibilizar consulta do seu acervo pela Internet ([ACC1999]).

Os serviços oferecidos atualmente aos usuários da Biblioteca Central são consulta ao acervo tanto na rede local ou via Internet. O usuário não tem como efetuar uma reserva, verificar se uma reserva chegou sem vir até a seção de empréstimo.

O processo de automação da Biblioteca Central está sempre buscando oferecer a informação de forma rápida e de fácil acesso ao seu usuário. Sendo assim está visando oferecer seus serviços através da tecnologia WAP (Internet para celulares), onde seu usuário tendo um aparelho que suporte a tecnologia poderá consultar o acervo da biblioteca, fazer sua reserva ou verificar se o livro que reservou já está a sua espera, não precisando se deslocar até a Biblioteca Central.

## 3 WAP

Este capítulo apresenta a fundamentação das técnicas e tecnologias que serão utilizadas na implementação do protótipo.

### 3.1 DEFINIÇÕES

WAP (*Wireless Application Protocol*) é um conjunto de protocolos e definições, destinado a disponibilizar conteúdo da Internet em dispositivos de comunicação móvel.

A especificação do protocolo WAP foi resultado do esforço de um conjunto de empresas que deu origem ao chamado WAP Fórum, órgão responsável pela publicação das especificações WAP, e pela homologação de produtos WAP.

O telefone celular, o PDA (*Personal Digital Assistants*) e o *pager* são exemplos de dispositivos que farão uso do WAP. Além destes, qualquer outro dispositivo sem fio (*two-way radios, smartphones e communicators*) com homologação pelo WAP Fórum poderá se beneficiar dos serviços oferecidos pela especificação WAP, como correio eletrônico, comércio eletrônico, *home banking, booking, etc.*

WAP é projetado para trabalhar com a maioria das redes sem fio existentes, tais como as baseadas TDMA (*Time Division Multiple Access*), CDMA (*Code Division Multiple Access*), GSM (*Global System for Mobile Communication*). A especificação de sua camada WDP (*Wireless Datagram Protocol*), que credencia diferentes tecnologias de redes sem fio, vem se expandindo constantemente.

A idéia do WAP é agregar valor a serviços de redes sem fio colocando “**inteligência**” em servidores WAP e adicionando somente um micro-navegador (chamado *micro-browser*) no terminal móvel.

### 3.2 WAP FÓRUM

O WAP Fórum é uma associação de indústrias criada em Junho de 1997 - inicialmente formada pela norte americana Phone.com, Ericsson, Nokia e Motorola - para desenvolver um padrão para serviços de informação destinados a redes sem fio e telefonia móvel.

O WAP Fórum é um Fórum aberto a qualquer participante da indústria relacionada à tecnologia WAP. Em Jun/2000 já contava com mais de 500 membros, os quais representam

cerca de 90% do mercado de aparelhos móveis. As principais empresas desenvolvedoras de software e provedores de infra-estrutura para redes sem fio engrossam a lista do WAP Fórum ([FOR2000]).

O WAP Fórum estabeleceu vários grupos de trabalho, dentre os quais destacam-se: WAP Architecture (ArchG), WAP Wireless Protocol (WPG), WAP Wireless Security (WSG) e WAP Wireless Application Working Groups (WAG) ([FOR2000]).

Objetivo do WAP Fórum é agregar empresas de vários segmentos da indústria de redes sem fio para garantir interoperabilidade de produto e alavancar o mercado de redes sem fio ([FOR2000]).

O WAP Fórum [FOR2000], define seus objetivos como sendo:

- a) disponibilizar conteúdo da Internet e/ou serviços de dados avançados a telefones celulares digitais ou outros terminais sem fio;
- b) criar uma especificação global de protocolos para redes sem fio que irá interoperar entre diferentes tecnologias de redes sem fio (CDMA, GSM, TDMA, etc);
- c) incentivar a criação de conteúdo e aplicações que serão disponibilizados através de uma variedade de redes sem fio e de tipos de terminais;
- d) adotar, sempre que possível, padrões e tecnologia existentes e estendê-los somente quando for necessário.

O WAP Fórum possui um programa de certificação de produto ao qual fabricantes se credenciam para obter um certificado de conformação de seus produtos com o WAP 1.1 *Suite Protocol* (versão WAP corrente). Além deste, possui um programa de registro de desenvolvedor e um programa de verificação de conteúdo. Utilizando estes programas desenvolvedores de software podem licenciar seus sites WAP como "*WML Compatible*" ([FOR2000]).

O WAP Fórum [FOR2000] mantém relações com outros órgãos padronizadores internacionais com o objetivo de assegurar a convergência de futuras especificações. Alguns destes órgãos são:

- a) ETSI (*European Telecommunication Standard Institute*);
- b) W3C (*World Wide Web Consortium*);
- c) TIA (*Telecommunication Industry Association*);
- d) CTIA (*Cellular Telecommunication Industry Association*).

### 3.3 AMBIENTE WAP

A tecnologia WAP está posicionada na convergência de duas tecnologias de rede que estão evoluindo muito rapidamente: a transmissão de dados sem fio e a Internet.

A idéia básica do WAP é utilizar os conceitos da Internet modificando-os para que eles possam ser úteis dentro das limitações de uma rede sem fio e de aparelhos com recursos restritos. Fazendo com que desenvolvedores Web construam serviços WAP com ferramentas muito semelhantes às existentes no mercado, os usuários poderão desfrutar da Internet em aparelhos sem fio, como os celulares.

A maioria das tecnologias desenvolvidas para a Internet foi projetada para computadores com grande capacidade de processamento e relativa capacidade de comunicação e conectados à Internet através de um par de fios da rede telefônica convencional, o que lhe dá acesso moderado a uma rede relativamente confiável ([FOR1999]).

Segundo o WAP Fórum [FOR1999], dispositivos portáteis para redes sem fio possuem recursos de computação mais restritos se comparados a computadores:

- a) menor poder de processamento;
- b) pouca memória (RAM e ROM);
- c) consumo restrito de energia;
- d) telas pequenas (*display* matricial com poucas linhas);
- e) diferentes dispositivos de entradas de dados (como o teclado do telefone celular).

Redes móveis de dados apresentam um ambiente mais hostil de comunicação se comparado ao das redes com fios. Devido a limitações de potência, disponibilidade de espectro e mobilidade, segundo o WAP Fórum [FOR1999] as redes de dados sem fio tendem a possuir:

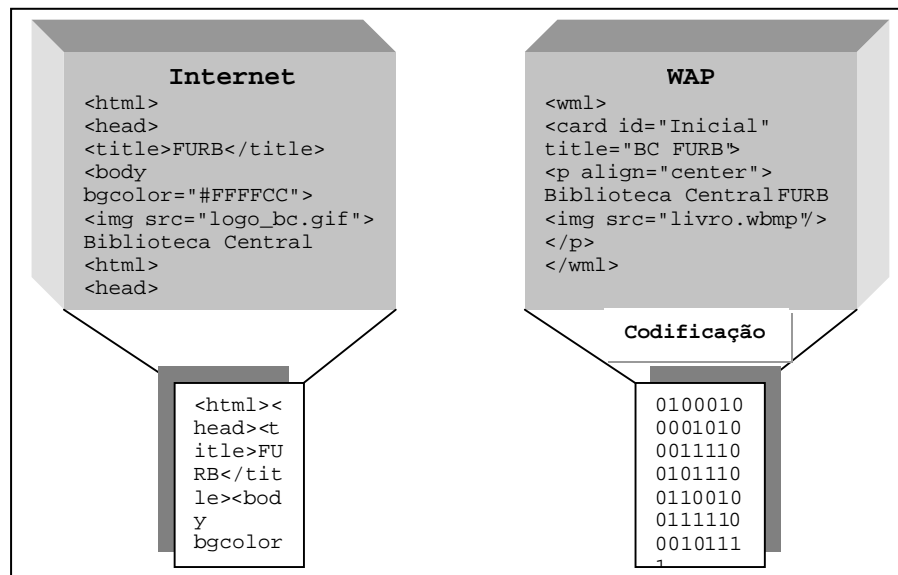
- a) menor banda passante;
- b) maior latência de transmissão de dados;
- c) menor estabilidade na conexão;
- d) menor disponibilidade.

Como consequência, a comunicação sem fio fica prejudicada: por exemplo, a máxima taxa para transmissão de dados utilizando um telefone celular não passa de 9,6Kbit/s na rede

telecomunicações GSM, enquanto um computador pessoal, utilizando conexão telefônica chega facilmente a 40 Kbit/s.

A especificação WAP deve, portanto, adequar seus protocolos e serviços a fim de minimizar as diferenças entre estes dois ambientes. Assim o protocolo WAP utiliza transmissão binária de dados (figura 1), e é aperfeiçoado para alta latência<sup>1</sup> e baixa largura da banda<sup>2</sup> da rede.

**Figura 1 – Codificação conteúdo via WAP**



Fonte: [WAP2000a]

### 3.4 MODELOS DE REDE

A pilha de protocolos WAP é análoga à usada hoje na Internet, com a diferença de ser otimizada ao máximo para tentar superar os obstáculos que os dispositivos sem fio oferecem.

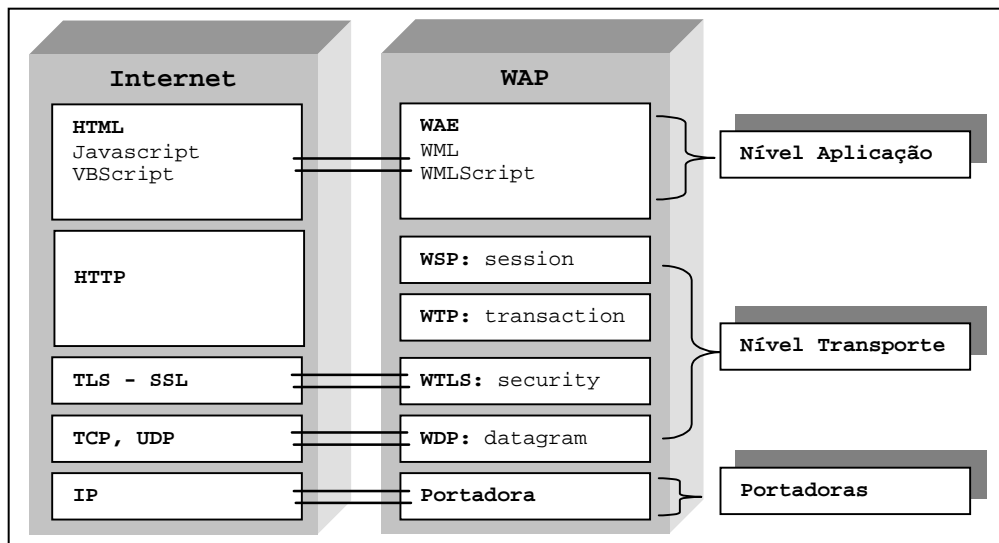
Os protocolos WAP cobrem tanto o nível de aplicação (WML e WMLScript) como o nível de transporte (WSP e WTP, WTLS e WDP). É bom salientar que existem vários tipos de tecnologia portadoras que podem carregar dados, como SMS, CDMA, entre outras, mas em qualquer uma delas o desenvolvimento de aplicações WAP se dará da mesma forma.

Sendo assim o celular acessa as páginas WAP, da mesma forma que o computador acessa as páginas da Internet. Na figura 2 é mostrado um comparativo dos protocolos do modelo Internet e do modelo WAP.

<sup>1</sup> tempo de sincronismo entre cliente e servidor.

<sup>2</sup> largura banda passante do canal de comunicação que é explorada.

Figura 2 - Comparativo protocolos Internet e WAP

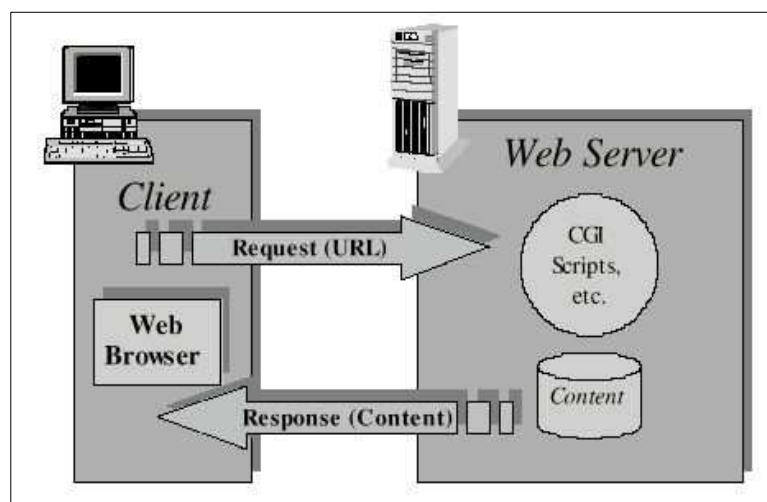


Fonte: [WAP2000b]

### 3.4.1 O MODELO WWW

A arquitetura da WWW (*World Wide Web*) baseia em um modelo flexível e estável para programação. Aplicações e conteúdos podem ser apresentados em formatos padrões e são acessados por aplicações conhecidas como navegadores. Os navegadores da Internet são aplicações de rede, isto é, eles mandam um pedido a um servidor e o servidor responde com uma informação codificada em formatos padrões, como mostrado na figura 3 ([FOR1999]).

Figura 3 - Modelo WEB



Fonte: [WAP2000b]

Segundo o WAP Fórum ([FOR1999]), os padrões WWW especificam muitos dos mecanismos necessários para construir um ambiente para um aplicação de uso geral, incluindo :

- a) **modelo de nomenclatura padrão** - todos os servidores e conteúdos na WWW são nomeados com o padrão Internet chamado *Uniform Resource Locator* (URL);
- b) **tipos diferentes de conteúdo** - a todos os conteúdos na WWW são atribuídos tipos específicos desse modo permitindo aos navegadores processar corretamente o conteúdo a partir do seu tipo;
- c) **formatos padrão de conteúdo** - todos os navegadores suportam um conjunto de formatos. Isso inclui o *HyperText Markup Language* (HTML), a linguagem *Javascript*, entre outros;
- d) **protocolos de comunicação padrão** - protocolos de rede padronizados permitem a qualquer navegador se comunicar com qualquer servidor de Internet. O protocolo utilizado na WWW é o *HyperText Transport Protocol* (HTTP).

Essa infra-estrutura permite que usuários alcancem um grande número de aplicações e serviços de conteúdo. Também permite aos desenvolvedores de aplicações uma maior facilidade de criar aplicações e serviços de conteúdo para um grande número de clientes.

Conforme o WAP Fórum [FOR1999], os protocolos da WWW são definidos em três classes de servidores :

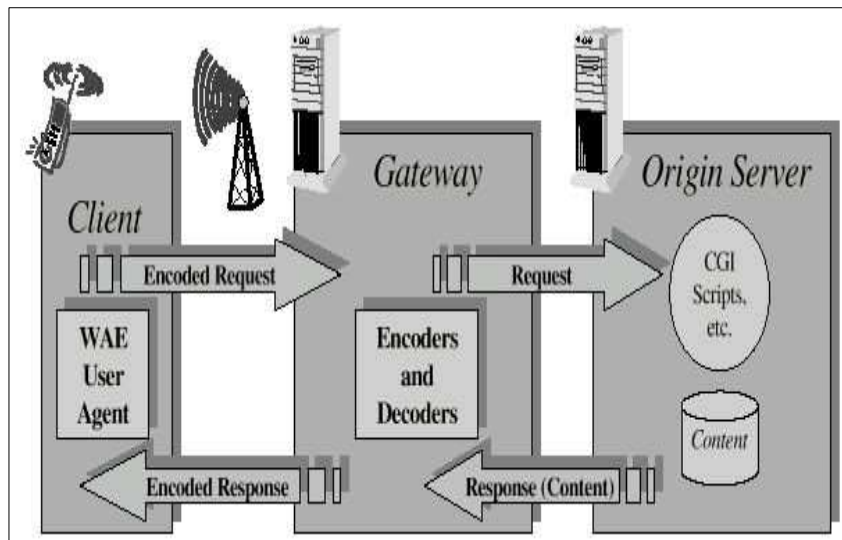
- a) **servidor de origem** – é o servidor onde um dado recurso (conteúdo) reside ou está para ser criado;
- b) **proxy** - um programa intermediário que atua tanto como um servidor quanto um cliente com o propósito de fazer pedidos em nome de outros clientes. O *proxy* tipicamente reside entre clientes e servidores que não possuem uma maneira de se comunicar diretamente, ex. através de um *firewall*. O *proxy* deve implementar os requisitos das especificações da WWW tanto do cliente como do servidor;
- c) **gateway** - um servidor que atua como um intermediário para outro servidor. Diferentemente do *proxy*, um *gateway* recebe pedidos como se fosse o servidor de origem de um recurso solicitado. O cliente que fez o pedido pode não estar ciente de que está se comunicando com um *gateway*.

### 3.4.2 O MODELO WAP

O modelo de programação do WAP é similar ao modelo WWW. Isto provê vários benefícios à comunidade de desenvolvimento, incluindo a familiarização com o modelo existente, a confiabilidade de uma arquitetura testada e a possibilidade de utilização de ferramentas existentes ([FOR1999]).

Entretanto, otimizações e extensões foram realizadas de forma a adaptar este modelo à realidade de ambientes móveis. Sempre que possível, padrões existentes foram adotados ou utilizados como base para desenvolvimento da tecnologia WAP podemos observar na figura 4 ([FOR1999]).

**Figura 4 - Modelo WAP**



Fonte: [WAP2000b]

O conteúdo e as aplicações WAP foram especificados em um conjunto de formatos bem conhecidos baseados nos formatos WWW. Os dados são transportados utilizando um conjunto de protocolos de comunicação baseados nos protocolos WWW. Um micro-navegador é o terminal que coordena a interface com o usuário e é análogo aos navegadores da Internet ([FOR1999]).

O WAP Fórum [FOR1999] definiu um conjunto de componentes que permitem a comunicação entre dois terminais móveis e os servidores da rede, incluindo :

- a) **modelo de nomenclatura padrão** - o padrão WWW de URLs foi utilizado para identificar o conteúdo WAP nos servidores de origem. Outro padrão WWW, o URIs, foi utilizado para identificar recursos locais num dispositivo, como funções



de controle de chamadas;

- b) **tipos diferentes de conteúdo** - a todos os conteúdos WAP são fornecidos tipos consistentes com os tipos do WWW. Isso permite aos agentes WAP dos usuários processar corretamente o conteúdo baseado no seu tipo;
- c) **formatos padrão de conteúdo** - Formato de conteúdo WAP são baseados na tecnologia WWW e incluem marcadores de *display*, informação de calendário, objetos de cartão eletrônico de negócios, imagens e linguagens *script*;
- d) **protocolos de comunicação padrão** - os protocolos de comunicação WAP permitem a comunicação de pedidos feitos pelo navegador de um terminal móvel ao servidor Internet da rede.

Os tipos de conteúdo WAP e seus protocolos foram otimizados para a utilização em ambientes móveis. Portanto, o WAP utiliza a tecnologia de *proxy* para conectar os dispositivos sem fio e a Internet. O *proxy* WAP é tipicamente responsável pelas seguintes funcionalidades conforme WAP Fórum [FOR1999]:

- a) **gateway de protocolo** - o *gateway* de protocolo traduz pedidos da pilha de protocolo WAP (WSP, WTP, WTLS, e WDP) para a pilha de protocolos WWW (HTTP e TCP/IP) ;
- b) **codificadores e decodificadores de conteúdo** - os codificadores de conteúdo traduzem o conteúdo WAP para formatos compactos para reduzir o tamanho dos dados na rede .

Essa infra-estrutura garante que usuários de um terminal móvel possam navegar em uma grande variedade de conteúdos e aplicações WAP, e que os desenvolvedores de aplicações possam construir serviços de conteúdo e aplicações que rodem numa larga base de terminais móveis. O *proxy* WAP permite que aplicações e conteúdo possam ser hospedados em um servidor WWW padrão e possam ser desenvolvidos utilizando-se tecnologias WWW já estabelecidas tais como CGI *scripting*, ASP (*Active Server Pages*).

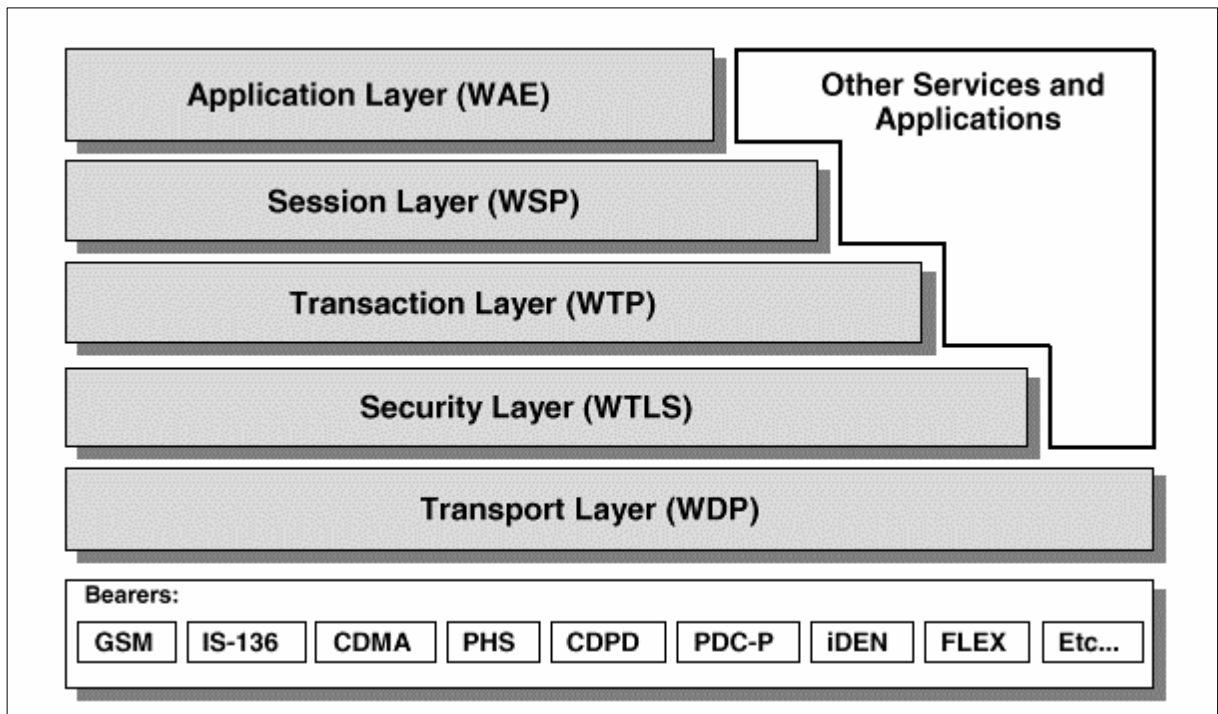
## 3.5 ARQUITETURA

Este item trata da estrutura de protocolos que compõem a tecnologia WAP.

### 3.5.1 COMPONENTES DA ARQUITETURA WAP

A arquitetura WAP fornece um ambiente extenso e escalonável para o desenvolvimento de aplicações de dispositivos de comunicação móvel. Isso é alcançado através do projeto de uma pilha inteira de protocolos (figura 5). Cada camada da arquitetura é acessível através da camada acima dela, tais como através de outros serviços e aplicações.

Figura 5 - Arquitetura WAP



Fonte: [WAP2000b]

A arquitetura em camada do WAP permite que outros serviços e aplicações possam utilizar as características da pilha WAP através de um conjunto de interfaces bem definidas. Aplicações externas podem acessar as camadas de sessão, transação, segurança e transporte diretamente. Nas seções seguintes serão mostradas as descrições das várias camadas da arquitetura WAP.

### 3.5.1.1 CAMADA DE APLICAÇÃO - WAE (WIRELESS APPLICATION ENVIROMENT)

O ambiente de aplicações sem fio (WAE) é um ambiente de aplicações de uso geral fundamentado numa combinação da WWW e tecnologias de telefonia móvel. O objetivo primário do WAE é estabelecer um ambiente interoperável que permita que várias operadoras e provedores de serviços possam criar aplicações e serviços que alcancem uma grande variedade de diferentes plataformas de uma maneira eficiente e vantajosa. O WAE inclui um micro-navegador que contém as seguintes funcionalidades conforme WAP Fórum [FOR1999]:

- a) *Wireless Markup Languge* (WML) - uma linguagem simples, similar ao HTML, porém otimizada para o uso em terminais móveis portáteis ;
- b) *WMLScript* - uma linguagem simples de *script*, similar ao *JavaScript*;
- c) *Wireless Telphony Application* (WTA,WTAI) - serviços de telefonia e interfaces de programação;
- d) formatos de conteúdos - um conjunto bem definido de formatos de dados, que inclui imagens, registro de telefones e informação de calendário.

### 3.5.1.2 CAMADA DE SESSÃO - WSP (WIRELESS SESSION PROTOCOL)

O protocolo de sessão (WSP) fornece à camada aplicação do WAP uma interface consistente para duas sessões de serviços. A primeira é um serviço orientado a conexão que opera acima da camada de transação (WTP). A segunda é um serviço que não necessita de conexão que opera acima do serviço de datagramas, com ou sem segurança (WDP).

O protocolo de sessão atualmente consiste de serviços próprios para aplicações de navegação (WSP/B). O WSP/B fornece as seguintes funcionalidades segundo WAP Fórum [FOR1999]:

- a) funcionalidade e semântica do HTTP/1.1 numa codificação compacta;
- b) suspensão e retomada de sessões;
- c) uma facilidade comum para um *push* (envio unidirecional), com ou sem confirmação, de dados;
- d) protocolo de negociação de capacidades.

A essência do projeto do WSP é uma forma binária do HTTP. Consequentemente, pedidos mandados a um servidor e respostas indo a um cliente podem incluir tanto cabeçalho

quando informação. Todos os métodos definidos pelo HTTP/1.1 são admitidos. Capacidade de negociação também pode ser usada para concordar com um conjunto estendido de métodos de pedidos, para garantir a compatibilidade com aplicações HTTP/1.1. O WSP fornece transferência de dados formatados para a camada de aplicação. Os cabeçalhos de conteúdo do HTTP/1.1 são usados para definir o tipo do conteúdo, codificação do conjunto de caracteres, linguagens, etc. Contudo, codificações binárias compactas são definidas para cabeçalhos conhecidos para reduzir o *overhead* de protocolo.

Uma sessão pode ser suspensa enquanto estiver ociosa de forma a liberar recursos da rede e economizar bateria. Um protocolo de reconexão permite que a sessão possa prosseguir sem o *overhead* de um novo estabelecimento de sessão. A sessão também pode ser retomada numa rede de diferentes portadores (*bearer*).

### **3.5.1.3 CAMADA DE TRANSAÇÃO - WTP (WIRELESS TRANSACTION PROTOCOL)**

O protocolo de transação (WTP) roda em cima do serviço de datagramas e fornece um protocolo leve orientado a transação que é adequado para a implementação em clientes com pouca capacidade de processamento (estações móveis). O WTP opera eficientemente em cima de datagramas de redes sem fio, seguros ou não, e fornece as seguintes características segundo WAP Fórum [FOR1999]:

- a) três classes de serviços de transação: pedidos unidirecionais não confiáveis, pedidos unidirecionais confiáveis e pedidos bi-direcionais confiáveis;
- b) segurança opcional entre usuário-usuário – envia a confirmação de recebimento de cada mensagem;
- c) concatenação de PDUs e reconhecimentos atrasados para reduzir o número de mensagens enviadas;
- d) transações assíncronas.

### **3.5.1.4 CAMADA DE SEGURANÇA - WTLS (WIRELESS TRANSPORT LAYER SECURITY)**

O WTLS é um protocolo de segurança inspirado no protocolo padrão *Transport Layer Security* (TLS), anteriormente conhecido como *Secure Sockets Layer* (SSL). O WTLS foi criado para ser usado com os protocolos de transporte WAP e foi otimizado para uso em canais de comunicações com pouca banda passante. Segundo WAP Fórum [FOR1999] o WTLS fornece as seguintes características :

- a) integridade de dados - O WTLS contém facilidades para assegurar que os dados enviados entre dois terminais e o servidor de aplicação continuem inalterados e não corrompidos;
- b) privacidade - O WTLS possui facilidades para assegurar que a informação transmitida através do terminal a um servidor de aplicação seja privada e não possa ser entendida por alguém que possa ter interceptado o fluxo de dados;
- c) autenticação - O WTLS facilita o estabelecimento da autenticidade do terminal e do servidor;
- d) proteção contra *Denial-of-service* - WTLS consegue detectar e rejeitar dados que foram duplicados ou não foram verificados com sucesso. WTLS faz com que muitos ataques típicos de *denial-of-service* sejam mais difíceis de serem executados e protege as camadas de protocolos acima dele.

WTLS pode ser usado também para comunicações seguras entre dois terminais como por exemplo para autenticação de transações com cartões eletrônicos de negócios.

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.

### **3.5.1.5 CAMADA DE TRANSPORTE - WDP (*WIRELESS DATAGRAM PROTOCOL*)**

A camada de transporte na arquitetura WAP é chamada de WDP. A camada WDP opera acima dos serviços das portadoras (*bearers*), suportado por vários tipos de redes. Como um serviço de transporte geral, o WDP fornece um serviço estável para a camada WAP de protocolo acima dele e comunica transparentemente com qualquer serviço de portadoras (*bearers*) disponíveis ([FOR2000]).

Os protocolos que constituem o WAP foram projetados para operar sobre uma variedade de serviços de portadoras, incluindo *short messages*, dados em pacotes e dados em circuitos chaveados. Os *bearers* oferecem diferentes níveis de qualidade de serviço com respeito a taxa de erro, atrasos e taxa de transmissão. Os protocolos do WAP foram projetados para compensar ou tolerar essas variações do nível de serviço.

Como a camada WDP fornece uma convergência entre os serviços de *bearer* e o resto da pilha WAP, a especificação WDP lista todos *bearers* que são suportados e as técnicas utilizadas para permitir que os protocolos WAP possam rodar em cima de cada *bearer*.

### 3.5.2 PORTADORAS - BEARER SERVICES

No WAP a transferência de dados é realizada por diferentes serviços de portadoras, incluindo *short message*, comutação por circuitos ou comutação por pacotes, implementado pelas diferentes tecnologias de rede sem fio.

Como as diversas redes possuem parâmetros de comunicação distintos como vazão, BER (taxa de erro de bit) e atraso, os protocolos do WAP devem compensar ou tolerar estas desigualdades de recursos.

Como exemplo de serviços de portadoras para a rede GSM (*Global System for Mobile Communication* - a principal rede sem fio digital, padrão "de fato" na Europa e Ásia e adotada para a banda "C" no Brasil), podemos citar:

- a) SMS (*Short Message Service*), serviço caracterizado pela transferência de mensagens curtas (com até 160 caracteres), com confirmação de entrega. As mensagens podem conter palavras alfa-numéricas ou não textuais (binárias);
- b) HSCSD (*High-Speed Circuit Switched Data*), para comutação por pacotes;
- c) GPRS (*General Packet Radio Service*), para comutação por circuitos.

## 3.6 LINGUAGENS WAP

Foram criadas linguagens de programação específicas para o WAP parecidas com as linguagens atuais para Internet fixa. O WAP Fórum [FOR1999] produziu duas linguagens que permitem a criação de interfaces para navegação WAP.

A primeira é a WML (*Wireless Mark-up Language*), uma linguagem baseada em XML (*Extensible Mark-up Language*), é constituída por um sub-conjunto de funcionalidades do HTML. Esta é a linguagem de programação utilizada para criação do conteúdo de sites WAP, e permite a navegação entre os documentos WML.

Para a criação de páginas dinâmicas o WAP Fórum especificou o *WMLScript*, parecido com o *JavaScript*. Esta linguagem possui as mesmas características presentes nas linguagens

*scripts* utilizados nos navegadores convencionais, tais como *Javascript*, *VBScript*. Ela permite, por exemplo, que o dispositivo realize validações de entrada.

### 3.6.1 WIRELESS MARKUP LANGUAGE - WML

O WML é a linguagem de programação utilizada para a criação do conteúdo dos sites WAP. Trata-se de uma linguagem de “Markup” baseado na família XML, que herdou as características do HDML e HTML.

A linguagem WML foi projetada para fazer bom uso da faixa estreita de transmissão de dados disponível no ambiente de redes sem fio, assim como do pequeno *display* dos terminais móveis. Como parte deste contexto, criaram-se os conceitos de *deck* e *card*.

Um simples documento WML, ou seja, um documento com elementos contidos dentro de um par de *tags* `<wml>`, é conhecido como um *deck* (baralho), representando uma aplicação ou um serviço.

Um *deck* contém um ou vários *cards* (cartas). Um *card* é definido como uma unidade básica de *display* no WAP. Um *card* pode ser visto como uma tela de informação. Caso se disponha de um terminal móvel dotado de visor com apenas uma linha o usuário terá que utilizar barras de rolagem para ler um único *card*.

Um *card* pode conter texto, imagens, *hiperlinks* e campos de entrada. Campos de entrada podem ser da forma de botões, múltipla escolha ou do tipo caixa de entrada.

Um *deck* é similar a uma página HTML uma vez que é identificado por um URL. Assim sendo, um *deck* é uma unidade de conteúdo de transmissão, que é enviada integralmente do servidor WAP para o terminal móvel. A novidade neste contexto é que várias telas (*cards*) podem ser baixadas para um cliente de uma só vez.

Um *deck* não deve exceder a 1400 bytes, o que não corresponde ao tamanho de arquivo WML e sim um arquivo binário codificado a ser transmitido, obtido a partir do arquivo fonte WML.

Para usar caracteres especiais na linguagem WML, é necessário usar o seu código equivalente da tabela Latin-1 (ISO 8859-1) podemos ver no anexo II. Se quiser utilizar a palavra Endereço por exemplo, e usar o cedilha, o resultado não será o esperado pois a

palavra sairá escrita de forma incorreta. Para que saia escrita da forma correta, é necessário escrevê-la assim: Endere#231;o.

Em WML, imagens são de um tipo especial chamado WBMP (*Wireless BitMap*). São imagens em preto e branco ou coloridas com até 150x150 pixels. É importante notar ainda que nem todos terminais WAP serão capazes de exibir imagens (deve-se usar a diretiva opcional "alt", como no HTML, para que tais terminais apresentem uma legenda no lugar da imagem).

### 3.6.2 WMLSCRIPT

A especificação WAP define também uma linguagem de *script* chamada *WMLScript*, semelhante ao *JavaScript* da Internet fixa. *WMLScripts* são funções criadas por provedores de conteúdo que podem ser invocadas por nomes, aceitam parâmetros e retornam valores.

Com *WMLScript*, um *card* pode por exemplo executar funções que verifiquem se todos os campos de entrada estão preenchidos ou que transmitam informação entre *cards* ou entre *decks*.

Utilizando *WMLScript*, seleções do usuário (ou entradas) podem ser manipuladas e encaminhadas a *cards* já carregados, eliminando a excessiva transação entre clientes e servidores remotos.

## 3.7 PÁGINAS DINÂMICAS EM WML

O uso de páginas estáticas para o controle/apresentação de informações ao usuário mostra-se impraticável, ao ponto que restringe as informações e serem estáticas, e a necessidade de constantes atualizações que se fazem necessárias no dia-a-dia das páginas.

Por meio de linguagens de *script* ( ASP, PHP ) e CGI, é possível criar essas páginas com o seu conteúdo dinâmico de acordo com as condições apresentadas ao usuário.

Uma linguagem *Script* pode ser entendida como uma página que é interpretada pelo provedor, devolvendo um formato HTML de acordo com a requisição de um browser cliente.

O WAP tenta adaptar as tão bem sucedidas tecnologias adotadas pela Web. Possui a linguagem WML que é muito parecida com o HTML e que possui uma linguagem chamada *WMLScript* que tem algumas características do *JavaScript*.



Também pode-se utilizar as linguagens de *Script* CGI e ASP, já que o WML funciona como HTML (a única diferença é no retorno dos *tags*). Quando trabalha-se com *Script*, o servidor em vez de devolver um página HTML devolverá uma página WML.

Para se criar uma aplicação WML dinâmica é necessário que a linguagem utilizada sinalize para o browser o cabeçalho padrão utilizado.

### 3.7.1 ASP E WML

O ASP (*Active Server Pages*) é um ambiente de programação interpretado por *scripts* no servidor, onde pode-se criar páginas dinâmicas interativas e de alta performance. As páginas, são executadas no servidor e não no cliente. Quando o cliente solicita a página, o servidor interpreta seu conteúdo e devolve somente o resultado. Um arquivo que contém código ASP é identificado pela extensão “.asp” ([RAD2000]).

Pode-se utilizar ASP para criar páginas dinâmicas WML, bastando alterar o cabeçalho padrão utilizado para WML.

As *Active Server Page* contêm duas partes: código programático e WML embutido. O código programático pode ser escrito em uma *linguagem script*.

Para utilizar-se ASP com WML, precisamos de um servidor que interprete páginas ASP, como o *Internet Information Services* (IIS) do Windows 2000, e configurá-lo para apresentar as páginas WML.

## 3.8 APLICAÇÕES

Quais seriam as necessidades de um usuário de comunicação móvel à Internet? Com certeza, não serão as mesmas da Internet fixa. Tem-se que disponibilizar conteúdos personalizados e úteis aos usuários da comunicação móvel.

Os primeiros negócios que despontam na utilização do WAP são relacionados ao mercado financeiro, como permissão de acesso, cotações, informações econômicas e realiza de transações, como transferências, pagamentos de contas, investimentos, saldos, extratos, etc ([MOB2000]).

Outros negócios bastante promissores podem ser todos aqueles relacionados ao conteúdo de portais já existentes, contendo informações do dia, roteiros, guias culturais,

boletins de trânsito, meteorologia e manchetes, que são fáceis de serem adaptadas a tecnologia WAP ([MOB2000]).

Em teoria, todo site que oferece algum tipo de serviço ao internauta pode ter sua versão WAP, bastando entrar nesse segmento de forma planejada.

Afirma Anna Flavia Ribeiro em [MOB2000] , que o grande negócio do WAP é o comércio eletrônico. Compras on-line com ofertas sendo lançadas no celular, estão começando a brotar, e a tendência é que se firmem como um grande negócio.

Aplicações mais comuns no WAP segundo [MOB2000]:

- a) e-mail: fundamentais para executivos e/ou qualquer profissional;
- b) e-banking: cresce rapidamente pela boa relação custo-benefício e pela possibilidade de redução de custos nas transações bancárias;
- c) transações-financeiras: possibilidade de tomada de decisão em tempo real;
- d) agenda: organização da vida pessoal, permitindo um melhor controle de horários e compromissos;
- e) busca: ferramenta facilitadora no dia-a-dia;
- f) consulta de preços: facilita decisões de compras;
- g) reservas: passagens, hotéis, restaurantes;
- h) notícias: manchetes e resumos de indicadores e fatos;
- i) empresariais: ferramentas para maior agilidade nas transações, com redução de custos.

## **4 DESENVOLVIMENTO DO PROTÓTIPO**

Neste capítulo é apresentada a funcionalidade do protótipo desenvolvido, modelagem, ferramentas utilizadas e é demonstrado o protótipo.

### **4.1 ESPECIFICAÇÃO**

O protótipo disponibiliza os serviços de consulta ao acervo, reserva e verificação de reservas do usuário junto ao serviço de empréstimo da Biblioteca Central através de celulares e aparelhos com tecnologia WAP.

A metodologia utilizada no desenvolvimento da especificação do sistema foi a análise estruturada.

#### **4.1.1 FERRAMENTA UTILIZADA**

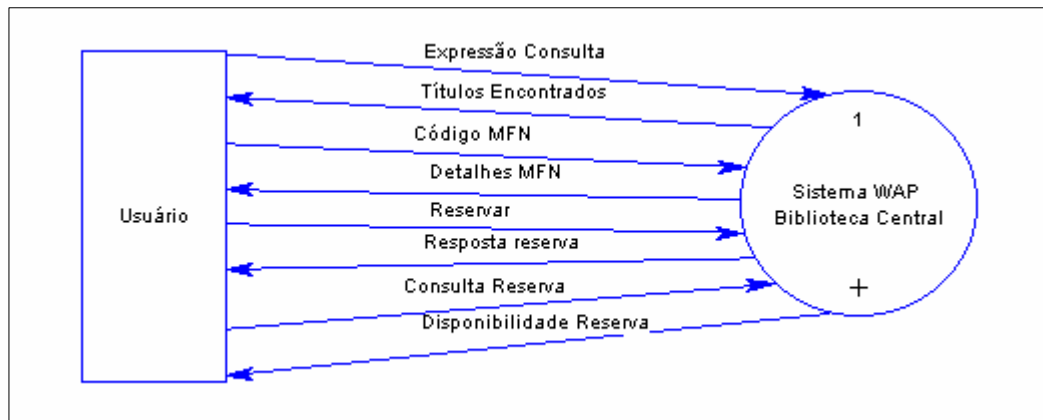
Para especificação do protótipo foi utilizada a ferramenta Powerdesigner. Foram criados o diagrama de contexto, e o modelo entidade relacionamento (MER) das tabelas da base de dados da Biblioteca Central utilizadas no sistema.

É uma ferramenta Case utilizada para especificação de projetos de software. Temos 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 utilizá-lo para criar novas estruturas de dados ou fazer engenharia reversa das estruturas de dados já existentes.

#### **4.1.2 DIAGRAMA DE CONTEXTO**

No diagrama de contexto (figura 6) apresentam-se a entidade externa envolvida no Sistema WAP da Biblioteca Central. Tem-se como entidade externa somente o usuário pois o sistema tem como objetivo oferecer acesso ao usuário aos serviços prestados.

**Figura 6 - Diagrama de Contexto**



### 4.1.3 MODELO ENTIDADE RELACIONAMENTO

Na figura 7 apresenta-se o seu modelo entidade relacionamento (MER) das tabelas utilizadas no sistema.

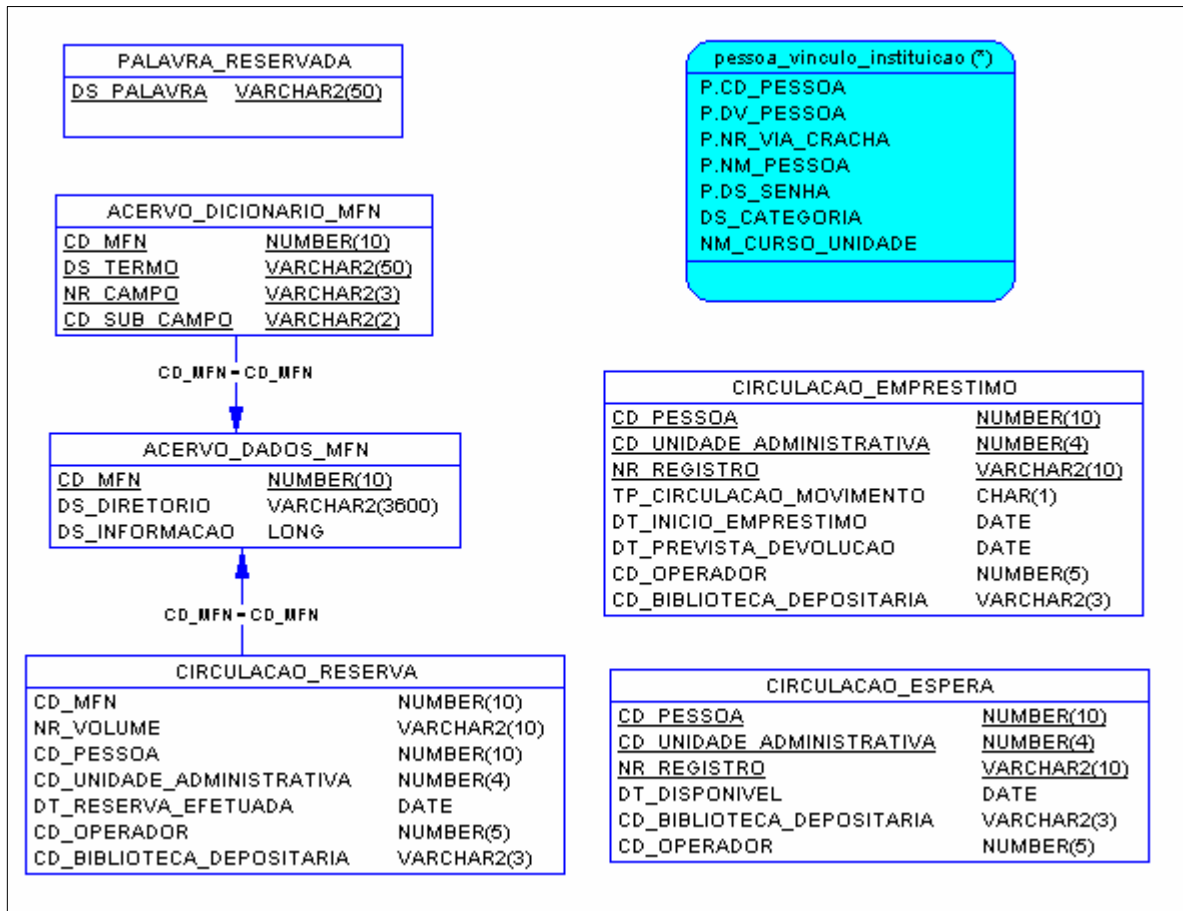
A base de dados da Biblioteca Central está implantada em *Oracle 8* para *Netware*.

As informações inseridas na base de dados estão estruturadas de acordo com o Formato USMARC (Catalogação legível por computador – Padrão Norte-Americano) que é o padrão adotado na Rede Bibliodata-Calco, da qual a Biblioteca participa desde 1988.

A Rede Biblioteca-Calco é uma rede de catalogação cooperativa que reúne aproximadamente 120 Bibliotecas de 70 instituições brasileiras (Universidades, Bibliotecas Públicas, Colégios Militares, Órgãos Públicos, etc.), constituindo-se a principal rede de bibliotecas em atividade no País.

MFN significa *Master File Number*, é o numero que identifica um registro no CDS/ISIS.

Figura 7 - modelo entidade relacionamento (MER)



A estrutura das tabelas da figura 7 é composta pelos seguintes módulos:

- PALAVRA\_RESERVADA:** está tabela armazena todas as palavras reservadas, ou seja, palavras que serão descartadas em um expressão de busca. Também quando indexa a base as palavras reservadas são descartadas. Exemplos de palavras reservadas: AND, WITH, DE, DA;
- ACERVO\_DADOS\_MFN:** é a tabela principal do sistema, que armazena os dados de todo o acervo da biblioteca. Temos os seguintes campos:
  - CD\_MFN - que identifica o material;
  - DS\_DIRETORIO – contém os parágrafos do MFN , sua localização dentro do campo; DS\_INFORMACAO e o tamanho da informação;
  - DS\_INFORMACAO – contém todas as informações do MFN;
- ACERVO\_DICIONARIO\_MFN:** esta tabela é usada para fazer a busca no acervo, e contém os termos indexados da tabela ACERVO\_DADOS\_MFN;
- CIRCULACAO\_RESERVA:** contém as reservas de materiais feitas pelos usuários, pois reserva-se o material pelo seu CD\_MFN para que, ao chegar o primeiro

exemplar daquele MFN, este será reservado. Um MFN pode conter um ou mais exemplares;

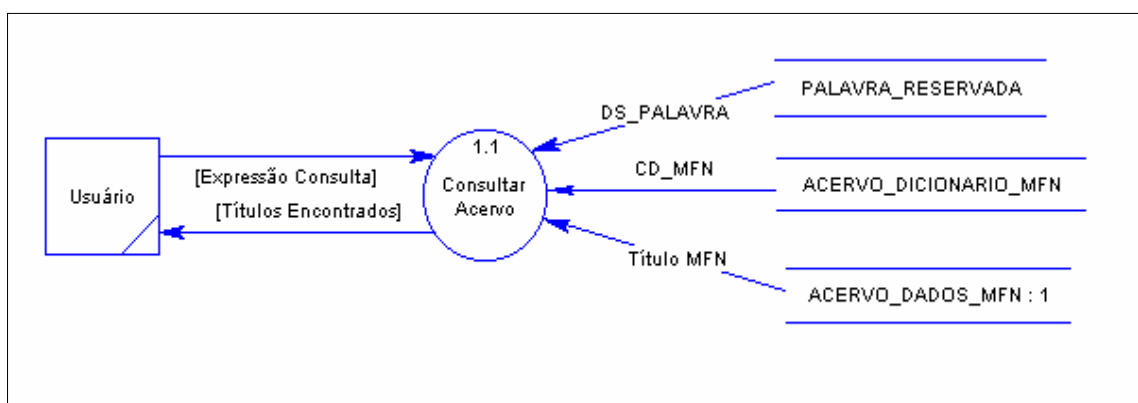
- e) CIRCULACAO\_EMPRESTIMO: controla todos os empréstimos realizados aos usuários, identificando se o material está disponível;
- f) CIRCULACAO\_ESPERA: controla material em espera. Quando um material é devolvido e está reservado ele passa ao estado de espera;
- g) PESSOA\_VINCULO\_INSTITUICAO: esta é uma *view* gerada através das tabelas do sistema acadêmico da FURB onde são filtrados somente os campos relevantes a biblioteca para efetuar empréstimos aos usuários. Contém dados de todos os usuários da Biblioteca Central da FURB.

#### 4.1.4 DIAGRAMAS DE FLUXO DE DADOS

Nos diagramas de fluxo de dados, para cada evento pode-se observar as principais funcionalidade do sistema.

**Consultar acervo** (figura 8). O usuário passa o tipo de busca e a expressão de busca ao sistema, o sistema ignora as palavras encontradas na tabela PALAVRA\_RESERVADA, monta uma expressão de busca SQL e executa em cima da tabela ACERVO\_DICIONARIO\_MFN que retornara os códigos dos MFN encontrados, com os quais serão recuperados os títulos na tabela ACERVO\_DADOS\_MFN e mostrados ao usuário.

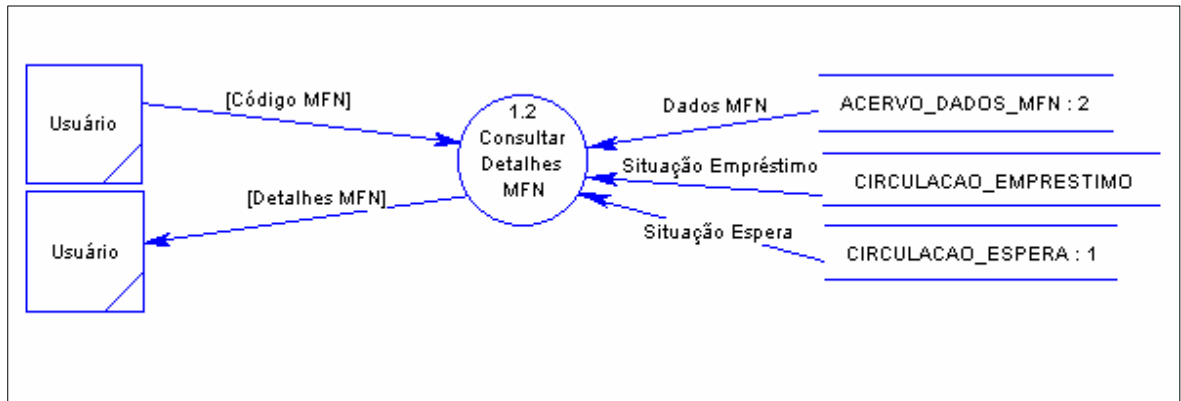
**Figura 8 - Consultar Acervo**



**Consultar detalhes do MFN** (figura 9). Quando o usuário seleciona um título para verificar detalhes, é passado como parâmetro ao sistema o CD\_MFN e o sistema monta uma busca em cima da tabela ACERVO\_DADOS\_MFN para retornar os dados do MFN como

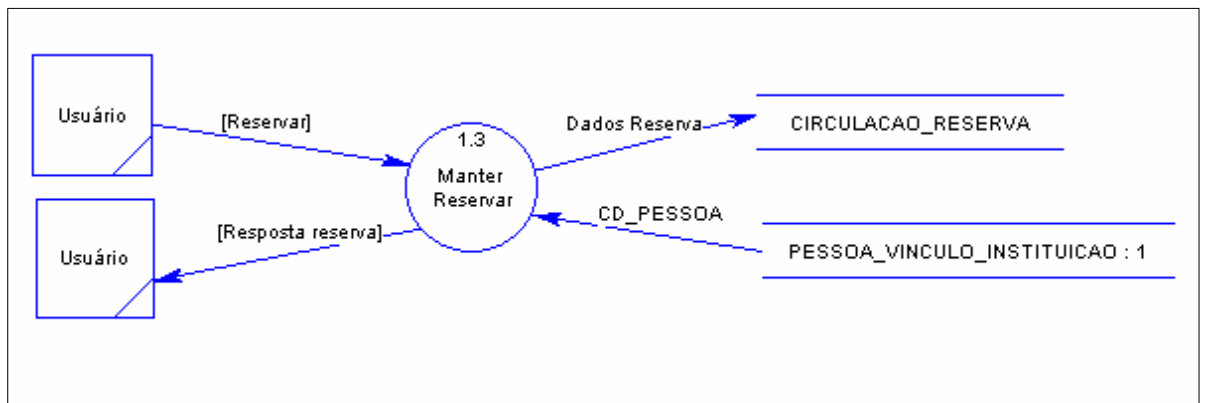
autor, classificação, registros. Depois consulta as tabelas CIRCULACAO\_EMPRESTIMO e CIRCULACAO\_RESERVA para verificar a disponibilidade do material.

**Figura 9 - Consultar detalhes MFN**



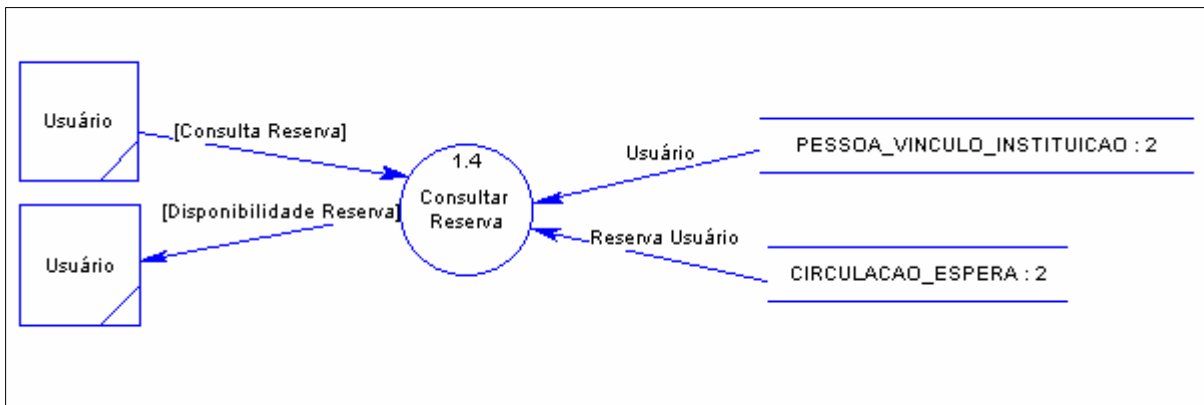
**Manter reserva MFN** (figura 10). Quando o usuário consulta detalhes do MFN será recuperada dados e a disponibilidade do material. Se o material não estiver disponível, o usuário pode reservá-lo. Para efetuar uma reserva o usuário passa ao sistema o CD\_MFN e CD\_PESSOA, o sistema verifica o código, insere um registro na tabela CIRCULACAO\_ESPERA com a reserva do usuário e retorna uma resposta ao usuário.

**Figura 10 - Manter reserva**



**Consultar reserva** (figura 11). Quando o usuário solicita ao sistema para verificar se a reserva solicitada está a sua espera passa o CD\_PESSOA, o sistema verifica se o código é válido e monta uma busca na tabela CIRCULACAO\_ESPERA para verificar se o material já está disponível e responde ao usuário dizendo se o material está ou não disponível.

Figura 11 - Consultar reserva



## 4.2 IMPLEMENTAÇÃO

Neste item são descritas as ferramentas utilizadas para a implementação, e uma apresentação do protótipo.

### 4.2.1 TÉCNICAS E FERRAMENTAS UTILIZADAS

Para implementação do protótipo foram utilizadas as ferramentas EasyPad Waptor, UP. Simulador, servidor IIS do Windows 2000 e as linguagens de programação WML e ASP.

O EasyPad Waptor foi utilizado para editar as páginas WML e *scripts* ASP.

O servidor IIS do Windows 2000 foi utilizado como servidor Web para executar os *scripts* ASP.

O UP Simulador foi utilizado para emulador o aparelho celular no teste das páginas WML.

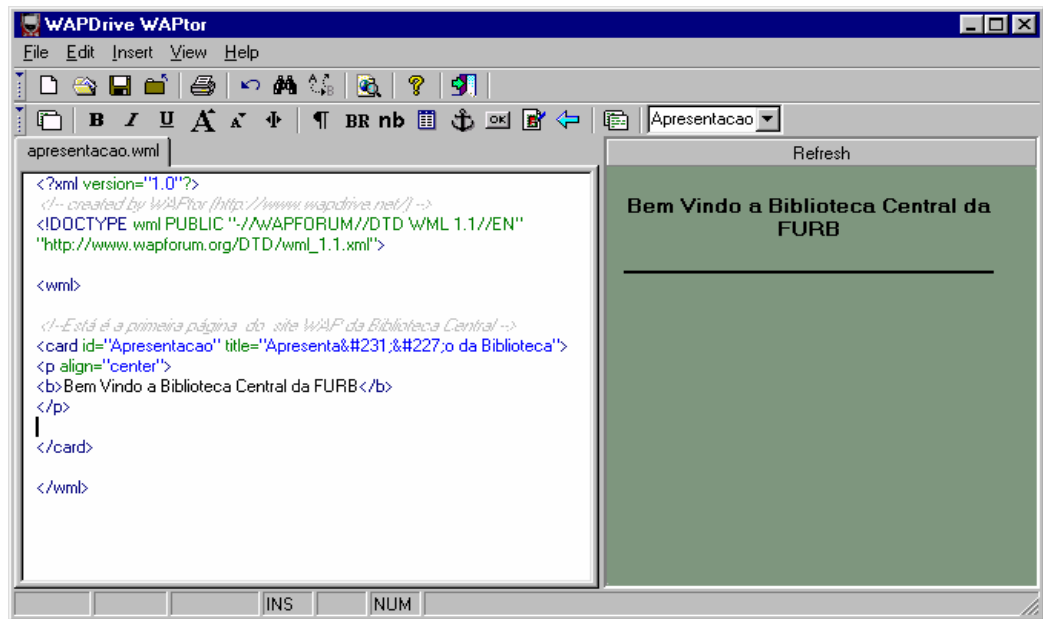
#### 4.2.1.1 EASYPAD WAPTOR

O EasyPad Waptor (Figura 12) é um editor WML e permite visualizar o resultado do código que está sendo criado, possuindo vários recursos visuais de *tags* WML.



A janela do EasyPad Waptor possui duas partes principais, a janela de edição do código a esquerda, onde pode-se editar o código fonte WML ou inserir *scripts* ASP e, a direita, a janela de visualização do resultado da página, que só apresenta o resultado do código WML.

Figura 12 - WAPDrive WAPtor



#### 4.2.1.2 UP SIMULADOR

O UP Simulador é um emulador de celular da Phone.com. onde pode-se testar páginas WML. Verificar os resultados e depurar o código.

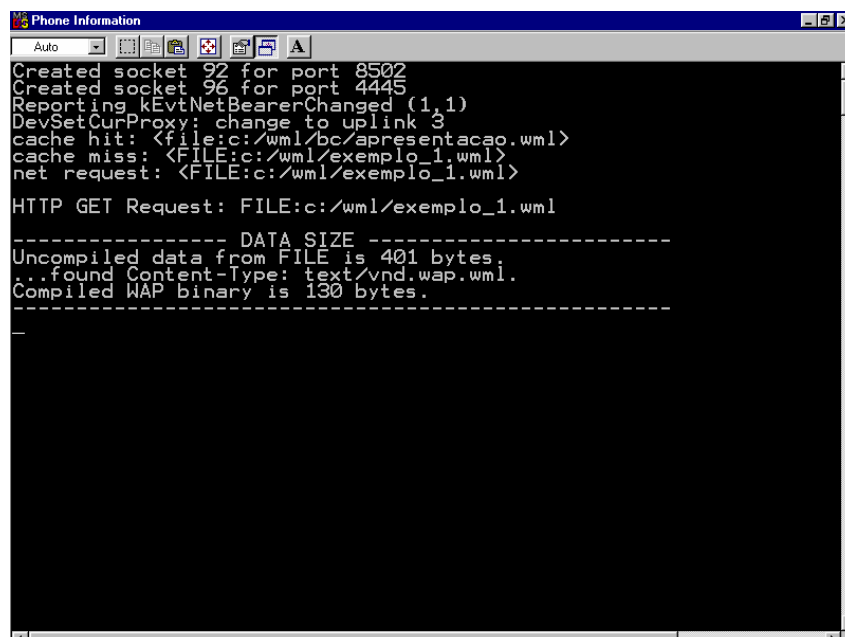
Temos a janela principal (figura 13) onde aparece o resultado da página WML.

Figura 13 - UP Simulador



Na figura 14 aparece a janela secundária onde é mostrado o tamanho da página convertida de WML para binário e possíveis erros no código .

Figura 14 - Debug UP Simulador



### 4.2.1.3 O SERVIDOR IIS WINDOWS 2000 – INTERNET INFORMATION SERVICES

O IIS é o servidor Web do Windows 2000 que foi utilizado para executar as páginas WML e ASP. Para que um servidor Web reconheça o conteúdo como definido nas especificações WAP, é importante configurá-lo corretamente com os tipos MIME (*Multipurpose Internet Mail Extensions*) apropriados, que são mostrados no quadro 1:

**Quadro 1 – Configuração Mime**

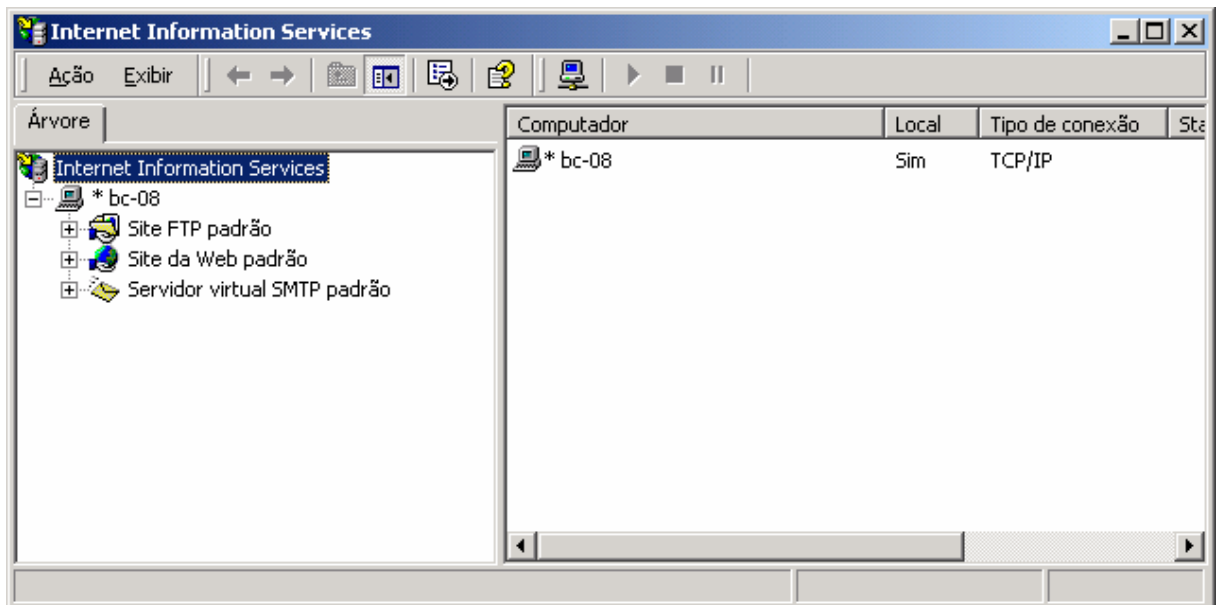
Tipo de Arquivo	Extensão	MIME Type
Código-fonte WML	.wml	text/vnd.wap.wml
WML compilado	.wmlc	application/vnd.wap.wmlc
Código-fonte WMLScript	.wmls	text/vnd.wap.wmlscript
WMLScript compilado	.wmlcs	application/vnd.wap.wmlscriptc
Imagem bitmap	.wbmp	image/vnd.wap.wbmp

Fonte: [DIA2000]

Passos para configuração do IIS 5.0 (Windows 2000):

- a) selecione o servidor no Gerenciador do *Internet Services Manager*;
- b) clique com o botão direito sobre o servidor Web Site e selecione a opção Propriedades no menu. A caixa de diálogo Propriedades do servidor aparecerá;
- c) na caixa mapeamento MINE do computador clique em botão Editar;
- d) aparecerá a janela Tipos de Arquivos, onde entrará com os novos tipos MINE;
- e) clique no botão Novo Tipo;
- f) na janela Tipo de Arquivo entre com a Extensão associada e o Tipo de Conteúdo (MIME) respectivo, contidos no quadro 1, clique em Ok para confirmar.
- g) repita os procedimentos e f para os outros tipos MIME e extensões, contidos no quadro 1;
- h) reinicialize o servidor.

Figura 15 – IIS Windows 2000



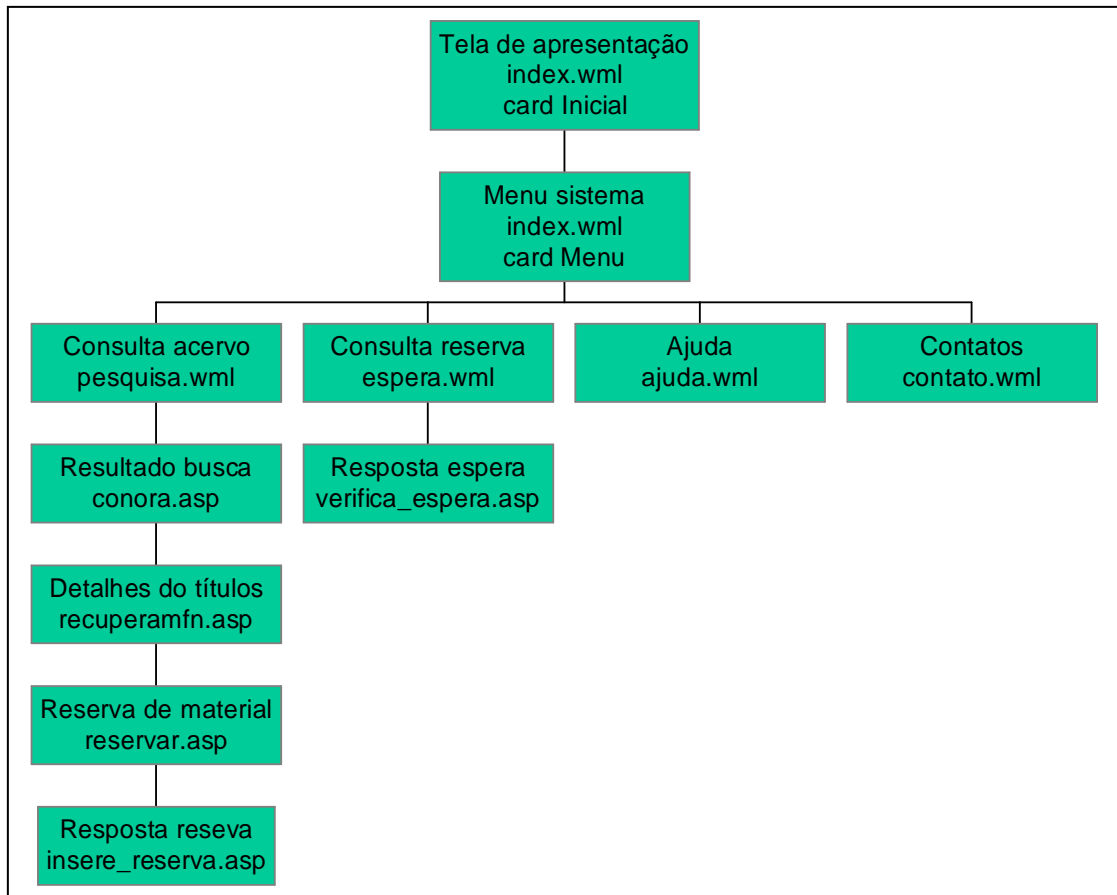
#### 4.2.2 OPERACIONALIDADE DA IMPLEMENTAÇÃO

Este item trata da operacionalidade da implementação como esta estrutura e como estão organizados os arquivos. Todos os arquivos fontes estão listados no anexo I.

Criou-se uma classe chamada MFN (classmfn.asp) para facilitar o acesso e recuperação dos dados do MFN e uma biblioteca (lib.inc) com as funções utilizadas.

Na figura 16 é mostrado o macro fluxo do sistema.

Figura 16 – Macro fluxo sistema



Explicar-se a funcionalidade de cada item do macro fluxo (figura 16) abaixo:

**Tela apresentação**, tela inicial do sistema contém a identificação do sistema, é apresentada quando o usuário entra no sistema. Esta implementada no arquivo index.wml, é gerada pelo *card* inicial.

**Menu sistema**, tela menu contém as opções do sistema, é apresentada quando o usuário seleciona menu na tela de apresentação ou entrara automaticamente após 50 segundos. Esta implementada no arquivo index.wml, é gerada pelo *card* menu. Faz chamada as páginas pesquisa.wml, espera.wml, ajuda.wml, contato.wml.

**Consulta acervo** realizam a busca no acervo da biblioteca, esta implementada no arquivo pesquisa.wml, esta tela é gerada pelo *deck* pesquisa. A página pesquisa.wml faz chamada à página conora.asp que realiza consulta ao acervo passando como parâmetro o tipo de busca (TpBusca) e a expressão de busca (ExpBusca).

**Resultado busca** retorna os títulos encontrados na base de dados da biblioteca. Esta implementada no arquivo conora.asp, realiza a consulta no acervo e retorna os títulos encontrados na tela. Utiliza a função Monta\_ExpBusca da biblioteca de funções lib.inc, que recebe como parâmetro o tipo de busca (TpBusca) e a expressão de busca (ExpBusca) do usuário, descartando as palavras reservadas, e devolve uma expressão SQL que executa uma busca na tabela dicionário de dados e verifica se existem registros. Se existirem registros, recupera o título dos registros utilizando o método Recupera\_Titulo da classe MFN que se encontra no arquivo classmfnc.asp.

A função Monta\_Expbusca recebe como parâmetro o tipo de busca e a expressão de busca desmonta a expressão de busca utilizada a função Verifica\_Reservada, para retirar as palavras reservadas e a função Monta\_SQL formata uma expressão SQL que retorna como resultado da função.

A função Verifica\_Reservada recebe como parâmetro uma palavra, verifica se a palavra é reservada ou não fazendo uma consulta na tabela PALAVRA\_RESERVADA.

A função Monta\_SQL recebe como parâmetro uma expressão de busca inicial, um termo a ser buscado e o tipo de busca a ser realizado. Verifica se existe uma expressão inicial se não existir cria e se existir concatena o novo termo de busca.

A procedure Recupera\_Titulo é uma função membro da classe MFN, responsável por recuperar o título do material e apresentar ao usuário.

**Detalhes título** esta tela retorna os detalhes de um título selecionado pelo usuário. Esta implementada no arquivo recuperamfn.asp Recebe como parâmetro o código do MFN (CodMFN) utilizando o método Recupera\_MFN da classe MFN que se encontra no arquivo ClassMFN.asp para recuperar os dados do MFN.

Recupera\_MFN é uma função membro da classe MFN, recupera os dados sobre o material, a disponibilidade e apresenta ao usuário.

**Reserva de material** esta tela pede a identificação do usuário. Está implementada no arquivo reservar.asp, recebe como parâmetro o código do MFN (CodMFN) chama a página insere\_reserva.asp e passa como parâmetro o código do MFN (CodMFN) e o código do usuário (CdPessoa).

**Resposta reserva** esta tela apresenta uma resposta ao usuário da reserva efetuada. Está implementada no arquivo `insere_reserva.asp`, recebe como parâmetro `CodMFN` e `CdPessoa`, chama a função `Verifica_CodigoPessoa` verificando se o código do usuário é válido. Se for válido insere a registro na tabela `CIRCULACAO_RESERVA` com a reserva do usuário, e responde ao usuário informando que a sua reserva foi efetuada. Se não for válido responde ao usuário informando que o código é inválido.

**Consulta reserva** solicita a identificação do usuário para realizar uma consulta verificando se a reserva do usuário já chegou. Esta página faz chamada a página `verifica_espera.asp` e passa como parâmetro o código do usuário (`CdPessoa`).

**Resposta espera** está página recebe como parâmetro o código do usuário (`CdPessoa`) e verifica se o código é válido através da função `Verifica_CodigoPessoa`. Se o código for válido, consulta a tabela `CIRCULACAO_ESPERA` verificando se existem livros a espera do usuário.

**Ajuda** está tela apresenta informações do funcionamento e opções de ajuda. Esta implementada no arquivo `ajuda.wml`.

**Contatos** está tela apresenta formas de entrar em contato com a Biblioteca Central. Esta implementada no arquivo `contatos.wml`.

### 4.3 APRESENTAÇÃO DO PROTÓTIPO

Este item apresenta a visualização do protótipo desenvolvido, com uma seqüência coerente de utilização do mesmo passo a passo.

A seguir são apresentadas as telas do protótipo e uma breve descrição delas.

Figura 17 - Tela de apresentação

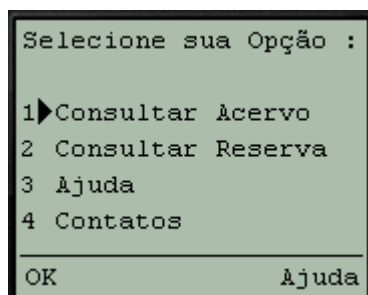


Na tela de apresentação (figura 17) o usuário pode selecionar uma de duas opções, clicando no botão menu entrará na tela de menu de serviços ou clicando em ajuda entrará na tela de ajuda do sistema. Se não selecionar nenhuma das opções o sistema entrará automaticamente na tela de menu após 50 segundos.



Nas demais telas serão apresentadas somente a parte da tela do celular.

**Figura 18 - Menu sistema**

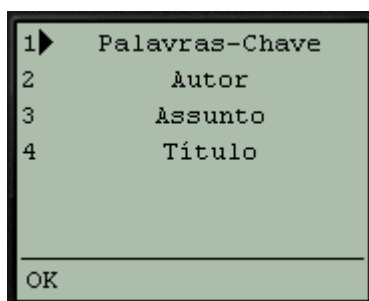


Na tela de menu do sistema (figura 18) o usuário deve selecionar uma das opções e clicar em OK para ter acesso ao serviço. Tem-se as opções :

- a) consultar acervo: que realiza busca no acervo;
- b) consultar reserva: verifica se já chegou a reserva do usuário;
- c) ajuda: traz dicas de como utilizar o sistema;
- d) contatos: apresenta formas de entrar em contato com a biblioteca.

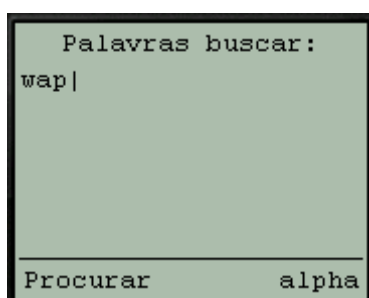
A seguir é apresentada a funcionalidade da opção consultar acervo.

**Figura 19 - Tipo de busca**



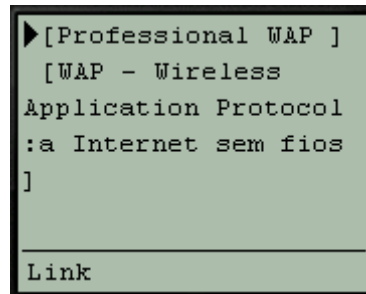
Na tela tipo de busca (figura 19) o usuário selecionara qual o tipo de busca a ser realizada.

**Figura 20 - Expressão de busca**



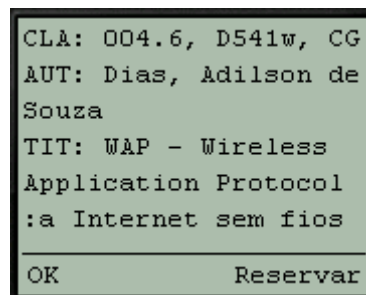
Na tela expressão de busca (figura 20) o usuário digitara a expressão de busca a ser realizada, deve clicar em procurar para realizar a consulta no acervo. A expressão de busca pode ser uma palavra ou uma frase.

**Figura 21 - Resultado busca**



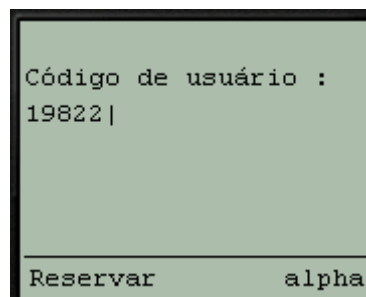
A tela de resultado busca (figura 21) retornará os títulos encontrados no acervo, o usuário deve selecionar o título de interesse e clicar em link para apresentar detalhes do título e a disponibilidade.

**Figura 22 - Detalhes do título**



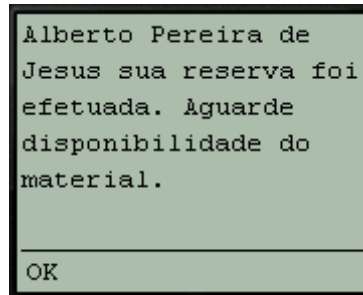
A tela detalhes do título (figura 22) retorna a classificação do material, tipo do material, autor, título, editora, ano de publicação e situação do material se esta ou não disponível. Se não existirem exemplares disponíveis o usuário pode reservar clicando em reservar.

**Figura 23 - Reserva de material**



Na tela de reserva de material (figura 23) o usuário deve entrar com seu código de usuário da biblioteca e clicar em reservar para efetuar a reserva.

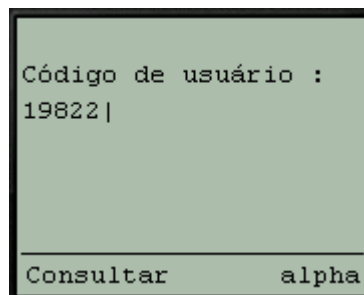
**Figura 24 - Reposta reserva**



```
Alberto Pereira de
Jesus sua reserva foi
efetuada. Aguarde
disponibilidade do
material.
OK
```

Na tela de resposta da reserva (figura 24) se o código do usuário for válido, insere o registro de reserva e retorna uma resposta ao usuário. Se o código do usuário não for válido apresenta uma mensagem identificando que o código do usuário é inválido

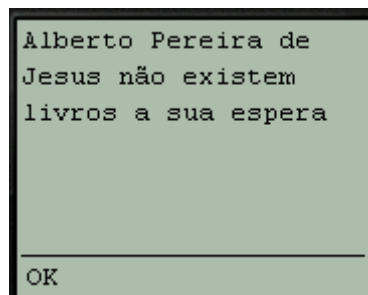
**Figura 25 - Consulta reserva**



```
Código de usuário :
19822|
Consultar alpha
```

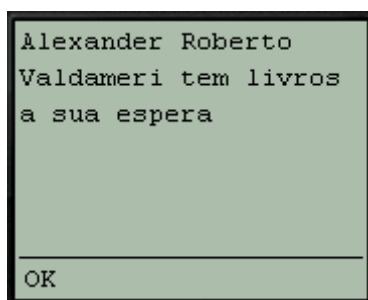
Na tela de consulta de reserva (figura 25) o usuário deve entrar com seu código de usuário na biblioteca e clicar em consultar o sistema irá verificar a disponibilidade do material.

**Figura 26 - Espera negativa**

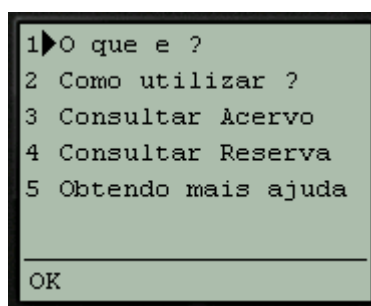


```
Alberto Pereira de
Jesus não existem
livros a sua espera
OK
```

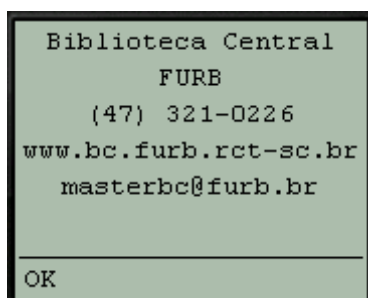
Na tela de resposta (figura 26), é apresentada a resposta negativa dizendo que o material ainda não está disponível.

**Figura 27 - Espera positiva**

Na tela de resposta (figura 27), é apresentada a resposta positiva dizendo que o material está disponível a espera do usuário na seção de empréstimo.

**Figura 28 - Ajuda**

A tela de ajuda (figura 28) descreve o que é o Sistema WAP da Biblioteca Central, apresenta dicas de como utilizar. O usuário deve selecionar a opção desejada e clicar em ok.

**Figura 29 - Contatos**

A tela de contatos (figura 29) apresenta as formas de entrar em contato com a Biblioteca Central da FURB.

## 5 CONCLUSÕES E SUGESTÕES

Este capítulo apresenta as conclusões, limitações e sugestões referentes ao trabalho desenvolvido.

### 5.1 CONCLUSÕES

Pode-se considerar de grande valia o estudo sobre a tecnologia WAP e o conhecimento sobre celulares e redes sem fio. Acredita-se, segundo a literatura e alguns autores, que a tecnologia WAP transformará o cotidiano das pessoas, o modo como elas vivem e trabalham. Criou-se um novo espaço, onde os usuários recebem notícias, mensagens, consultam informações sobre seu saldo no banco, consultam se existem produtos em estoque, verificam se a reserva de seu livro chegou na biblioteca, em qualquer lugar do planeta, diminuindo a distância entre emissores e receptores.

A biblioteca também se inclui nestas transformações, pois trabalha diretamente com a disseminação da informação.

A criação de soluções WAP exige um pouco da nova tecnologia, imaginação, habilidade e domínio das tecnologias já consolidadas. Pois, os dados são os mesmos, só que apresentados em um novo formato.

Os objetivos do trabalho foram alcançados. O protótipo proposto foi realizado e está pronto para entrar em funcionamento. Todas as funções são compatíveis com a nova interface de consulta da Internet fixa da Biblioteca Central.

Quanto à tecnologia envolvida, acredita-se que está apenas nascendo e passará a fazer parte do nosso dia-a-dia. O mercado está em ampla expansão e com possibilidades de grandes negócios, prevendo-se que em 2003 o número de acessos a Internet pelos celulares será maior do que pelos computadores.

Em relação às ferramentas utilizadas, o PowerDesign, mostrou-se eficiente no suporte a metodologia estruturada. O EasyPad Waptor facilitou a edição de códigos WML pois apresenta várias *tags* WML visuais. O UP Simulador apresentou facilidade para os testes das páginas WML apresentando recursos para depuração. Com relação ao servidor IIS do Windows 2000, o mesmo apresentou estabilidade na execução de *scripts*, tornando sua utilização eficiente para esse tipo de aplicação.

Quanto a linguagem ASP, mostrou-se eficiente para criação de páginas acessando banco de dados. Enquanto a linguagem WML mostrou-se funcional e muito similar ao HTML.

A informação hoje é o maior valor econômico. O fato de ser acessível e disponível a qualquer momento torna a tecnologia WAP verdadeiramente revolucionária.

## 5.2 LIMITAÇÕES

Como limitações têm-se os seguintes itens:

- a) incompatibilidade de dados. Todo o acervo encontra-se em arquivos do CDS/ISIS que foram importados para o *Oracle*. Desta forma existem registros que apresentam problemas de formatação em seus parágrafos, podendo apresentar erros quando recuperados pelo protótipo.
- b) incompatibilidade da linguagem ASP com *Oracle*. O objeto *RecordSet* apresenta problemas com a função *RecordCount*, *PageSize* e *PageCount*. Assim, a paginação dos resultados não foi realizada no protótipo. Se fizer uma consulta que recupere muitos registros vai ultrapassar o tamanho máximo do arquivo que é 1,400 bytes, gerando erro para o usuário. Para resolver este problema deve-se implementar uma função de paginação dos resultados.

## 5.3 SUGESTÕES

Como sugestões para trabalho futuros, temos as seguintes possibilidades:

- a) estudar e utilizar a linguagem *WMLScript* para validação de dados do usuário;
- b) estudar aspectos de segurança na tecnologia WAP;
- c) incorporar mais funcionalidades ao protótipo, como validação de senha, verificação da situação do aluno, serviços de e-mail.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [ACC1998] ACCETTA, Izildinha Ramos. **Serviço de Levantamento e Comutação Bibliográfica da Biblioteca Central da FURB: A qualidade da comunicação na recuperação da informação**. Blumenau, 1998. Monografia Curso de Pós-Graduação em nível de especialização em “Qualidade na Comunicação” promovido pela Universidade Regional de Blumenau.
- [CUN1999] CUNHA, Murilo Bastos da. Desafios na construção de um biblioteca digital. **Ciência da Informação**, Brasília, v. 28, n. 3,p. 257-268, set. 1999.
- [DIA2000] DIAS, Adilson de Souza. **WAP - wireless application protocol : A Internet sem fios**. Rio de Janeiro: Ciência Moderna, 2000.
- [FOR2000] FORUM, Wireless Application Protocol. **Official wireless application protocol: the complete standard with searchable cd-rom**. New York: John Wiley & Sons, 1999.
- [GOW2000] GOWap. **O portal WAP do seu celular** 2000. Endereço Eletrônico: <http://www.gowap.com.br>. Data da consulta: 13/08/2000.
- [HOM2000] AREHART, Charles, CHIDAMBARAM, Nirmal, GURUPRASAD, Shashirikan, Et . Al. **Professional WAP ( wireless application protocol )**. Birmingham: Wrox Press, 2000.
- [MOB2000] Mobilidade digital. **Empreendedor**, São Paulo, v. 6, n. 72,p. 36-38, out. 2000.
- [RAD2000] RADUENZ, Gilmar Vereano. **Protótipo de aplicativo para escritório de advocacia com acesso a Internet baseado em ASP**. Blumenau, 2000. Monografia (Bacharelado em Ciências da Computação) Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau.
- [TEI1997] TEIXEIRA, Cenidalva Miranda de Souza, SCHIEL, Ulrich. A Internet e seu impacto nos processos de recuperação da informação. **Ciência da Informação**, Brasília, v. 26, n. 1,p. 65-71, jan./abr. 1997.

- [TEL2000] Telefônica Celular. **Saiba mais sobre WAP** 2000. Endereço Eletrônico: <http://www.telefonicacelular.com.br/wap>. Data da consulta: 14/08/2000.
- [TUD2000] Tudo, a toda hora em qualquer lugar. **Info Exame**, São Paulo, v. 15, n. 173, p. 82-85, ago. 2000.
- [WAP2000a] WAPBrasil. **Artigos, faq e tutoriais** 2000. Endereço Eletrônico: <http://www.wapbr.org>. Data da consulta: 13/08/2000.
- [WAP2000b] **WAP Forum** 2000. Endereço Eletrônico: <http://www.wapFórum.com>. Data da consulta: 10/08/2000.



# ANEXO I

## Fontes Sistema

### index.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="Inicial" title="Biblioteca Central" ontimer="#Menu">
<timer value="50" />
<p align="center">
    <small>Biblioteca Central FURB a sua Biblioteca agora em
WAP</small>
    <br/>
    
</p>
<do type="accept" label="Menu">
    <go href="#Menu"/>
</do>
<do label="Ajuda" type="options">
    <go href="ajuda.wml"/>
</do>
</card>

<card id="Menu" title="Menu">
<p align="center">
Selecione sua Opção :<br/></p>
<p mode="nowrap">
<select multiple="false" name="t">
    <option onpick="pesquisa.wml">Consultar Acervo</option>
    <option onpick="espera.wml">Consultar Reserva</option>
    <option onpick="ajuda.wml">Ajuda</option>
    <option onpick="contato.wml">Contatos</option>
</select>
</p>
<do label="Ajuda" type="options">
    <go href="ajuda.wml"/>
</do>
</card>
</wml>

```

**pesquisa.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="Pesquisa" title="Consultar Acervo">
<p align="center">
    <select name="TpBusca">
        <option value="0">Palavras-Chave</option>
        <option value="1">Autor</option>
        <option value="2">Assunto</option>
        <option value="3">T&#237;tulo</option>
    </select>
Palavras buscar:
<input name="ExpBusca" type="text" maxlength="100" />
    <do type="accept" label="Procurar">
        <go href="Conora.asp" method="post">
            <postfield name="TpBusca" value="$TpBusca" />
            <postfield name="ExpBusca" value="$ExpBusca" />
        </go>
    </do>
</p>
</card>
</wml>
```

**conora.asp**

```

<%@ Language=VBScript %>
<% Option Explicit %>
<!--#include file="classmfn.asp"-->
<!--#INCLUDE FILE="lib.inc" -->
<% 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="Busca" title="Busca Acervo">
<p>
<%
Dim conn
Dim rs
Dim rsdados
Dim str
Dim objMFN
'faz a conexão nativa com o Oracle
set conn=server.createobject("adodb.connection")
conn.open "Provider=MSDAORA.1;Password=dba0604;User ID=dsv;Data
Source=P800;Persist Security Info=True"
'cria a tabela dicionario de dados executando busca na tabela
acervo_dicionario_MFN
set rs=server.createobject("adodb.recordset")
set
rs=conn.execute(Monta_ExpBusca(Request("expbusca"),Request("tpbusca")
))
'verifica se foram encontrados registros
if rs.eof then
response.write("Nenhum t&#237;tulo encontrado")
else
Set objMFN = New MFN
'cria a tabela de dados_mfn
set rsdados=server.createobject("adodb.recordset")
'enquanto existirem registros
while not rs.eof
'executa consulta na tabela acervo_dados_MFN
set rsdados=conn.execute("select * from acervo_dados_mfn where
CD_MFN=" & rs("CD_MFN"))
'carrega dados do registro para a classe MFN
objMFN.Carrega_MFN_Codigo(rsdados("CD_MFN"))
objMFN.Carrega_MFN_Diretorio(rsdados("DS_DIRETORIO"))
objMFN.Carrega_MFN_Informacao(rsdados("DS_INFORMACAO"))
'imprime os títulos chamando a função recupera_MFN criando link
com o código MFN
%><a
href="RecuperaMFN.asp?CodMFN=<%=objMFN.Recupera_CDMFN() %>"><%=objMFN.
Recupera_Titulo%></a>
<br/><%
rs.movenext
wend
'fecha tabela
rsdados.close
set rs=nothing
End If
'fecha conexão com o banco
conn.close
set conn=nothing
%>
</p>
</card>
</wml>

```

**recuperamfn.asp**

```

<%@ Language=VBScript %>
<% Option Explicit %>
<!--#include file="classmfn.asp"-->
<% 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="recupera" title="Recupera Dados MFN">
<p>
<%
Dim conn
Dim rsdados
Dim objMFN
'faz a conexão nativa com o Oracle
set conn=server.createobject("adodb.connection")
conn.open "Provider=MSDAORA.1;Password=dba0604;User ID=dsv;Data
Source=P800;Persist Security Info=True"
'cria a tabela de dados_mfn
set rsdados=server.createobject("adodb.recordset")
'cria o objeto mfn
Set objMFN = New MFN
'executa consulta na tabela acervo_dados_mfn
set rsdados=conn.execute("select * from acervo_dados_mfn where
CD_MFN=" & Request("CodMFN"))
'carrega dados para o objeto
objMFN.Carrega_MFN_Codigo(rsdados("CD_MFN"))
objMFN.Carrega_MFN_Diretorio(rsdados("DS_DIRETORIO"))
objMFN.Carrega_MFN_Informacao(rsdados("DS_INFORMACAO"))
response.write "<br/>"
'escreve dados do mfn na tela e devolve o numero de registro
disponiveis
'se não existem registros disponiveis habilita reserva
If objMFN.Recupera_MFN() = 0 then
%>
<do label="Reservar" type="options">
  <go href="reservar.asp?CodMFN=<%=objMFN.Recupera_CDMFN() %>" />
</do>
<%
End If
'fecha tabela
rsdados.close
set rsdados=nothing
'fecha conexão com o banco
conn.close
set conn=nothing
%>
</p>
</card>
</wml>

```

**reservar.asp**

```

<% 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="Reservar" title="Reservar">
<p><br/>C&#243;digo de usu&#225;rio :
<input name="CdPessoa" type="text" maxlength="30" />
<do type="accept" label="Reservar">
    <go href="insere_reserva.asp?CodMFN=<%=Request("CodMFN")%>"
method="post">
        <postfield name="CdPessoa" value="$CdPessoa" />
    </go>
</do>
</p></card>
</wml>

```

**insere\_reserva.asp**

```

<%@ Language=VBScript %>
<% Option Explicit %>
<!--#include file="classmfnclass.asp"-->
<!--#INCLUDE FILE="lib.inc" -->
<% 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="Insere" title="Insere Reserva">
<p>
<%
    Dim conn
    Dim rsdados
    Dim CD_MFN
    Dim CD_PESSOA
    Dim NM_PESSOA
'faz a conexo nativa com o Oracle
set conn=server.createobject("adodb.connection")
conn.open "Provider=MSDAORA.1;Password=dba0604;User ID=dsv;Data
Source=P800;Persist Security Info=True"
    'carrega parametros
    CD_MFN = Clng(Request("CodMFN"))
    CD_PESSOA = Clng(Request("CDPESSOA"))
    'verifica o codigo do usuario
    NM_PESSOA = Verifica_CodigoPessoa(CD_PESSOA)
    If NM_PESSOA <> "NULO" Then
        response.write(NM_PESSOA)
        response.write(" sua reserva foi efetuada. Aguarde
disponibilidade do material.")
        'cria circulacao_reserva e insere registro de reserva
        set rsdados=server.createobject("adodb.recordset")
        set rsdados=conn.execute("insert into
circulacao_reserva(CD_MFN,NR_VOLUME,CD_PESSOA,CD_UNIDADE_ADMINISTRATI
VA,DT_RESERVA_EFETUADA,CD_OPERADOR,CD_BIBLIOTECA_DEPOSITARIA) values
(" & CD_MFN & "," ' ', " & CD_PESSOA & ",999,SYSDATE,10,'BC')")
    Else
        response.write("C&#243;digo inv&#225;lido")
    End If
conn.close
set conn=nothing
%>
</p></card>
</wml>

```

**espera.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="Espera" title="Verifica Espera">
<p><br/>
C&#243;digo de usu&#225;rio :<input name="CdPessoa" type="text"
maxlength="30" />
<do type="accept" label="Consultar">
  <go href="verifica_espera.asp" method="post">
    <postfield name="CdPessoa" value="$CdPessoa" />
  </go>
</do>
</p></card>
</wml>

```

**verifica\_espera.asp**

```

<%@ Language=VBScript %>
<% Option Explicit %>
<!--#include file="classmfnclass.asp"-->
<!--#INCLUDE FILE="lib.inc" -->
<% 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="VerEspera" title="Verifica Espera">
<p>
<%
Dim conn
Dim rsdados
Dim CD_PESSOA
Dim NM_PESSOA
'faz a conexo nativa com o Oracle
set conn=server.createobject("adodb.connection")
conn.open "Provider=MSDAORA.1;Password=dba0604;User ID=dsv;Data
Source=P800;Persist Security Info=True"
CD_PESSOA = Cng(Request("CDPESSOA"))
'verifica se o codigo do usuario e valido
NM_PESSOA = Verifica_CodigoPessoa(CD_PESSOA)
If NM_PESSOA <> "NULO" Then
  'cria a tabela de circulacao_espera
  set rsdados=server.createobject("adodb.recordset")
  'verifica se existem registro a espera
  set rsdados=conn.execute("SELECT * FROM CIRCULACAO_ESPERA WHERE
CD_PESSOA=" & CD_PESSOA)
  'se nao tiver registro a espera
  If rsdados.eof Then
    response.write(NM_PESSOA & " no existem livros a sua
espera")
  else
    response.write(NM_PESSOA & " tem livros a sua espera")
  End if
Else
  response.write("Codigo invalido")
End If
'fecha a conexo banco
conn.close
set conn=nothing
%>
</p></card>
</wml>

```

**ajuda.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="Ajuda" title="Ajuda">
<p>
<select title="categorias" multiple="false" name="t">
  <option onpick="#what">O que e ?</option>
  <option onpick="#how">Como utilizar ?</option>
  <option onpick="#find_categ">Consultar Acervo</option>
  <option onpick="#find_word">Consultar Reserva</option>
  <option onpick="#more_help">Obtendo mais ajuda</option>
</select>
</p>
</card>

<card id="what" title="O que e?">
<p>
  O Sistema WAP da Biblioteca Central disponibiliza os servicos da
  Biblioteca Central da FURB aos usuarios atraves do seu celular podem
  consultar se existem um determinado livro no acervo, ou consultar se
  chegou a reserva que efetuou.
</p>
</card>

<card id="how" title="Como utilizar?">
<p>
  Voce pode consultar o acervo atraves das categorias ou de
  palavras-chave e consultar se chegou a sua reserva.
</p>
</card>

<card id="find_categ" title="Consultando Acervo">
<p>
  Selecione [Consultar Acervo] no menu principal, selecione a
  categoria da busca e entre com a palavra-chave que descreva o
  titulo. Uma
  lista com a relacao dos titulos aparecera no seu visor. Selecione
  o titulo desejado e click em [Link] aparecera uma tela com os
  detalhes sobre o livro.
</p>
</card>

<card id="find_word" title="Consultando Reservas">
<p>
  Selecione [Consultar Reserva] no menu principal, digite o seu
  codigo de usuario, aparecera uma tela de resposta contendo a
  informacao se o livro esta em espera ao usuario ou nao.
</p>
</card>

<card id="more_help" title="Mais Informações">
<p>
  Acesse http://www.bc.furb.rct-sc.br ou escreva para
  albertop@furb.br para maiores informacoes.
</p>
</card>
</wml>

```

**contatos.wml**

```
<?xml version="1.0"?>
<!-- created by WAPtor (http://www.wapdrive.net/) -->
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="Contatos" title="Contatos">
<p align="center">
  Biblioteca Central FURB<br/>
  (47) 321-0226<br/>
  www.bc.furb.rct-sc.br<br/>
  masterbc@furb.br
</p></card>
</wml>
```



**lib.inc**

```

<%
'esta procedure conecta o banco de dados oracle
Sub Conecta_Banco()
    'variavel de conexão com o banco
    Dim conn
    'faz a conexão nativa com o Oracle
    set conn=server.createobject("adodb.connection")
    conn.open "Provider=MSDAORA.1;Password=dba0604;User ID=dsv;Data
    Source=P800;Persist Security Info=True"
End Sub

'esta funcao recebe como parametro um frase de busca e devolve
'a expressao formtada
Function Monta_Expbusca(sFrase,iTipoBusca)
    'variável auxiliar utilizada para quebrar frase em palavra
    Dim sPalavra
    'variável onde vai ser montada a expressão de busca
    Dim sExpBusca
    'posição inicial onde vai ser quebrada a frase
    Dim iPosIni
    Dim iCont

    'Quebra frase de busca em palavras inicializa iPosIni com 1 para
    copiar a partir primeiro caracter
    iPosIni = 1
    'enquanto não for o final da frase
    For iCont = 1 to Len(sFrase)
        'verifica se é espaço quebra a frase
        If Mid(sFrase,iCont,1) = " " Then
            'pega palavra
            sPalavra = UCase(Mid(sFrase,iPosIni,(iCont-iPosIni)))
            'monta expressão de busca
            '-->verificar se não é palavra reservada
            If Verifica_Reservada(sPalavra) Then
                sExpBusca = Monta_SQL(sExpBusca,sPalavra,iTipoBusca)
            End If
            'seta iPosIni a partir da proxima letra da frase
            iPosIni = iCont + 1
        End If
    Next
    'pega ultima palavra ou a única
    sPalavra = UCase(Mid(sFrase,iPosIni,(iCont-iPosIni)))
    '-->verificar se não é palavra reservada
    'monta expressão de busca
    If Verifica_Reservada(sPalavra) Then
        sExpBusca = Monta_SQL(sExpBusca,sPalavra,iTipoBusca)
    End If
    'devolve resultado a função
    monta_expbusca = (sExpBusca)
End Function

'esta funcao verifica se uma palavra é reservada ou não
'recebe uma palavra como parametro e devolve false se for reservada
Function Verifica_Reservada(sPalavra)
    Dim rsPalReservada
    'cria a tabela palavra reservada e efetua uma consulta da palavra
    set rsPalReservada=server.createobject("adodb.recordset")
    set rsPalReservada=conn.execute("SELECT * FROM PALAVRA_RESERVADA
    WHERE DS_PALAVRA='" & sPalavra & "'")
    'se existir registro é palavra reservada
    If not rsPalReservada.EOF Then

```

```

        Verifica_Reservada = False
    Else
        Verifica_Reservada = True
    End If
    rsPalReservada.close
    set rsPalReservada=nothing
End Function

'esta funcao verifica se o tipo de busca que deve ser feito
'recebe expbusca como parametro e devolve um SQL a ser executado
Function Monta_SQL(sExpBusca,sPalavra,iTipoBusca)
    Dim sSQL
    sSQL = (sExpBusca)
    If sSQL <> "" then
        sSQL= sSQL & " INTERSECT " & "SELECT DISTINCT CD_MFN FROM
ACERVO_DICIONARIO_MFN WHERE DS_TERMOS= '" & sPalavra & "'"
    Else
        sSQL= sSQL & "SELECT DISTINCT CD_MFN FROM
ACERVO_DICIONARIO_MFN WHERE DS_TERMOS= '" & sPalavra & "'"
    End if
    Select Case iTipoBusca
        'Busca por autor
        Case 1
            sSQL = sSQL & " AND (NR_CAMPO='100' OR NR_CAMPO='110' OR
NR_CAMPO='111' OR NR_CAMPO='130' OR NR_CAMPO='700' OR NR_CAMPO='710'
OR NR_CAMPO='711' OR NR_CAMPO='730')"
            'Busca por assunto
            Case 2
                sSQL = sSQL & " AND (NR_CAMPO='600' OR NR_CAMPO='610' OR
NR_CAMPO='611' OR NR_CAMPO='630' OR NR_CAMPO='650' OR
NR_CAMPO='651')"
            'Busca por título
            Case 3
                sSQL = sSQL & " AND (NR_CAMPO='210' OR NR_CAMPO='214' OR
NR_CAMPO='222' OR NR_CAMPO='242' OR NR_CAMPO='243' OR NR_CAMPO='245'
OR NR_CAMPO='246' OR NR_CAMPO='740')"
            End Select
        Monta_SQL = sSQL
    End Function

```

'esta procedure conecta o banco de dados oracle no servidor F50 do NI  
'e verifica se o código do usuário é válido

```

Function Verifica_CodigoPessoa(sCD_PESSOA)
    Dim conF50
    Dim rsPessoa
    'faz a conexão nativa com o Oracle no servidor f50
    set conF50=server.createobject("adodb.connection")
    conF50.open "Provider=MSDAORA.1;Password=dba0604;User
ID=biblioteca;Data Source=F50;Persist Security Info=True"
    'cria a tabela Prd.Pessoa
    set rsPessoa=server.createobject("adodb.recordset")
    set rsPessoa=conF50.execute("SELECT NM_PESSOA FROM PRD.PESSOA
WHERE CD_PESSOA=" & sCD_PESSOA)
    If rsPessoa.eof Then
        Verifica_CodigoPessoa = "NULO"
    Else
        Verifica_CodigoPessoa = rsPessoa("NM_PESSOA")
    End If
End Function
%>

```

**classmf.n.asp**

```

<%
Class MFN
public CD_MFN
public DS_DIRETORIO
public DS_INFORMACAO

public Sub Carrega_MFN_Codigo(CODIGO)
    CD_MFN = CODIGO
End Sub

public Sub Carrega_MFN_Diretorio(DIRETORIO)
    DS_DIRETORIO = DIRETORIO
End Sub

public Sub Carrega_MFN_Informacao(INFORMACAO)
    DS_INFORMACAO = INFORMACAO
End Sub

'recupera codigo do objeto MFN
public Function Recupera_CDMFN()
    Recupera_CDMFN = CD_MFN
End Function

'retira marcadores de paragrafo
public Function Retira_Marcadores(strPalavra)
    strPalavra = Replace(strPalavra,"#",",",1)
    strPalavra = Replace(strPalavra,"^a",",",1)
    strPalavra = Replace(strPalavra,"^b",",",1)
    strPalavra = Replace(strPalavra,"^c",",",1)
    strPalavra = Replace(strPalavra,"^d",",",1)
    strPalavra = Replace(strPalavra,"^A",",",1)
    strPalavra = Replace(strPalavra,"^B",",",1)
    strPalavra = Replace(strPalavra,"^C",",",1)
    strPalavra = Replace(strPalavra,"^D",",",1)
    strPalavra = Replace(strPalavra,"^|",",",1)
    Retira_Marcadores = strPalavra
End Function

'recupera titulo do MFN
public Sub Recupera_Titulo()
    Dim StrPar
    Dim Paragrafo
    Dim TamInfo
    Dim PosIni
    Dim cont_byte
    'inicializa variável no primeiro byte
    cont_byte = 1
    'enquanto não for o final da string DS_DIRETORIO
    while cont_byte <= Len(DS_DIRETORIO)
        'copia 12 bytes que contém o paragrafo e a localização dentro do
        campo DS_INFORMACAO
        StrPar = Mid((DS_DIRETORIO),cont_byte,12)
        'copia o paragrafo
        Paragrafo = (Mid((StrPar),1,3))
        'copia o tamanho da informação
        TamInfo = (Mid((StrPar),4,4))
        'copia a posição inicial da informação
        PosIni = (Mid((StrPar),8,5))
        'se for paragrafo do titulo
        If Paragrafo = 245 Then
            'copia a informação para a variavel
            StrPar = Mid(DS_INFORMACAO,(PosIni+1),(TamInfo))
        End If
        cont_byte = cont_byte + 1
    End While
End Sub

```

```

        If InStr(StrPar,"^c") > 0 Then
            StrPar = Mid(StrPar,1,(InStr(StrPar,"^c")-2))
        End If
        'retira caracteres de marcação de paragrafos
        strPar = Retira_Marcadores(strPar)
        'responde escrevendo o título
        response.write(StrPar)
    End If
    'pula para o próximo paragrafo
    cont_byte = cont_byte + 12
Wend
End Sub

'recupera dados do MFN
public Function Recupera_MFN()
    Dim StrReg
    Dim StrPar
    Dim StrBiblio
    Dim StrTpEmp
    Dim Paragrafo
    Dim TamInfo
    Dim PosIni
    Dim cont_byte
    Dim iCont_Disp
    'inicializa variável no primeiro byte
    cont_byte = 1
    iCont_Disp = 0
    'enquanto não for o final da string DS_DIRETORIO
    while cont_byte <= Len(DS_DIRETORIO)
        'copia 12 bytes que contém o paragrafo e a localização dentro do
        campo DS_INFORMACAO
        StrPar = Mid((DS_DIRETORIO),cont_byte,12)
        'copia o paragrafo
        Paragrafo = (Mid((StrPar),1,3))
        'copia o tamanho da informação
        TamInfo = (Mid((StrPar),4,4))
        'copia a posição inicial da informação
        PosIni = (Mid((StrPar),8,5))

        Select Case Paragrafo
            'visualiza classificação
            Case 90
                StrPar = Mid(DS_INFORMACAO,(PosIni+1),(TamInfo))
                StrPar = Replace(StrPar,"#",",",1)
                StrPar = Replace(StrPar,"^|",",",1)
                StrPar = Replace(StrPar,"^a",",",1)
                StrPar = Replace(StrPar,"^b",",",1)
                StrPar = Replace(StrPar,"^c",",",1)
                StrPar = Replace(StrPar,"^d",",",1)
                StrPar = Replace(StrPar,"^A",",",1)
                StrPar = Replace(StrPar,"^B",",",1)
                StrPar = Replace(StrPar,"^C",",",1)
                StrPar = Replace(StrPar,"^D",",",1)
                response.write "CLA: " & StrPar & "<br/>"
                'response.write "Classificação -->> " & StrPar & "<BR>"
            'visualiza autor
            Case 100
                StrPar = Mid(DS_INFORMACAO,(PosIni+1),(TamInfo))
                StrPar = Retira_Marcadores(StrPar)
                'response.write "Autor -->> " & StrPar & "<BR>"
                response.write "AUT: " & StrPar & "<br/>"
            'visualiza titulo
            Case 245

```

```

        StrPar = Mid(DS_INFORMACAO,(PosIni+1),(TamInfo))
        StrPar = Retira_Marcadores(StrPar)
        'response.write "Titulo -->> " & StrPar
        response.write "TIT: " & StrPar
    'visualiza area de publicacao
    Case 260
        StrPar = Mid(DS_INFORMACAO,(PosIni+1),(TamInfo))
        StrPar = Retira_Marcadores(StrPar)
        response.write " " & StrPar
    'visualiza descrição física
    Case 300
        StrPar = Mid(DS_INFORMACAO,(PosIni+1),(TamInfo))
        StrPar = Retira_Marcadores(StrPar)
        response.write " - " & StrPar & "<br/>"
        response.write "Registro Dep Sit. <br/>"
    'verifica registros do MFN
    Case 949
        StrPar = Mid(DS_INFORMACAO,(PosIni+1),(TamInfo))
        'verifica número de registro, tipo emprestimo e biblioteca
        depositaria
        If InStr(StrPar,"^t") > 0 Then
            'copia número registro
            StrReg = Mid(StrPar,1+3,(InStr(StrPar,"^t")-4))
            'copia tipo emprestimo
            StrTpEmp = Mid(StrPar,InStr(StrPar,"^t")+2,2)
            'copia biblioteca depositaria
            StrBiblio =
            Mid(StrPar,InStr(StrPar,"^b")+2,(InStr(StrPar,"^e")-
            InStr(StrPar,"^b"))-2)
            End If
            StrPar = Replace(StrPar,"#",",",1)
            StrPar = Replace(StrPar,"^a",",",1)
            response.write StrReg & " " & StrBiblio & " "
            'escreve a situação do exemplar
            If StrTpEmp = "CR" Then
                response.write("Col Res")
            Else
                If objMFN.Verifica_Disponibilidade(StrReg) Then
                    iCont_Dis = iCont_Dis + 1
                End If
            End If
            response.write "<br/>"
        End Select
        cont_byte = cont_byte + 12
    Wend
    Recupera_MFN = iCont_Dis
End Function

public Function Verifica_Disponibilidade(NrReg)
Dim rscirc_emp
Dim rscirc_esp
    'cria a tabela circulacao_emprestimo
    set rscirc_emp = server.createobject("adodb.recordset")
    set rscirc_emp = conn.execute("select * from circulacao_emprestimo
    where nr_registro='" & NrReg & "'")
    'cria a tabela circulacao_espera
    set rscirc_esp = server.createobject("adodb.recordset")
    set rscirc_esp = conn.execute("select * from circulacao_espera
    where nr_registro='" & NrReg & "'")

    If not rscirc_emp.bof Then
        response.write("Empre")
        Verifica_Disponibilidade = False
    End If
End Function

```

```
Else
  If not rscirc_esp.bof Then
    response.write("Espera")
    Verifica_Disponibilidade = False
  Else
    response.write("Disp")
    Verifica_Disponibilidade = True
  End If
End If
End Function
End Class
%>
```

## ANEXO II

Tabela Latin-1 (ISO 8859-1) e seus códigos:

Nome	Código	Resultado
&nbsp;	&#160;	espaço
&iexcl;	&#161;	¡
&cent;	&#162;	¢
&pound;	&#163;	£
&curren;	&#164;	¤
&yen;	&#165;	¥
&brvbar;	&#166;	¦
&sect;	&#167;	§
&uml;	&#168;	¨
&copy;	&#169;	©
&not;	&#172;	¬
&reg;	&#174;	®
&deg;	&#176;	°
&plusmn;	&#177;	±
&acute;	&#180;	´
&frac14;	&#188;	¼
&frac12;	&#189;	½
&frac34;	&#190;	¾
iquest;	&#191;	¿
Agrave;	&#192;	À
Aacute;	&#193;	Á
Acirc;	&#194;	Â
Atilde;	&#195;	Ã
Auml;	&#196;	Ä
Aring;	&#197;	Å
AElig;	&#198;	Æ
Ccedil;	&#199;	Ç
Egrave;	&#200;	È
Eacute;	&#201;	É
Ecirc;	&#202;	Ê
Euml;	&#203;	Ë
Igrave;	&#204;	Ì
Iacute;	&#205;	Í
Icirc;	&#206;	Î
Iuml;	&#207;	Ï
Ntilde;	&#209;	Ñ
Ograve;	&#210;	Ò
Oacute;	&#211;	Ó
Ocirc;	&#212;	Ô
Otilde;	&#213;	Õ
Ouml;	&#214;	Ö
times;	&#215;	×
Oslash;	&#216;	Ø
Ugrave;	&#217;	Ù
Uacute;	&#218;	Ú
Ucirc;	&#219;	Û
Uuml;	&#220;	Ü
Yacute;	&#221;	Ý
THORN;	&#222;	Þ
szlig;	&#223;	ß
agrave;	&#224;	à
aacute;	&#225;	á

acirc;	&#226;	â
atilde;	&#227;	ã
auml;	&#228;	ä
aring;	&#229;	å
aelig;	&#230;	æ
ccedil;	&#231;	ç
egrave;	&#232;	è
eacute;	&#233;	é
ecirc;	&#234;	ê
euml;	&#235;	ë
igrave;	&#236;	ì
iacute;	&#237;	í
icirc;	&#238;	î
iuml;	&#239;	ï
ntilde;	&#241;	ñ
ograve;	&#242;	ò
oacute;	&#243;	ó
ocirc;	&#244;	ô
otilde;	&#245;	õ
ouml;	&#246;	ö
divide;	&#247;	÷
oslash;	&#248;	ø
ugrave;	&#249;	ù
uacute;	&#250;	ú
ucirc;	&#251;	û
uuml;	&#252;	ü
yacute;	&#253;	ý
thorn;	&#254;	þ
yuml;	&#255;	ÿ



