

ANDRÉA LILIANA MARUGEIRO FORTUNATO

**CONVERSÃO AUTOMÁTICA ENTRE AS LINGUAGENS DE
MARCAÇÃO DE TEXTO HTML E WML**

Trabalho de conclusão de curso apresentado ao Curso de Ciência da Computação.

UNIVERSIDADE PRESIDENTE ANTÔNIO CARLOS

Orientador: Prof. M. Sc. Emerson Rodrigo Alves Tavares

BARBACENA

2003

ANDRÉA LILIANA MARUGEIRO FORTUNATO

**CONVERSÃO AUTOMÁTICA ENTRE AS LINGUAGENS DE
MARCAÇÃO DE TEXTO HTML E WML**

Este trabalho de conclusão de curso foi julgado adequado à obtenção do grau de Bacharelado em Ciência da Computação e aprovado em sua forma final pelo Curso de Ciência da Computação da Universidade Presidente Antônio Carlos.

Barbacena – MG, 4 de dezembro de 2003.

Prof. M.Sc. Emerson Rodrigo Alves Tavares - Orientador do Trabalho

Prof. Luís Augusto Mattos Mendes - Membro da Banca Examinadora

Prof^a. Lorena Sophia Campos de Oliveira - Membro da Banca
Examinadora

“A razão cardeal de toda superioridade humana é sem dúvida a vontade. O poder nasce do querer. Sempre que o homem aplique a veemência e a perseverante energia de sua alma a um fim, ela vencerá os obstáculos, e se não atingir o alvo, fará pelo menos coisas admiráveis.”

José Martiniano de Alencar

AGRADECIMENTOS

Agradeço a Deus, cuja vontade permitiu que concluísse esse trabalho. Ao professor, Emerson Rodrigo Alves Tavares, orientador, que transmitiu os conhecimentos iniciais à realização desse trabalho e todo suporte. À minha família que vem incentivando todas as minhas realizações e por fim, aos demais professores da UNIPAC que mostraram-se capazes nas suas atribuições.

RESUMO

Este trabalho tem como objetivo descrever um modelo estático de transformação entre as linguagens de marcação de texto HTML – WML. A conversão é orientada pela sintaxe entre as linguagens consideradas. O modelo desenvolvido foi baseado na conversão léxica das estruturas da linguagem de origem (HTML) respeitando os limites da linguagem de destino (WML). Os templates foram desenvolvidos observando a sintaxe abstrata comum entre as linguagens em questão. O reconhecimento léxico foi feito a partir da leitura dos tokens e seu tratamento por uma tabela de equivalência. Como resultado, foi implementado um conversor simplificado de páginas da Web para páginas que permitem acesso através de dispositivos móveis.

Palavras-chave: Tradução automática, HTML, WML.

SUMÁRIO

<u>LISTA DE FIGURAS.....</u>	<u>9</u>
<u>LISTA DE TABELAS.....</u>	<u>10</u>
<u>1 INTRODUÇÃO.....</u>	<u>11</u>
<u>2 REVISÃO BIBLIOGRÁFICA.....</u>	<u>13</u>
<u>3 MODELO DE TRADUÇÃO DE LINGUAGENS HTML - WML.....</u>	<u>34</u>
<u>4 CONCLUSÃO.....</u>	<u>45</u>
<u>REFERÊNCIAS BIBLIOGRÁFICAS.....</u>	<u>47</u>
<u>ANEXO A – DTD DA LINGUAGEM WML.....</u>	<u>48</u>

LISTA DE FIGURAS

<u>FIGURA 1 – MODELO DE REFERÊNCIA DE REDE WAP [FORT, 2002].....</u>	<u>15</u>
<u>FIGURA 2 – OS DISPOSITIVOS WAP SOLICITAM E RECEBEM DADOS DOS SERVIDORES WAP. [FORT, 2003].....</u>	<u>17</u>
<u>FIGURA 4 – ESTRUTURA BÁSICA DE UM PROGRAMA HTML.....</u>	<u>21</u>
<u>FIGURA 5 - NOKIA 7110 [NOKIA, 2003].....</u>	<u>36</u>
<u>FIGURA 6 – MODELO DE TRADUÇÃO.....</u>	<u>36</u>
<u>FIGURA 7 – TELA DO PROGRAMA DE CONVERSÃO.....</u>	<u>39</u>
<u>FIGURA 8 – TEXTO HTML EXIBIDO.....</u>	<u>40</u>
<u>FIGURA 9 – EXIBIÇÃO DO CÓDIGO WML OBTIDO COM A CONVERSÃO.....</u>	<u>40</u>
<u>FIGURA 10 – SALVANDO DOCUMENTO WML.....</u>	<u>41</u>
<u>FIGURA 11 – EXEMPLOS DE TRADUÇÃO.....</u>	<u>42</u>
<u>FIGURA A1- DIAGRAMA DO DTD DA LINGUAGEM WML 1.0.....</u>	<u>53</u>

LISTA DE TABELAS

<u>TABELA 1: PILHA DE PROTOCOLOS WAP [FORT ,2003].....</u>	<u>15</u>
<u>TABELA 2 – TAGS DE PARÁGRAFOS E QUEBRAS DE LINHA [RAMA, 2002].....</u>	<u>22</u>
<u>TABELA 3 - TAGS DE ESTILO DE TEXTO[RAMA, 2002].....</u>	<u>23</u>
<u>TABELA 4 - PADRÕES DE FUNDO [RAMA, 2002].....</u>	<u>24</u>
<u>TABELA 5 - PARÂMETROS DA TAG IMG [RAMA, 2002].....</u>	<u>25</u>
<u>TABELA 6 - PRINCIPAIS PARÂMETROS DA TAG<A>[RAMA, 2002].....</u>	<u>25</u>
<u>TABELA 7 – TAGS PARA GERAR LISTAS[RAMA, 2002].....</u>	<u>26</u>
<u>TABELA 8 - TAGS PARA A CONSTRUÇÃO DE TABELAS[RAMA, 2002].....</u>	<u>26</u>
<u>TABELA 9 – PARÂMETROS DA TAG <FORM>[RAMA ,2002].....</u>	<u>27</u>
<u>TABELA 10 – PARÂMETROS DA TAG <INPUT>[RAMA, 2002].....</u>	<u>27</u>
<u>TABELA 11 - PARÂMETROS DA TAG <SELECT>[RAMA, 2002].....</u>	<u>28</u>
<u>TABELA 12 – TAGS DE WML [FORT, 2002].....</u>	<u>33</u>
<u>TABELA 13 – ENTIDADES DE CARACTERES ESPECIAIS DA WML[FORT, 2002].....</u>	<u>33</u>
<u>TABELA 14 – TAGS CORRESPONDENTES.....</u>	<u>37</u>
<u>TABELA 15 – TAGS NÃO SUPORTADAS.....</u>	<u>38</u>

1 INTRODUÇÃO

Nos últimos anos tem se verificado um aumento na demanda por serviços de informações para os usuários que utilizam dispositivos móveis, tais como Telefones Celulares e Palmtops.

A mídia atualmente vem dando um grande destaque à Internet Móvel. Visões fantásticas de fregueses efetuando uma consulta à Internet apenas com o auxílio de seus telefones celulares para escolher o restaurante em que vão jantar é uma realidade bem próxima[MARC 2001].

É verdade que os usuários estão se tornando cada vez mais dependentes dos serviços oferecidos via Internet mas para isso precisam estar conectados à rede através de um fio. É fato também que muitos usuários passam a maior parte do tempo em trânsito ou em locais onde um cabo para se conectar a Internet é um empecilho[MARC 2001]. Para resolver esses problemas surge a tecnologia WAP (*Wireless Application Protocol* – Protocolo para aplicações sem fio) que tem a capacidade de tornar possível o acesso à Internet por meio de dispositivos móveis sem fio.

A linguagem HTML (*HyperText Markup Language* – Linguagem de marcação de hipertexto) é uma linguagem universal destinada à marcação de documentos de hipertexto. Ela é utilizada para definir páginas na Web, baseada em códigos embutidos em um

documento que pode servir para ajustar fontes, criar listas, mostrar imagens, entre outros tipos de formatações de páginas. As formatações de páginas, as imagens coloridas, os *hyperlinks* que possibilitam navegar pelo mundo virtual, foram desenvolvidos utilizando HTML [CHAI 2001]. Um problema é que não é possível visualização de HTML através de dispositivos móveis.

Com WAP, é possível acessar páginas no padrão WML (*Wireless Markup Language*, uma espécie de linguagem HTML para redes móveis). A linguagem WML é utilizada para especificar conteúdo e uma interface de usuário para dispositivos WAP considerando a limitada capacidade dos dispositivos como telefones celulares e outros terminais móveis sem fio.[MESS 2000] [FREI 2001]

Este trabalho tem como objetivo estudar as linguagens HTML e WML e criar uma forma de se traduzir um código em HTML para um código WML.

O corpo desse trabalho está organizado da seguinte forma: o capítulo 2 trata dos conceitos e histórico da tecnologia WAP e das linguagens HTML e WML. O capítulo 3 procura restringir o escopo do trabalho definindo o que vai ser possível de se converter e propõe um modelo simplificado de conversão estática do código HTML para WML. O capítulo 4 apresenta a conclusão deste trabalho.

2 REVISÃO BIBLIOGRÁFICA

2.1 TECNOLOGIA WAP

O WAP (*Wireless Application Protocol*) é um ambiente de aplicação e um grupo de protocolos de comunicação para dispositivos sem fio, criado para permitir acesso à Internet e a um conjunto de serviços de telefonia avançados[FREI, 2001]. Ele pode ser visto como sendo uma ponte que interliga o mundo dos dispositivos móveis e a Internet, assim um usuário que pode ter acesso à informações através de um microcomputador de mesa também pode fazer o mesmo através de um dispositivo de bolso, com a vantagem de que o conteúdo está disponível ao mesmo tempo em que o usuário se movimenta.

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

2.1.1 ORIGEM DO PROTOCOLO

A especificação do WAP teve origem em 1997 quando um grupo de empresas formado pela Ericsson, Nokia, Motorola e Unwired Planet deram origem ao chamado WAP Fórum, que hoje ainda é o órgão responsável pela publicação das especificações de produtos WAP [FORT, 2003]. A proposta inicial do WAP Fórum foi a de definir uma ampla especificação para a indústria a fim de permitir o desenvolvimento de aplicações sobre redes de comunicação sem fio[FORT, 2003].Atualmente, fazem parte do WAP Fórum quase todas as principais empresas de software, hardware e dispositivos, bem como as principais operadoras e os principais provedores de telecomunicações.

O WAP foi desenvolvido exclusivamente para a computação sem fio e foi criado para dar conta das limitações únicas e fundamentais desse tipo de computação[FORT, 2003]:

- Dispositivos com memória e potência de processamento limitadas
- Potência e vida útil da bateria limitadas
- Telas pequenas
- Recursos limitados de entrada de dados e interação com o usuário
- Velocidades de conexão e largura de banda limitadas
- Conexões instáveis

O WAP não sendo um protocolo individual, mas sim um conjunto de protocolos, como mostrado na tabela 1, e padrões que formam uma pilha completa de protocolos e linguagens especiais de marcação como a WML e script que juntos proporcionam uma solução completa para a transmissão sem fio.

WAP Applications

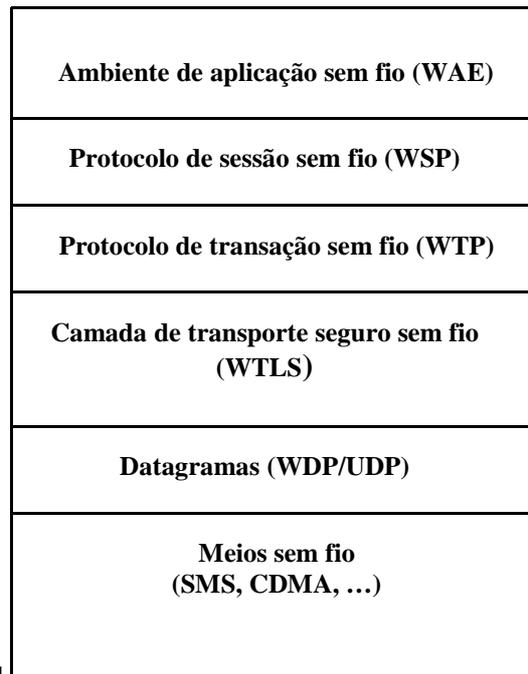


Figura 1 – Modelo de referência de rede WAP [FORT, 2002]

Abreviação	Descrição do nome
WAE	Wireless Application Environment Camada Aplicativo. Inclui o micronavegador no dispositivo, WML (o Wireless Markup Language), WMLScript (uma linguagem de script de cliente), serviços de telefonia e um conjunto de formatos para os dados que costumam ser usados (como imagens, cadernos telefônicos e agendas).
WSP	Wireless Session Protocol Camada Sessão. Fornece funcionalidade HTTP1.1, com o gerenciamento básico de estado de sessão e um recurso para envio e recebimento de dados confidenciais e não confidenciais.
WTP	Wireless Transaction Protocol Camada Transação. Fornece serviços de transporte (unidirecionais e bidirecionais) e tecnologias relacionadas.
WTLS	Wireless Transport Layer Security Camada Segurança. Fornece privacidade e segurança dos dados, autenticação e proteção contra ataques a serviços não-autorizados.
WDP	Wireless Datagram Protocol Camada Transporte Geral.

Tabela 1: Pilha de protocolos WAP [FORT, 2003]

O conjunto de protocolos WAP é desenvolvido para minimizar o tamanho de largura de banda requerido e maximizar o número de tipos de redes sem fio que podem distribuir o conteúdo WAP[FREI, 2001].

2.1.2 FUNCIONAMENTO DO MODELO WAP

O conteúdo e aplicações WAP são especificadas em um conjunto de formatos padronizados e baseados nos formatos padrões da Web. Os dados são transportados utilizando um conjunto de protocolos de comunicação baseados nos protocolos WWW e o micronavegador de um dispositivo móvel de forma análoga ao navegador Web.

O protocolo WAP define um conjunto de componentes padronizados que permitem a comunicação entre terminais móveis e servidores de rede, incluindo[FREI, 2001]:

- Modelo de nomes padrão: O padrão WWW de URL's foi utilizado para identificar o conteúdo WAP nos servidores de origem. Outro padrão WWW, o URI, foi utilizado para identificar recursos locais num dispositivo, como funções de controle de chamadas.
- Tipagem de conteúdo: Todos os conteúdos WAP, possuem um tipo de conteúdo específico, consistente com o modelo WWW.
- Formato de conteúdo padrão: O formato de conteúdo WAP é baseado na tecnologia WWW e inclui marcações de visualização, informação de agenda, cartões de visita, imagens e linguagens de script.
- Protocolos de comunicação padrão: Os protocolos WAP permitem a transmissão de pedido do browser para o servidor de rede Web.

Como mostrado na figura 1, o dispositivo WAP faz uma solicitação ao servidor WAP, que retorna os dados solicitados para o dispositivo para processamento.

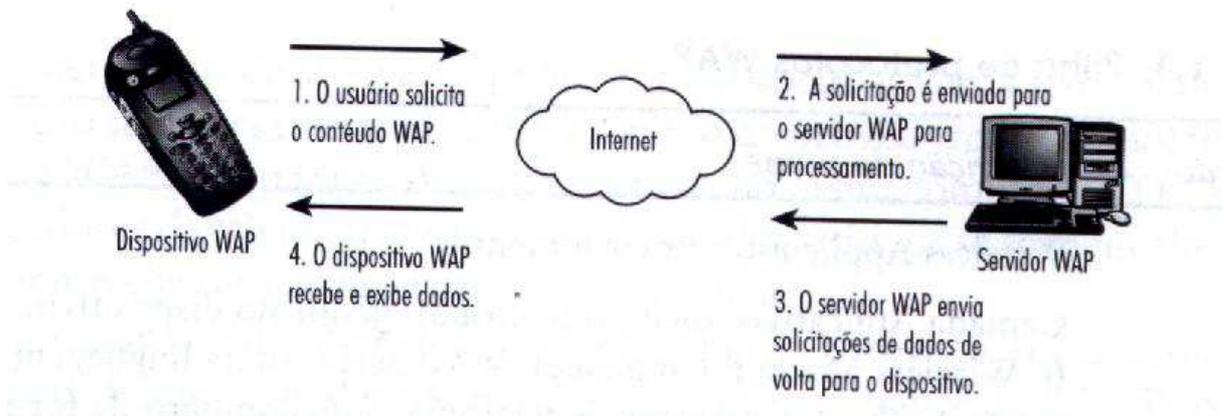


Figura 2 – Os dispositivos WAP solicitam e recebem dados dos servidores WAP.[FORT, 2003]

Para que um dispositivo WAP possa solicitar dados de servidores HTTP é necessário posicionar um gateway WAP entre o dispositivo WAP e o servidor HTTP e funcionar como um interpretador entre eles. Ver figura 3.

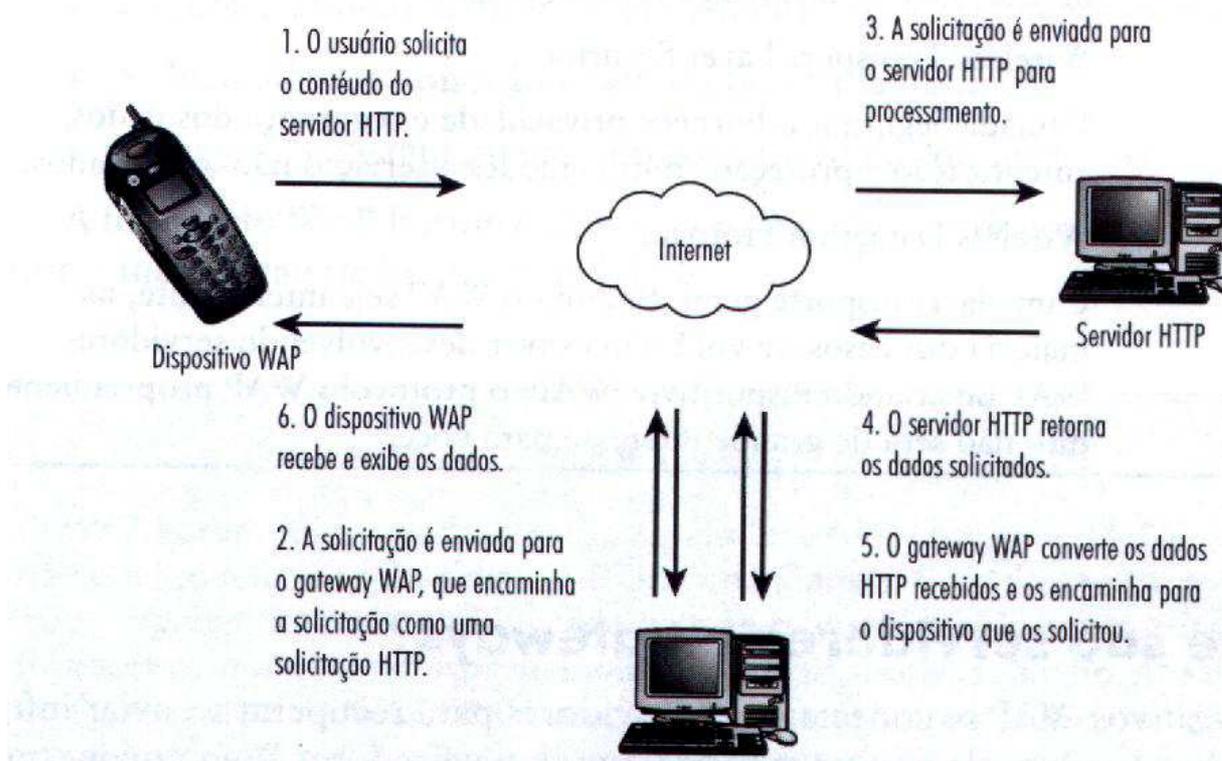


Figura 3 – Os dispositivos WAP podem solicitar e receber dados de servidores HTTP através de gateways WAP[FORT, 2003].

O gateway WAP é um software que realiza a conexão entre o cliente e o servidor (HTTP).

O uso do gateway WAP se justifica porque permite que o conteúdo e as aplicações, criadas para serem acessadas através de um dispositivo móvel, possam ser hospedadas em um servidor WWW comum, beneficiando-se de toda estrutura e tecnologia da Internet existente[FREI, 2001].

O gateway WAP possui as funcionalidades de um Gateway de protocolo, traduzindo as solicitações WAP para os padrões da Internet. Também se comporta como um codificador e decodificador de conteúdo colocando o conteúdo a ser transmitido em um formato de código compactado, *bytecode*, com a intenção de reduzir o tamanho e o número de pacotes que trafegam na rede de dados sem fio.

Da mesma maneira que os servidores HTTP podem ser usados para distribuir o conteúdo WAP, muitas das ferramentas, das linguagens e dos ambientes de desenvolvimento usados para desenvolver aplicativos da Web também podem ser usados para desenvolver aplicativos Wap[FORT, 2002]. Dentre eles, estão Perl, ASP e Java.

2.1.3 WAP E WML

A maior parte do trabalho no desenvolvimento WAP se concentra na linguagem de marcação utilizada para criar os aplicativos WAP. Os dispositivos WAP apresentam requisitos e restrições especiais de interface com o usuário, a utilização da HTML não é uma opção. Por exemplo[FORT, 2003]:

- Os dispositivos WAP não dispõem de dispositivos apontadores. Por isso, as interfaces como mouse são impossíveis.
- Os dispositivos WAP não apresentam teclados completos; eles costumam ter teclados numéricos simples de telefone com algumas teclas adicionais.

- As telas dos dispositivos WAP são pequenas. Elas não conseguem oferecer suporte a quadros, tabelas complexas, gráficos muito grandes e controle sofisticado de cor e fonte.
- Os dispositivos WAP não oferecem suporte real para multimídia, som e vídeo.

Por causa desses requisitos especiais dos dispositivos móveis é que foi criada uma linguagem de marcação especial para o WAP: a WML[FORT, 2003]. A WML é muito parecida com HTML, mas não é tão extensa quanto esta.

2.2 LINGUAGENS DE MARCAÇÃO

No início os computadores eram usados apenas para processar dados, em seguida viu-se a sua grande utilidade para organizar informações e enfim para a comunicação entre pessoas. A intensificação do uso do computador para a manipulação e troca de informação deu origem as linguagens de marcação que foram criadas para tentar minimizar o problema de transferência de um formato de representação para outro[BAX, 2001].

Neste trabalho o estudo é focado nas linguagens de marcação: HTML e WML.

2.2.1 HTML

HTML é uma linguagem originada de SGML (*Standard Generalized Markup Language*). A SGML é uma meta linguagem criada para se definir uma linguagem de marcas para a representação de informações em texto. Ela não é um conjunto pré determinado de marcas, e sim uma linguagem para se definirem quaisquer conjuntos de marcas. Cada documento SGML carrega sua própria especificação formal, o DTD (*Data Type Document*). O DTD é uma espécie de gramática formal que define como as marcas devem ser interpretadas, quais as regras que restringem o uso de cada marca nos diferentes contextos do

documento e até mesmo a ordem em que as marcas devem aparecer no documento[BAX, 2001].

A HTML é uma linguagem muito popular hoje, é um padrão utilizado em milhares de aplicações. A Web explodiu em 1990 e muitas empresas começaram a perceber que estavam ficando limitadas pela falta de flexibilidade de HTML no que dizia respeito à efetiva troca de informações via Web. Foi então que a HTML foi sendo estendida a cada nova versão.

A HTML define como se dá a organização da informação ou das idéias no documento [BAX, 2001]. Ela é limitada, não fazendo mais do que indicar como as informações devem aparecer no navegador, capaz apenas de marcação estrutural e não semântica.

A grande vantagem de HTML é sua facilidade em ser entendida pelo usuário da Web e de ser processada, mesmo em diferentes navegadores. Isso foi o aspecto principal responsável pela explosão da Web.

2.2.1.1 ESTRUTURA BÁSICA DE UM DOCUMENTO HTML

Os comandos da linguagem HTML são chamados de tags que normalmente são especificados em pares, delimitando um texto que sofrerá algum tipo de formatação, mas também podem ser exibidos individualmente[RAMA, 2002]. Por exemplo, uma tag pode dizer que um texto deve ser exibido em negrito, itálico, com um determinado tipo de fonte, etc. As tags são identificadas por serem envolvidos pelos sinais '< >' ou '< />'. Entre os sinais '< >' são especificados os comandos propriamente ditos. O formato genérico de um tag é: <Nome da tag> texto </Nome da tag>.

Como muitas linguagens, a HTML possui uma estrutura básica para que um browser interprete corretamente o programa. Ela deve possuir comandos básicos que sempre deverão estar presentes.

Um programa HTML possui três partes básicas que são vistas na figura abaixo:

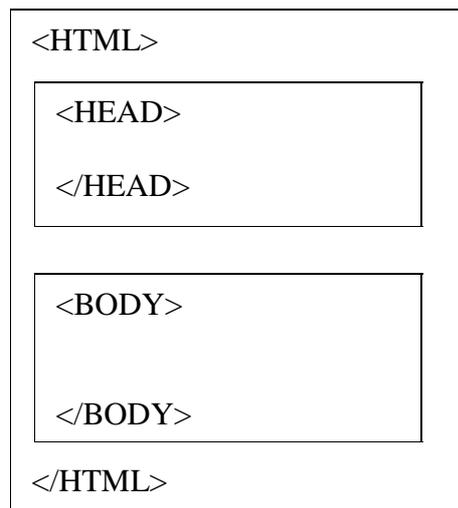


Figura 4 – Estrutura básica de um programa HTML.

Todo programa HTML deve iniciar com a tag `<HTML>` e ser encerrado com a tag `</HTML>`. Esse par de tags é essencial. As tags `<HEAD>` `</HEAD>` delimitam a área de cabeçalho que é opcional. Dentro dessas tags é que colocamos o título da página com as tags `<TITLE>` título da página `</TITLE>`.

A maioria dos comandos HTML será colocada na área do corpo do programa que é delimitada pela tag `<BODY>` `</BODY>`. Essas tags também são obrigatórias[RAMA , 2002].

2.2.1.2 TAGS

2.2.1.2.1 *Parágrafos e quebras de linha*

Ao contrário dos editores de texto tradicionais, nos quais você insere uma quebra ou avanço de linha toda vez que pressiona a tecla Enter, em um programa HTML é necessário

colocar um comando específico para que o browser entenda que você deseja mudar de linha e iniciar uma nova linha ou começar um novo parágrafo[RAMA, 2002].

Tag	Descrição
<P>	Para forçar o início de um novo parágrafo, ou seja, avançar uma linha em branco e iniciar o texto na segunda linha após o final do parágrafo anterior, deve ser usado o comando <P>. Ele pode aparecer individualmente ou em par <P> </P>.
 	A tag tem como função avançar para a linha imediatamente abaixo após aquela em que ele se encontra, sem, contudo, deixar uma linha em branco adicional.
<HR>	A tag <HR> tem a finalidade de inserir uma linha divisória na posição em que foi especificado, se ele for o único comando da linha, ou na linha seguinte, caso esteja no meio de um texto. Seu uso é recomendado para criar uma divisão na página, separando tópicos e assuntos distintos.

Tabela 2 – Tags de parágrafos e quebras de linha [RAMA, 2002]

2.2.1.2.2 *Estilos de texto*

Assim como no editor de texto, você pode criar uma série de efeitos no texto, alterando a forma ou o tipo de fonte. Todos as tags que alteram o estilo do texto precisam ser especificados em pares, marcando o início e o fim do texto que sofrerá a formatação[RAMA , 2002].

Tag	Descrição
 texto 	Aplica o estilo negrito
<I> texto </I>	Aplica o estilo itálico
<U> texto </U>	Aplica um sublinhado
 texto 	Similar ao negrito
<TT> texto </TT>	Deixa o texto com espaçamento regular
<BIG> texto </BIG>	Aumenta a fonte e aplica negrito
<SMALL> texto </SMALL>	Reduz e altera a fonte
^{texto}	Eleva o texto e diminui seu corpo(sobrescrito).
_{texto}	Rebaixa o texto e diminui seu corpo (subscrito).
<BLINK> texto </BLINK>	Faz o texto piscar
<PRE> texto </PRE>	Por meio desse tag pode-se incluir um texto que foi editado por um editor de texto e preservar a formatação

	original, como marcas de tabulação, fim de linha gerado pela tecla Enter e outros formatos.
<code><CENTER> texto </CENTER></code>	Como padrão os textos inseridos em uma página HTML são alinhados pela margem esquerda. Para centralizar um cabeçalho, parágrafo ou figura, deve ser usado a tag <code><CENTER> </CENTER></code> envolvendo o objeto que será centralizado.
<code><Hn> texto </Hn></code>	A tag <code><H></code> aplica um efeito negrito e altera o tamanho da letra. A linguagem HTML possui até seis tags, de <code><H1></code> até <code><H6></code> , que aplicam um tamanho de fonte diferenciado para o texto que envolvem. A tag <code><H></code> insere uma linha em branco antes e depois da linha que contém o comando [RAMA, 2002]. Eles são perfeitos para criar títulos e iniciar seções dentro de uma página.

Tabela 3 - Tags de estilo de texto [RAMA, 2002]

2.2.1.2.3 Cores no texto e no fundo da página

A tag responsável por alterar o tamanho, a cor e o tipo da fonte de um texto utilizado é ` ` [RAMA, 2002].

A sintaxe é mostrada pelo exemplo abaixo:

```
<FONT SIZE = 6 FACE = 'verdana' COLOR = 'blue'> texto </FONT>
```

O parâmetro `SIZE = n` especifica o tamanho da fonte utilizada. Os valores vão de 1, o menor, até 7, o maior. O padrão é 3.

O parâmetro `FACE` permite que se escolha uma fonte diferente para o texto.

O parâmetro `COLOR` especifica a cor do texto. Seu valor pode ser especificado em hexadecimal ou por um nome predefinido de cores.

Neste tag não é necessário o uso de todos os parâmetros.

Para as cores e padrões de fundo há uma mudança na sintaxe da tag `<BODY>`, que passa a ser como o do exemplo a seguir:

```
<BODY BGCOLOR=#FFFF00 TEXT= "White" LINK=#0000FF  
VLINK=#CC0000>
```

Neste exemplo o fundo é definido com a cor amarela, o texto branco, o link azul não visitado azul e o visitado vermelho.

BGCOLOR=	É a cor do fundo, onde o padrão é cinza ou branco, dependendo do browser.
TEXT=	É a cor do texto, em que o padrão é preto
LINK=	É a cor do texto do link, em que o padrão é azul.
VLINK=	É a cor do link que já foi visitado. A cor padrão é vermelho-púrpuro.

Tabela 4 - Padrões de fundo [RAMA, 2002]

2.2.1.2.4 *Imagens e links*

Para se incluir uma imagem em uma página HTML, esta imagem deve estar nos formatos GIF ou JPG. O formato GIF é o que mais se popularizou no início da Web. Porém, o formato JPG tomou seu lugar pelas vantagens que oferece em termos de tamanho da imagem. No formato JPG, uma imagem pode ser reduzida em até dez vezes comparada ao seu tamanho original [RAMA, 2002].

A tag que define a posição de uma imagem possui os seguintes parâmetros:

SRC= "URL"	É o nome da figura, se estiver no mesmo diretório que o programa ou o seu caminho completo.
ALT= "texto"	Exibe o texto, quando o browser não encontra a imagem, ou uma legenda, quando o cursor é posicionado sobre ela mostrando o texto especificado.
ALIGN= "TOP" "MIDDLE" "BOTTOM"	Especifica o alinhamento da imagem relativamente a linha de texto onde é exibida.
ALIGN= "LEFT" "RIGHT"	Especifica o alinhamento da imagem relativamente às bordas laterais da janela.
ISMAP	Determina que a imagem é um mapa de imagem.
WIDTH= "número" "número%"	Esse parâmetro especifica a largura de exibição da imagem independentemente do seu tamanho físico. Se a imagem for maior ou menor do que o valor especificado, ela será esticada ou comprimida para caber no espaço determinado. Se for colocado o sinal de percentual, ele é relativo à largura da janela.
HEIGHT= "número"	Especifica a altura de exibição da imagem independentemente do seu tamanho físico. Se a imagem for maior ou menor do que o valor

	especificado, ela será esticada ou comprimida para caber no espaço determinado.
BORDER= “número”	Especificada em pixels a largura da borda da imagem. O valor zero remove a imagem.
VSPACE= “número”	Determina em pixels o espaço que deve ser deixado em branco na parte superior e inferior da imagem.
HSPACE= “número”	Determina em pixels o espaço que deve ser deixado em branco nas laterais da imagem.

Tabela 5 - Parâmetros da tag IMG [RAMA, 2002]

Um exemplo de aplicação dessa tag:

```
<IMG SRC= “logo.gif” WIDTH=33 HEIGHT=25 BORDER=0 ALT= “ ”>
```

Com a exceção do parâmetro SRC, todos os demais são opcionais.

A tag <A> pode marcar um texto ou uma imagem como link, assim como criar um endereço para uma seção de um documento que será referida por um link.

Sintaxe básica:

```
<A HREF= “. . .” NAME= “. . .”>caracteres </A>
```

HREF= “URL”	Especifica o endereço do URL o qual o link está associado. Pode ser usada para referências dentro ou fora do documento.
NAME= “string”	Especifica o nome da seção de um documento que é referida por um link de hipertexto.

Tabela 6 - Principais parâmetros da tag<A>[RAMA, 2002]

Para vincular uma imagem basta especificar o nome do arquivo, incluindo sua extensão, no parâmetro HREF.

2.2.1.2.5 Listas

Existem dois tipos de listas. As não ordenadas que contêm uma série de itens sem numeração, e as ordenadas, que atribuem um número para cada elemento da lista.

Tag	Descrição
	A tag que gera a lista não ordenada, que deve envolver o primeiro e o último item da lista. Cada item da lista deve ser precedido da tag .
	A tag que gera a lista ordenada, que deve envolver o

	primeiro e o último item da lista. Cada item da lista deve ser precedido da tag.
--	--

Tabela 7 – Tags para gerar listas[RAMA, 2002].

2.2.1.2.6 Tabelas

A tag usado para criar tabelas é <TABLE> </TABLE>.

Tag	Tag	Finalidade
<TABLE>	</TABLE>	Cria a estrutura básica da tabela sem linhas e colunas.
<TABLE BORDER>	</TABLE>	Cria a estrutura básica da tabela com bordas.
<TR>	</TR>	Cria uma linha dentro da tabela.
<TD>	</TD>	Define uma célula dentro de uma linha.
<TH>	</TH>	Cria uma célula de título com o texto centralizado e em negrito.
<CAPTION>	</CAPTION>	Cria um título para a tabela acima dela.

Tabela 8 - Tags para a construção de tabelas[RAMA, 2002].

Neste trabalho o foco será apenas em tabelas simples.

2.2.1.2.7 Formulários

Os formulários são os responsáveis pelo fato de um usuário poder interagir com o servidor, enviando dados que serão processados nesse servidor e posteriormente retornados ao cliente[RAMA, 2002]. Porém é necessário que ao lado do servidor exista outro programa, escrito em outra linguagem de programação que será responsável pelo recebimento e o processamento dos formulários.

Para definir um formulário é necessário a tag <FORM> . . . </FORM>. Ele tem duas funções importantes que são especificar o local do programa que controlará o formulário e definir a forma ou o método como os dados serão enviados[RAMA, 2002].

Parâmetro	Finalidade
ACTION= “URL”	Especifica o local do servidor que vai processar o formulário.
METHOD= “método de troca de dados”	Indica o método usado pelo servidor para receber o formulário. Temos duas opções que são: POST e GET. O método POST transmite toda a informação do formulário imediatamente após o URL. E o GET faz o

	conteúdo do formulário ser anexado ao endereço do URL.
--	--

Tabela 9 – Parâmetros da tag <FORM>[RAMA ,2002].

A tag <INPUT> define um campo de entrada onde o usuário digita as informações do formulário. Cada campo atribui o seu conteúdo para uma variável que possui nome e tipo específicos.

TYPE= “tipo de variável”	Especifica o tipo de dado para a variável.
TYPE= “TEXT”	Aceita dados do tipo caractere.
TYPE= “PASSWORD”	Aceita dados do tipo caractere.
TYPE= “CHECKBOX”	Aceita campos do tipo sim/não
TYPE= “RADIO”	Mostra várias opções e aceita uma delas.
TYPE= “SUBMIT”	Cria um botão que envia, ao ser pressionado, o conteúdo do formulário.
TYPE= “RESET”	Cria um botão que, ao ser clicado, exclui o conteúdo de todos os campos.
TYPE= “HIDDEN”	Define um campo invisível, cujo conteúdo é enviado junto com os demais.
NAME= “TEXTSTRING”	Textstring deve ser substituído pelo nome da variável que receberá o conteúdo do campo.
VALUE= “TEXTSTRING”	Para os campos “TEXT” ou “PASSWORD”, representa o valor padrão do campo. Para campos do tipo CHECKBOX ou RADIO, representa o valor que será enviado se não for especificado outro. Para campos do tipo SUBMIT ou RESET, indica a etiqueta que aparece no botão.
CHECKED	Para campos do tipo CHECKBOX, indica que o campo deve ficar marcado como padrão.
SIZE= “tamanho”	“Tamanho” é um número que representa a quantidade de caracteres exibidos em campos do tipo TEXT ou PASSWORD. Esse número delimita apenas o tamanho do campo para digitação, e não a quantidade de caracteres que o campo pode ter.
MAXLENGTH= “comprimento”	“Comprimento” é um valor que define o número de caracteres permitidos para variáveis do tipo TEXT ou PASSWORD.

Tabela 10 – Parâmetros da tag <INPUT>[RAMA, 2002].

A tag <SELECT> . . . </SELECT> define e exhibe uma lista de itens que podem ser selecionados pelo usuário. Cada item da lista deve ser especificado por meio do parâmetro<OPTION>.

Parâmetro	Finalidade
NAME= “TEXTSTRING”	“Textstring” deve ser substituído pelo nome da variável que receberá o conteúdo do campo.

SIZE= "Tamanho"	Neste parâmetro opcional, Tamanho é um numero que representa a quantidade de itens que será exibida de uma vez pela lista.
MULTIPLE	Esse parâmetro opcional indica que podem ser escolhidos diversos itens da lista.

Tabela 11 - Parâmetros da tag <SELECT>[RAMA, 2002].

2.2.2 A ORIGEM E FUNCIONALIDADE DE WML

A WML (Wireless Markup Language) é uma linguagem de marcação usada para descrever a estrutura dos documentos que serão transmitidos nos dispositivos sem fio. Ela foi criada para solucionar limitações de tela, largura de banda e memória dos dispositivos móveis e sem fio. Como ela foi projetada para ser executada em vários dispositivos, a WML adota muito pouco do dispositivo que está executando o aplicativo e proporciona bem menos controle sobre os formatos de saída do que a HTML[FORT, 2003]. Assim como a HTML é baseada na SGML, a WML é baseada na XML (Extensible Markup Language). A XML é uma sub-linguagem da SGML que também é utilizada para definir outras linguagens que precise de uma arquitetura mais simples.

A WML oferece suporte a seis áreas principais[FORT, 2003]:

- Layout e apresentação de texto: embora a saída do código WML produzida varie entre os dispositivos e navegadores WML específicos, ela oferece suporte a quebras de linha, formatação de texto e alinhamento.
- Imagens: embora os dispositivos compatíveis com WAP não precisem oferecer suporte a imagens, a WML suporta o formato de imagem Wireless Bitmap (WBMP) e o alinhamento de imagens na tela. Esse formato gráfico foi criado pelo WAP Fórum especialmente para dispositivos móveis.
- Entrada do usuário: oferece suporte a listas de opções, listas de opções com vários níveis, entrada de texto e controles de tarefa.

- Organização dos cartões e decks: as interações com o usuário dividem-se em cartões e a navegação ocorre entre eles. Os decks são conjuntos relacionados que formam um arquivo simples, como um arquivo HTML simples. Na HTML, ver uma página é o mesmo que ver um cartão na WML. Mas, em vez de cada página HTML representar um arquivo HTML, vários cartões WML representam um deck WML, que é salvo como um único arquivo.
- Navegação: o WAP oferece suporte aos links ancorados e ao esquema de nomenclatura de URLs da Internet padrão, permitindo a navegação entre cartões em um deck, entre decks ou entre outros recursos na rede.
- Gerenciamento de estado e de contexto: para maximizar os recursos da rede, o WAP permite que variáveis sejam transmitidas entre arquivos WML. Em vez de enviar uma seqüência completa, as variáveis podem ser enviadas e substituídas no tempo de execução. O agente usuário pode armazenar em cache variáveis e arquivos WML, minimizando ocorrências de cachê e solicitações do servidor.

A WML é uma linguagem descritora de documentos baseada em tags assim como a HTML [FREI, 2001]. A linguagem WML é mais rígida na sua sintaxe, ela diferencia maiúsculas de minúsculas, não se pode utilizar espaços em branco entre um atributo e um sinal de igualdade e muitas outras regras.

A WML sendo derivada da XML dispõe de uma maneira através da qual o servidor WAP e os compiladores WML possam fazer referência à especificação XML. Para isso, a XML precisa de um prólogo que defina a versão da XML e um ponteiro para a linguagem ou definição XML que está sendo usada.

Em geral, um dispositivo WAP nunca verá esse prólogo porque o servidor e o compilador são os únicos dispositivos que se preocupam com a definição do formato XML[FORT, 2003].

Ex:

```
<?xml version= "1.0" ?>

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2 //EN
"http://www.wapforum.org/DTD/wml12.dtd">
```

O prólogo deve ser incluído em todos os arquivos WML.

2.2.2.1 DECKS E CARDS

Ao desenvolver para a Web, cada arquivo HTML representa uma página HTML. O desenvolvimento em WML é um pouco diferente.[FORT, 2003] Como cada página ou tela é muito pequena, não faz sentido cada um constituir um arquivo separado. As páginas WML – o conteúdo exibido em telas separadas – são denominadas cartões e esses cartões são colocados dentro de um deck de páginas relacionadas, que compõem um único arquivo.

Cada deck contém um ou vários cards, os quais podem ser definidos como sendo as telas de informação a serem visualizadas pelos usuários[FREI, 2001] . Cada card define a aparência de cada tela, seu funcionamento e as etapas que são realizadas quando o card é navegado. Assim, um card nunca pode estar vazio e precisa conter, pelo menos, um elemento[FORT, 2003] .

2.2.2.2 ESTRUTURA BÁSICA DE UMA PÁGINA WML

```
<?xml version= "1.0" ?>

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2 //EN
"http://www.wapforum.org/DTD/wml12.dtd">

<wml>

<head>. . . </head>

<template> . . . </template>

<card> . . . </card>
```

novos elementos cards opcionais

</wml>

Essa é uma visão geral do código. As primeiras linhas são declarações em XML. Elas especificam o tipo e a versão do documento e devem ser incluídas em todos os arquivos WML. O documento inteiro está incluído entre as tags <wml> e </wml>. Isso é necessário e toda página WML deve ter um conjunto de tags <wml> correspondentes. Os cards devem aparecer entre esses tags. O card é definido com a tag <card> e termina com a tag </card>. Todo o conteúdo do card deve estar incluído entre esses tags [FORT, 2003].

Qualquer código que aparecer entre as tags <template> </template> se aplicará a todos os cards do deck como se estivesse sido digitado diretamente naquele local[FORT, 2003]. A tag <head> contém a informação relacionando o cabeçalho ao todo inclusive tags de controle de acesso.

2.2.2.3 TAGS DE WML

Como já foi visto, a WML é menos extensa que a HTML e seu conjunto de tags é menor.

Tag	Descrição
<p> ... </p>	É a base de saída do cartão. O texto não pode ser colocado em um cartão diretamente; os parágrafos devem ser colocados nos cartões e o texto dentro de parágrafos.
 	Esse tag insere uma quebra de linha.
<a> ... 	Especifica o texto dentro das tags um hiperlink. Possui o atributo href que tem o endereço do destino que pode ser um URL.
<anchor> ... </anchor>	Esse tag define o texto que aparecerá como hyperlink mas não fornece o mecanismo para realmente enviar o usuário para um local diferente. Para isso é necessário um elemento <go> aninhado.
<access>	Aplica regras de controle de acesso a um deck, restringindo o acesso. Esse tag deve ser especificado entre as tags <head> </head>.
<go href= "# destino"/>	Esse tag instrui o dispositivo a navegar para determinado local, mas só o faz no contexto de alguma ação do usuário. Possui o parâmetro href que define o destino. Precisa ser vinculado a outra tag como o <anchor> ou mais usualmente, <do>.
<do type= "botão"> ... </do>	Esse tag exige que seja especificado um tipo de botão. Ele associa um URL a um tipo de botão. Entre as tags

	<code><do> </do></code> aparece a ação que deve ser executada.
<code><prev/></code>	Esse tag funciona como um botão de retorno e retorna ao cartão anterior. Aparece dentro da tag <code><do></code> .
<code><noop/></code>	Especifica que nenhuma operação deveria ser feita. Isso pode ser usado em um cartão para cancelar uma tarefa que foi especificada em um template a nível de deck.
<code><refresh> </refresh></code>	Esse tag indica uma atualização da tela e contexto do dispositivo como especificado pela tag <code>setvar</code> , por exemplo. Efeitos do lado do usuário (por exemplo, uma mudança na exibição de tela) poderão acontecer durante o processo de atualização.
<code><setvar></code>	Esse tag especifica a variável para começar o contexto do browser atual como o efeito de executar uma tarefa definida. O elemento será ignorado se o atributo nome não utilizar uma variável legal.
<code><postfield></code>	O elemento <code>postfield</code> especifica um nome de um campo e o valor para transmissão a um servidor de origem durante uma requisição de URL. A codificação atual do nome e valor dependerá do método usado para comunicar-se com o servidor de origem.
<code>... </code>	Escolhe uma fonte que indique ênfase.
<code>... </code>	Escolhe uma fonte que indique ênfase forte.
<code>... </code>	Coloca a fonte em negrito.
<code><i>... </i></code>	Coloca a fonte em itálico.
<code><u>... </u></code>	Sublinha o texto.
<code><small>... </small></code>	Diminui o tamanho da fonte.
<code><big>... </big></code>	Aumenta o tamanho da fonte.
<code><table>... </table></code>	Define uma tabela. Possui alguns parâmetros como: <code>columns</code> para definir o número de colunas, <code>title</code> que especifica para a tabela uma string que o navegador possa exibir ao usuário.
<code><tr> </tr></code>	Define uma linha na tabela. Deve ser utilizado entre as tags <code><table></code> .
<code><td> </td></code>	Define células e deve vir entre as tags <code><tr></code> .
<code><pre>... </pre></code>	Proporciona um maior controle sobre a aparência do texto quando o navegador exibe.
<code></code>	Esse tag indica que uma imagem será incluída no fluxo do texto. O layout da imagem é feito dentro do contexto do layout do texto.
<code><!-- -- ></code>	Indica comentário
<code><input ... /></code>	Esse tag coleta dados do usuário. A entrada do usuário é restrita pelo seu atributo <code>Format</code> (opcional).
<code><select>... </select></code>	Apresenta opções como uma lista de itens de texto na qual o usuário pode selecionar um ou mais itens.
<code><option></code>	Especifica uma única opção de escolha em um elemento selecionado.

<optgroup>	Permite agrupar elementos relacionados em uma hierarquia. Os browsers usam essa hierarquia para facilitar o layout de apresentação em uma grande variedade de dispositivos.
<fieldset>	Permite o agrupamento de campos relacionados e texto. Esse agrupamento permite otimizar o layout e navegação.

Tabela 12 – Tags de WML [FORT, 2002].

Em WML deve-se evitar incorporar no texto bruto, caracteres que sejam reservados da linguagem. Para isso são definidos caracteres especiais que são substituídos pelo caractere que se deseja exibir[FORT, 2002].

Entidade com nome	Numérica	Descrições
"	"	Aspas (“)
&	&	E comercial (&)
&após;	'	Apostrofo (‘)
<	<	Símbolo de menor que (<)
>	>	Símbolo de maior que (>)
 	 	Espaço em branco incondicional ()
­	­	Hífen condicional (- ou nada)

Tabela 13 – Entidades de caracteres especiais da WML[FORT, 2002]

3 MODELO DE TRADUÇÃO DE LINGUAGENS HTML - WML

Estruturas de duas linguagens podem possuir a mesma sintaxe abstrata. Com base nisso pode-se chegar a um modelo de conversão de linguagens de marcação de texto. A maior parte das tags das linguagens HTML e WML são sintaticamente compatíveis, facilitando assim, a criação do modelo de tradução. As tags incompatíveis são tratadas ou são

simplesmente desprezadas pelo fato de os dispositivos móveis não poderem atender a todos os recursos que uma página Web suporta.

O modelo de tradução a ser desenvolvido parte do pressuposto que o arquivo HTML encontra-se integralmente validado, ou seja, livre de qualquer erro sintático ou semântico.

Foi definido como base de interface para esse modelo, o dispositivo móvel do fabricante Nokia, modelo 7110 (figura 5). Esse aparelho foi escolhido porque possui uma tela maior em relação a outros aparelhos e isso facilita a visualização. Ele possui as características a seguir [NOKIA, 2003]:

- Tela de tamanho 96 x 65 pixels
- Visualização : 6 linhas de 16 caracteres
- Micronavegador wml para internet
- Suporte de entrada de dados de alta velocidade (14.4 Kbps)



Figura 5 - Nokia 7110 [NOKIA, 2003]

Este trabalho não tem dentre seus objetivos se preocupar com adaptação do código gerado para interface de qualquer tipo de dispositivo. Ele tem o Nokia 7110 como base e se adapta respeitando as características deste. Para uma adaptação automática de interface para qualquer dispositivo consultar a tese de mestrado em[MESS, 2000].

Neste modelo, o modo de tradução é feito através da análise léxica. O documento HTML é lido da esquerda para direita e seu fluxo de caracteres é agrupado em tokens desprezando brancos desnecessários e comentários. Os tokens para esse modelo serão todas as unidades léxicas que estiverem entre os símbolos “<” e “>”, as chamadas tags. Esses tokens constituem classes de símbolos que são reconhecidos e transformados, permitindo assim, a geração de um documento correspondente em WML.



Figura 6 – Modelo de tradução

A seguir, é definida na tabela 14 as tags HTML que o tradutor reconhece e seu correspondente em WML.

Tag HTML	Tag WML
<HTML>	<wml>
</HTML>	</wml>
<P>	<p>
</P>	</p>
	
	
<I>	<i>
</I>	</i>
	
	
<U>	<u>

</U>	</u>
<SMALL>	<small>
</SMALL>	</small>
<BIG>	<big>
</BIG>	</big>
<PRE>	<pre>
</PRE>	</pre>
<TABLE>	<table columns= "16">
<TABLEBORDER>	<table columns= "16">
</TABLE>	</table>
<TR>	<tr>
</TR>	</tr>
<TD>	<td>
</TD>	</td>
<BODY>	<card>
</BODY>	</card>
<A HREF =	<a href=
	
<CENTER>	<p align= "center">
</CENTER>	</p>
<H1>	
</H1>	
<H2>	
</H2>	
<H3>	
</H3>	
<H4>	
</H4>	
<H5>	
</H5>	
<H6>	
</H6>	
<HR>	
<INPUT TYPE= "text" NAME= ">	<input name= ">

Tabela 14 – Tags correspondentes

O reconhecimento e transformação dos tokens do documento HTML acontece como definido na tabela 14. É localizado na tabela a tag que corresponde ao token lido e também seu correspondente em WML. A transformação ocorre de forma direta.

Pelo fato de os dispositivos móveis serem mais restritos, algumas tags HTML(tabela 15) não são suportadas e foram desprezadas por esse modelo de tradução.

Tag	Descrição
<TT>texto</TT>	Deixa o texto com espaçamento regular.
^{texto}	Eleva o texto e diminui seu corpo(sobrescrito).
_{texto}	Rebaixa o texto e diminui seu corpo (subscrito).
<BLINK>texto</BLINK>	Faz o texto piscar
texto	Define formato para a fonte do texto.
... 	A tag que gera a lista não ordenada
... 	Envolve os itens de uma lista
... 	A tag que gera a lista ordenada

Tabela 15 – Tags não suportadas

O modelo de tradução obedece às características do dispositivo escolhido como base, Nokia 7110, e os textos ficam restritos a 16 caracteres por linha. Como em WML é necessário definir a quantidade de colunas numa tabela, ficaram definidas 16 colunas.

A tag de parágrafo <p>... </p> é obrigatória para se incluir um texto num documento WML. Para o tratamento dessa tag considerou-se que a cada início e término de uma linha, essa tag é automaticamente incluída.

Toda tag de figura encontrada no documento HTML foi transformada para o documento WML como um texto indicando figura. Isso porque o dispositivo base suporta apenas figuras WBMP (Wireless Bitmap). Se além de figura, a tag indicar também um link, a URL dessa figura é exibida no documento WML.

Para o desenvolvimento do aplicativo foi utilizada a linguagem de programação Delphi 5, orientada a objetos, que com o auxílio de uma tabela com as tags correspondentes criada em Access, permitiu a transformação HTML – WML. Também foram implementadas todas as incompatibilidades levantadas anteriormente.

Na figura 7 é mostrada a tela inicial do programa de conversão.

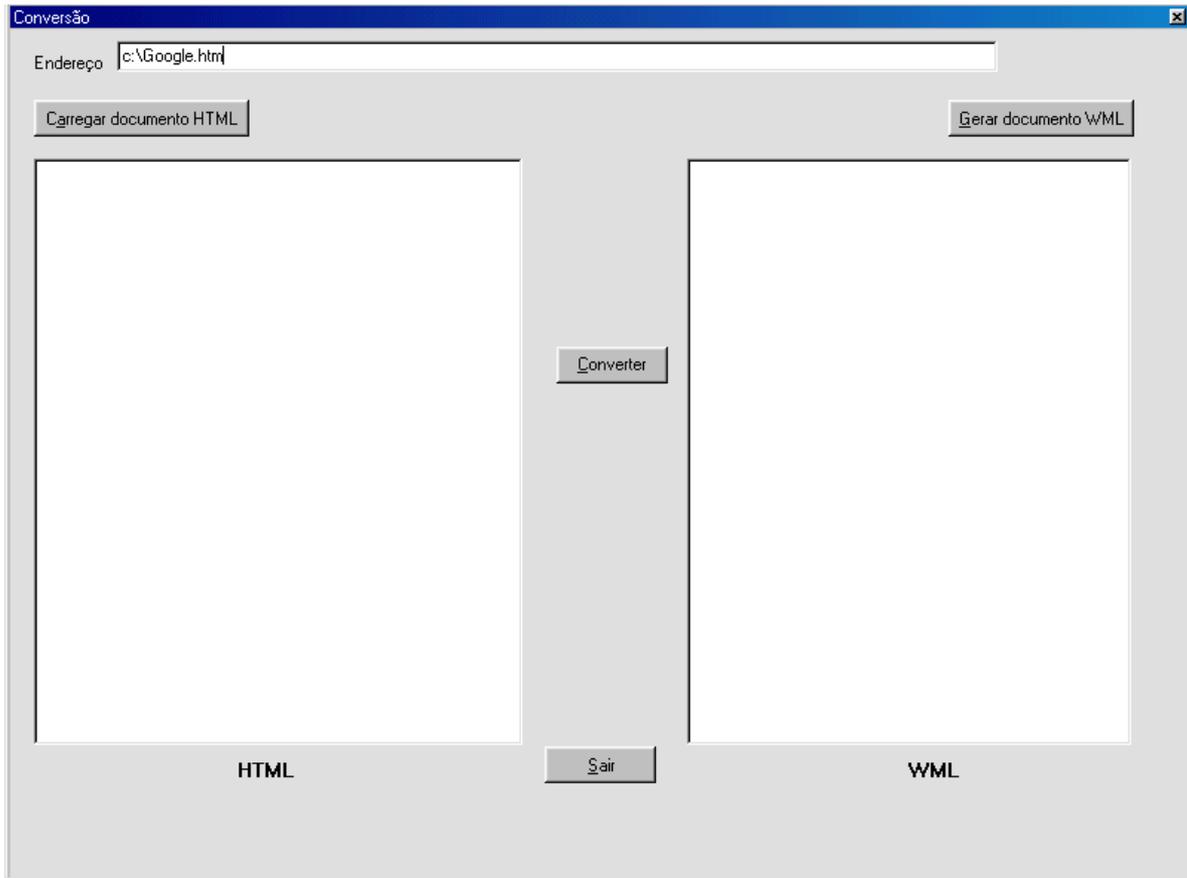


Figura 7 – Tela do programa de conversão

O edit endereço está fazendo o papel do servidor que fornece a URL do arquivo HTML. Após digitado o endereço do arquivo HTML é necessário clicar no botão: Carregar arquivo HTML, para que o mesmo possa ser exibido no ListBox HTML como mostrado na figura 8 . Assim que o documento HTML é carregado já é possível efetuar a conversão bastando para isso um clique no botão: Converter. A conversão para WML é exibida no ListBox WML como pode ser visto na figura 9.

Após a exibição da conversão pode-se gerar o documento WML através do botão: Gerar documento WML. Este botão permite que o usuário escolha o nome do arquivo WML e o local onde ele será salvo. Isso é mostrado pela figura 10.

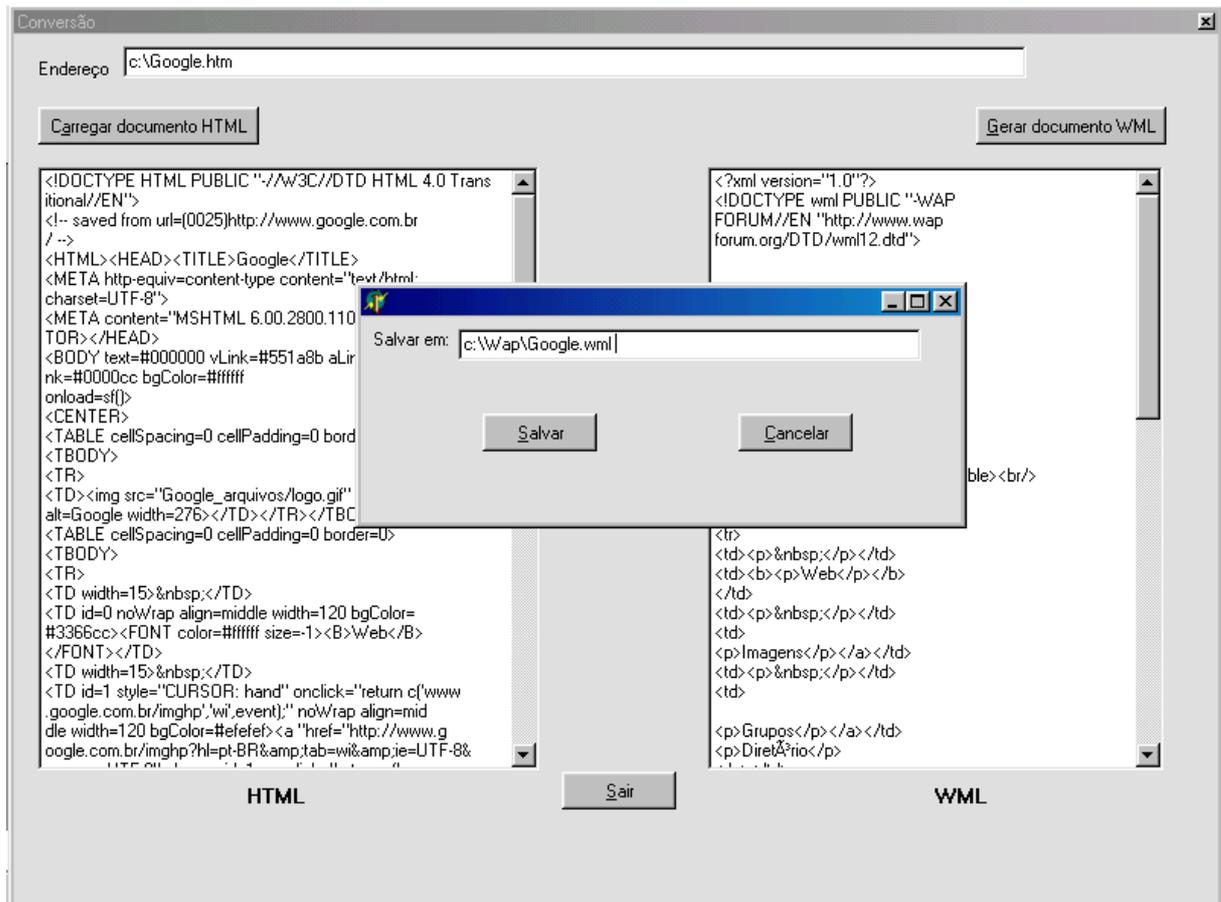


Figura 10 – Salvando documento WML

Na figura 11 podemos observar através das numerações, alguns exemplos de traduções. São algumas das tags incompatíveis, mas também podem ser observadas outras tags que têm sua tradução de forma direta através da tabela de equivalência. O número 1 está indicando a inclusão do prólogo que é feita em todos os arquivos WML. O número 2 mostra a transformação direta das tags iniciadoras de arquivo, <HTML> e <WML>. O número 3 exibe a tag de coluna definindo 16 colunas. O número 4 indica através de um texto “Figura”, a existência de uma figura no arquivo HTML que não é suportada em WML. O número 5 exibe a tag indicadora de link buscando a URL de origem do documento HTML.

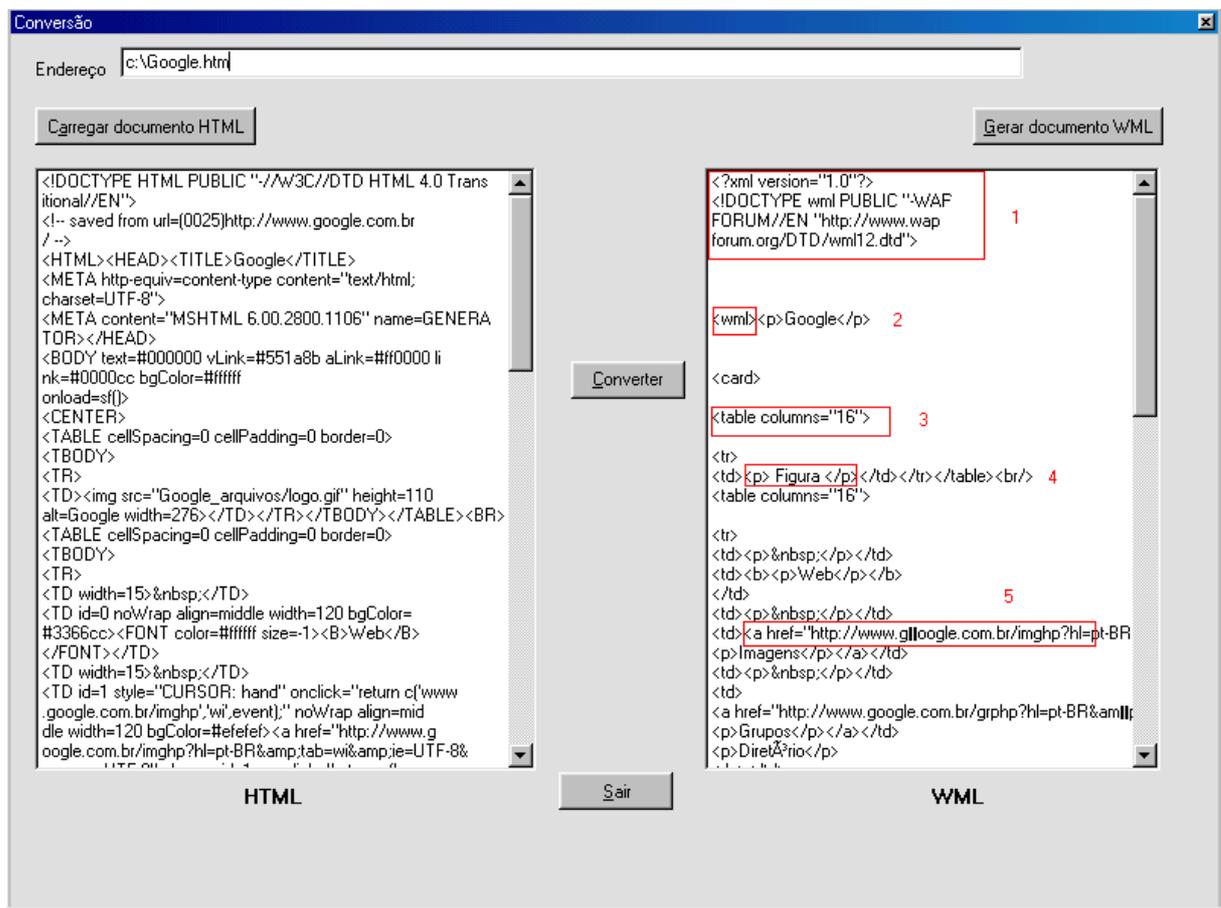
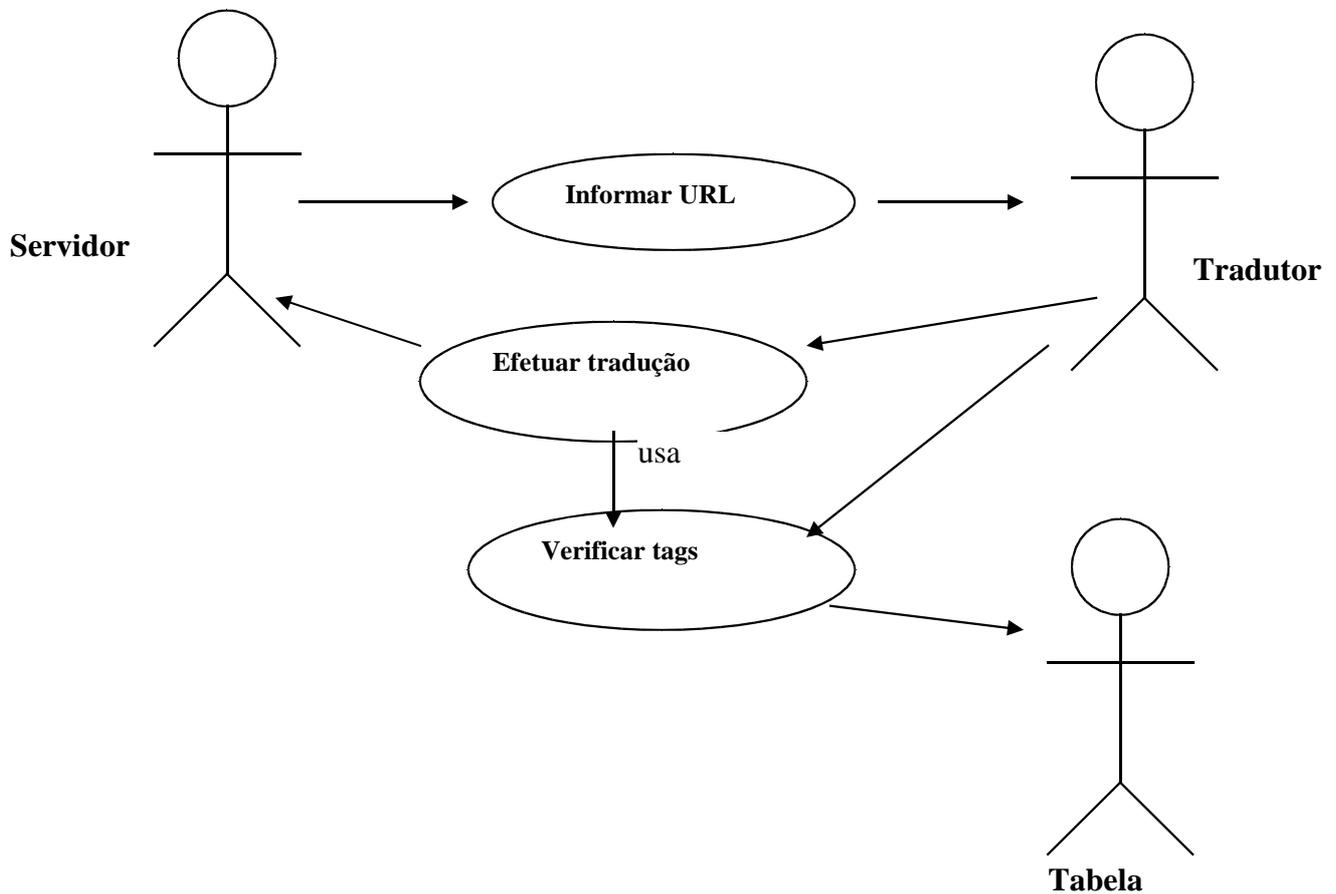


Figura 11 – Exemplos de tradução

3.1 MODELAGEM DO MODO DE TRADUÇÃO

3.1.1 DIAGRAMA DE CASO DE USO

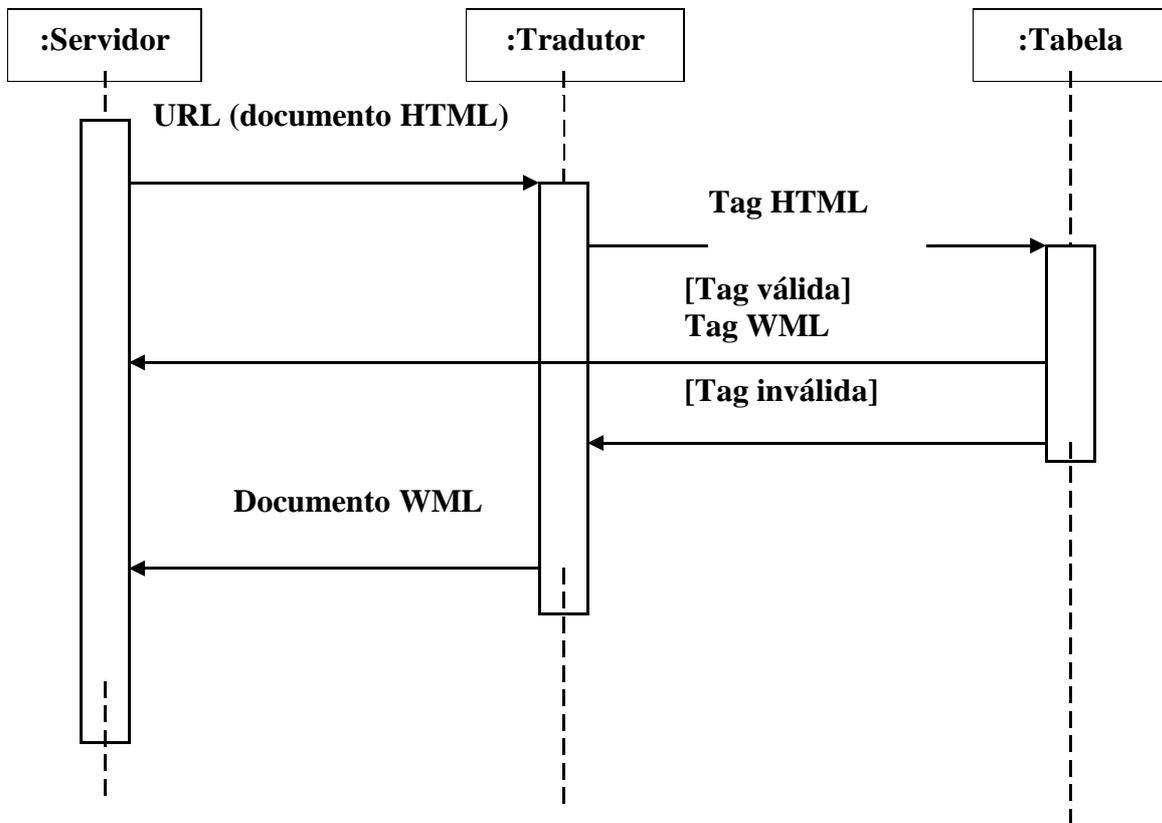


Informar URL: O servidor informa ao sistema tradutor a URL da página Web solicitada pelo usuário. Essa solicitação é feita através do WAP browser.

Verificar tags: O tradutor busca na tabela a tag em HTML.

Efetuar tradução: Se a tag HTML existe na tabela, ela é trocada por sua correspondente. Se a tag não é pertencente à tabela, ela é então tratada de forma especial ou simplesmente desprezada.

3.1.2 DIAGRAMA DE SEQUÊNCIA



Esse é o diagrama de seqüência que mostra os eventos da tradução. Inicialmente é informada a URL do documento HTML, o tradutor através da análise léxica retira cada tag desse documento e procura sua equivalência na tabela. Encontrando, é retornada a tag WML diretamente, formando o documento WML senão é retornada tag inválida para que o tradutor possa tratá-la para em seguida formar o documento WML.

4 CONCLUSÃO

Este trabalho mostrou um modelo de se implementar um conversor entre linguagens de marcação de texto a partir da análise léxica de uma linguagem de origem e gerar os templates capazes de representá-la em uma linguagem destino. O fato das duas linguagens escolhidas, HTML e WML, possuírem a maior parte das tags semelhantes facilitou a conversão.

Considerado a reestruturação de idéia original do projeto, que demandou um tempo considerável, a implementação do modelo de conversão proposto foi feito utilizando a comparação de strings. Esse método foi satisfatório mostrando boa eficiência na tradução.

Com todo o estudo feito sobre as linguagens de marcação consideradas e a tecnologia WAP, pode-se concluir que devido ao grande volume de informação existente no formato HTML e ao crescimento acelerado dos acessos por dispositivos portáteis, é de extrema necessidade que tais informações sejam adaptadas, para que se torne efetiva a realidade de acesso móvel sem fio à *Internet*.

Como sugestão para trabalhos futuros temos a expansão do modelo de tradução contido nesse trabalho para suportar características de outros dispositivos, verificação de erros na tradução, um modo de tradução mais dinâmico utilizando máquinas de estados e a introdução de um simulador wap para exibir os resultados obtidos na tradução.

REFERÊNCIAS BIBLIOGRÁFICAS

- [BAX, 2001] BAX, Marcelo Peixoto. **Introdução às linguagens de marca**, Escola de Ciência da Computação, UFMG. <cuba.paradigma.com.br/paradigma/artigos/artigos_02.pdf> Acesso em 09/09/2003.
- [CHAI, 2001] CHAIBEN, Hamilton. **Tutorial de HTML**, 2001. UFPA.
- [FORT, 2003] Forta Bem, Lauver Keith, Fonte Paul, Juncker M. Robert, Mandel Ronan, Bromby Dylan, **Desenvolvendo WAP com WML e WMLScript**, Campus, 2001.
- [FREI, 2001] FREITAS, Wagner L.C. de, **Um Framework para a construção de sistemas de monitoração, alerta e intervenção remotos**, 2001. UFP.
- [MARC, 2001] **O wap a serviço do cidadão, como complemento à Internet**. <http://www.abep.sp.gov.br> Acesso em 23/08/2003.
- [MESS, 2000] MESSIAS, Edierley B. **Técnicas de adaptação automática de conteúdo para dispositivos WAP**, 2000. Tese - Mestrado em Ciência da computação, UFMG, Belo Horizonte.
- [NOKIA, 2003] Nokia Brasil. <<http://www.nokia.com/phones/7110/index.htm>> Acesso em 11 nov. 2003.
- [RAMA, 2002] RAMALHO, Antônio José. **Teoria e Prática HTML 4**. Série Ramalho: Bekerley, 2002.

ANEXO A – DTD DA LINGUAGEM WML

Document Type Definition (DTD)

```

<!--
Wireless Markup Language (WML) Document Type Definition.
WML is an XML language. Typical usage:
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<wml>
...
</wml>
-->
<!ENTITY % length "CDATA"> <!-- [0-9]+ for pixels or [0-9]+%" for
percentage length -->
<!ENTITY % vdata "CDATA"> <!-- attribute value possibly containing
variable references -->
<!ENTITY % HREF "%vdata;"> <!-- URI, URL or URN designating a hypertext
node. May contain variable references -->
<!ENTITY % boolean "(true|false)">
<!ENTITY % number "NMTOKEN"> <!-- a number, with format [0-9]+ -->
<!ENTITY % coreattrs "id ID #IMPLIED
class CDATA #IMPLIED">
<!ENTITY % ContentType "%vdata;"> <!-- media type. May contain variable
references -->
<!ENTITY % emph "em | strong | b | i | u | big | small">
<!ENTITY % layout "br">
<!ENTITY % text "#PCDATA | %emph;">
<!-- flow covers "card-level" elements, such as text and images -->
<!ENTITY % flow "%text; | %layout; | img | anchor | a | table">
<!-- Task types -->
<!ENTITY % task "go | prev | noop | refresh">
<!-- Navigation and event elements -->
<!ENTITY % navelmts "do | onevent">
<!--===== Decks and Cards =====>
<!ELEMENT wml ( head?, template?, card+ )>
<!ATTLIST wml
xml:lang NMTOKEN #IMPLIED
%coreattrs;
>
<!-- card intrinsic events -->
<!ENTITY % cardev
"onenterforward %HREF; #IMPLIED
onenterbackward %HREF; #IMPLIED
ontimer %HREF; #IMPLIED"
>
<!-- card field types -->
<!ENTITY % fields "%flow; | input | select | fieldset">
<!ELEMENT card (onevent*, timer?, (do | p | pre)*)>
<!ATTLIST card

```

```

title %vdata; #IMPLIED
newcontext %boolean; "false"
ordered %boolean; "true"
xml:lang NMTOKEN #IMPLIED
%cardev;
%coreattrs;
>
<!--===== Event Bindings =====>
<!ELEMENT do (%task;)>
<!ATTLIST do
type CDATA #REQUIRED
label %vdata; #IMPLIED
name NMTOKEN #IMPLIED
optional %boolean; "false"
xml:lang NMTOKEN #IMPLIED
%coreattrs;
>
<!ELEMENT onevent (%task;)>
<!ATTLIST onevent
type CDATA #REQUIRED
%coreattrs;
>
<!--===== Deck-level declarations =====>
<!ELEMENT head ( access | meta )+>
<!ATTLIST head
%coreattrs;
>
<!ELEMENT template (%navelmts;)*>
<!ATTLIST template
%cardev;
%coreattrs;
>
<!ELEMENT access EMPTY>
<!ATTLIST access
domain CDATA #IMPLIED
path CDATA #IMPLIED
%coreattrs;
>
<!ELEMENT meta EMPTY>
<!ATTLIST meta
http-equiv CDATA #IMPLIED
name CDATA #IMPLIED
forua %boolean; "false"
content CDATA #REQUIRED
scheme CDATA #IMPLIED
%coreattrs;
>
<!--===== Tasks =====>
<!ELEMENT go (postfield | setvar)*>
<!ATTLIST go
href %HREF; #REQUIRED
sendreferer %boolean; "false"
method (post|get) "get"
enctype %ContentType; "application/x-www-form-urlencoded"
accept-charset CDATA #IMPLIED
%coreattrs;

```

```

>
<!ELEMENT prev (setvar)*>
<!ATTLIST prev
%coreattrs;
>
<!ELEMENT refresh (setvar)*>
<!ATTLIST refresh
%coreattrs;
>
<!ELEMENT noop EMPTY>
<!ATTLIST noop
%coreattrs;
>
<!--===== postfield =====>
<!ELEMENT postfield EMPTY>
<!ATTLIST postfield
name %vdata; #REQUIRED
value %vdata; #REQUIRED
%coreattrs;
>
<!--===== variables =====>
<!ELEMENT setvar EMPTY>
<!ATTLIST setvar
name %vdata; #REQUIRED
value %vdata; #REQUIRED
%coreattrs;
>
<!--===== Card Fields =====>
<!ELEMENT select (optgroup|option)+>
<!ATTLIST select
title %vdata; #IMPLIED
name NMTOKEN #IMPLIED
value %vdata; #IMPLIED
iname NMTOKEN #IMPLIED
ivalue %vdata; #IMPLIED
multiple %boolean; "false"
tabindex %number; #IMPLIED
xml:lang NMTOKEN #IMPLIED
%coreattrs;
>
<!ELEMENT optgroup (optgroup|option)+ >
<!ATTLIST optgroup
title %vdata; #IMPLIED
xml:lang NMTOKEN #IMPLIED
%coreattrs;
>
<!ELEMENT option (#PCDATA | onevent)*>
<!ATTLIST option
value %vdata; #IMPLIED
title %vdata; #IMPLIED
onpick %HREF; #IMPLIED
xml:lang NMTOKEN #IMPLIED
%coreattrs;
>
<!ELEMENT input EMPTY>
<!ATTLIST input

```

```

name NMTOKEN #REQUIRED
type (text|password) "text"
value %vdata; #IMPLIED
format CDATA #IMPLIED
emptyok %boolean; "false"
size %number; #IMPLIED
maxlength %number; #IMPLIED
tabindex %number; #IMPLIED
title %vdata; #IMPLIED
accesskey %vdata; #IMPLIED
xml:lang NMTOKEN #IMPLIED
%coreattrs;
>
<!ELEMENT fieldset (%fields; | do)* >
<!ATTLIST fieldset
title %vdata; #IMPLIED
xml:lang NMTOKEN #IMPLIED
%coreattrs;
>
<!ELEMENT timer EMPTY>
<!ATTLIST timer
name NMTOKEN #IMPLIED
value %vdata; #REQUIRED
%coreattrs;
>
<!--===== Images =====>
<!ENTITY % IAlign "(top|middle|bottom)" >
<!ELEMENT img EMPTY>
<!ATTLIST img
alt %vdata; #REQUIRED
src %HREF; #REQUIRED
localsrc %vdata; #IMPLIED
vspace %length; "0"
hspace %length; "0"
align %IAlign; "bottom"
height %length; #IMPLIED
width %length; #IMPLIED
xml:lang NMTOKEN #IMPLIED
%coreattrs;
>
<!--===== Anchor =====>
<!ELEMENT anchor ( #PCDATA | br | img | go | prev | refresh )*>
<!ATTLIST anchor
title %vdata; #IMPLIED
accesskey %vdata; #IMPLIED
xml:lang NMTOKEN #IMPLIED
%coreattrs;
>
<!ELEMENT a ( #PCDATA | br | img )*>
<!ATTLIST a
href %HREF; #REQUIRED
title %vdata; #IMPLIED
accesskey %vdata; #IMPLIED
xml:lang NMTOKEN #IMPLIED
%coreattrs;
>

```

```

<!--===== Tables =====>
<!ELEMENT table (tr)+>
<!ATTLIST table
title %vdata; #IMPLIED
align CDATA #IMPLIED
columns %number; #REQUIRED
xml:lang NMTOKEN #IMPLIED
%coreattrs;
>
<!ELEMENT tr (td)+>
<!ATTLIST tr
%coreattrs;
>
<!ELEMENT td ( %text; | %layout; | img | anchor | a )*>
<!ATTLIST td
xml:lang NMTOKEN #IMPLIED
%coreattrs;
>
<!--===== Text layout and line breaks =====>
<!ELEMENT em (%flow;)*>
<!ATTLIST em
xml:lang NMTOKEN #IMPLIED
%coreattrs;
>
<!ELEMENT strong (%flow;)*>
<!ATTLIST strong
xml:lang NMTOKEN #IMPLIED
%coreattrs;
>
<!ELEMENT b (%flow;)*>
<!ATTLIST b
xml:lang NMTOKEN #IMPLIED
%coreattrs;
>
<!ELEMENT i (%flow;)*>
<!ATTLIST i
xml:lang NMTOKEN #IMPLIED
%coreattrs;
>
<!ELEMENT u (%flow;)*>
<!ATTLIST u
xml:lang NMTOKEN #IMPLIED
%coreattrs;
>
<!ELEMENT big (%flow;)*>
<!ATTLIST big
xml:lang NMTOKEN #IMPLIED
%coreattrs;
>
<!ELEMENT small (%flow;)*>
<!ATTLIST small
xml:lang NMTOKEN #IMPLIED
%coreattrs;
>
<!ENTITY % TAlign "(left|right|center)">
<!ENTITY % WrapMode "(wrap|nowrap)" >

```

```

<!ELEMENT p (%fields; | do)*>
<!ATTLIST p
align %TAlign; "left"
mode %WrapMode; #IMPLIED
xml:lang NMTOKEN #IMPLIED
%coreattrs;
>
<!ELEMENT br EMPTY>
<!ATTLIST br
%coreattrs;
>
<!ELEMENT pre "(#PCDATA | a | br | i | b | em | strong | input | select )
*">
<!ATTLIST pre
xml:space CDATA #FIXED "preserve"
%coreattrs;
>
<!ENTITY quot "&#34;"> <!-- quotation mark -->
<!ENTITY amp "&#38;#38;"> <!-- ampersand -->
<!ENTITY apos "&#39;"> <!-- apostrophe -->
<!ENTITY lt "&#38;#60;"> <!-- less than -->
<!ENTITY gt "&#62;"> <!-- greater than -->
<!ENTITY nbsp "&#160;"> <!-- non-breaking space -->
<!ENTITY shy "&#173;"> <!-- soft hyphen (discretionary hyphen) -->

```

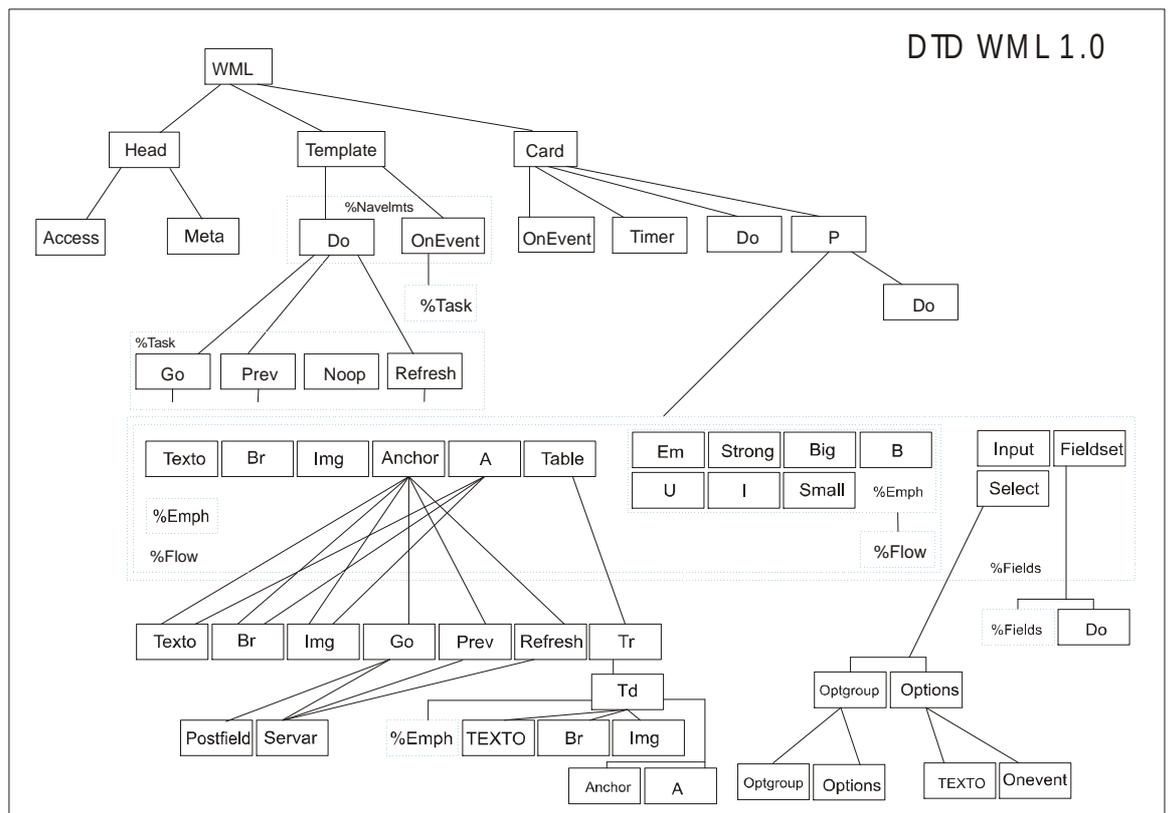


Figura A1- Diagrama do DTD da linguagem WML 1.0

-