

1

Introdução

WML é uma linguagem de programação baseada no XML (eXtensible Markup Language). A especificação oficial do WML foi desenvolvida e é mantida pelo [WAP Forum](http://www.wapforum.org), um consórcio industrial fundado pela [Nokia](http://www.nokia.com), [Phone.com](http://www.phone.com), [Motorola](http://www.motorola.com), e [Ericsson](http://www.ericsson.com). As especificações definem a sintaxe, variáveis e elementos usados no WML. Um atual WML 1.1 está disponível em:

http://www.wapforum.org/DTD/wml_1.1.xml

Neste tutorial são apresentados o WML básico e seus exemplos. Os exemplos demonstrarão eventos e a navegação.

Entendendo

O XML permite definir algumas tags e estas tags devem ser agrupadas conforme mencionado no documento sobre WML:

http://www.wapforum.org/DTD/wml_1.1.xml

Se um telefone ou outro dispositivo de comunicação é tido como WAP, significa que este possui um software, conhecido como microbrowser, e este software tem a capacidade de entender tudo que é especificado como sendo WML 1.1..

A primeira informação dentro de um XML é conhecida como um *prolog*. Quando o *prolog* for opcional, ele consistirá em duas linhas de código: A declaração XML (que define a versão do XML) e a declaração de documentação (um ponteiro indicando o documento DTD do wapforum). Veja um *prolog* abaixo:

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

Depois do *prolog*, cada XML contém um único elemento que contém todos outros sub-elementos e entidades. Como no HTML todos elementos são delimitados pelos caracteres:

<> e </>

Ficaria então: <elemento>Minha Informação</elemento>. Pode haver somente um elemento "documento" por documento. Com o WML, o elemento "documento" é <wml>; Todos outros elementos estarão contidos nele.

Tutorial WML – Páginas para Celular

Neilon Márcio

2

Os dois métodos mais comuns de armazenar dados com o XML são os elementos e atributos. Os elementos são itens estruturados e são representados pelas tags. Os elementos podem também conter sub elementos. Veja um exemplo a seguir:

```
<!--cartão login -->
<card id="LoginCard" title="Login"> seu nome. </card>
O comentário foi colocado entre os caracteres <!-- e -->.
```

O elemento *card* contém os atributos *id* e *title*.

Elementos WML

O WML define elementos que podem ser combinados para que se possa criar documentos WML. Os elementos podem ser divididos em dois grupos: os elementos Deck/Card e os elementos Eventos.

Deck/Card

wml card template head access meta

Eventos

do ontimer onenterforward onenterbackward onpick onevent postfield

Tarefas

go prev refresh noop

Variáveis

setvar

Entradas

input select option optgroup fieldset

Ancoras, Imagens e Timers

a anchor img timer

Formatação de Textos

br p table tr td

Veja as descrições de cada um destes elementos em Tags para WML, cada um destes elementos é utilizado na seguinte forma:

```
<elemento>valor do elemento</elemento>
```

3

Se um elemento não possuir nenhum dado entre ele (o que é freqüente), pode-se utilizar o caractere "/". Por exemplo, no caso do elemento `
`, pode-se usar `
`

Aplicações com WML

O WML foi desenvolvido para uma pequena banda, e displays muitos pequenos. Uma parte deste desenvolvimento é o conceito utilizado como deck/cards. Um único documento WML é conhecido como Deck. Uma única interação com o usuário é conhecida como card. A vantagem do projeto é que múltiplas telas podem ser baixadas para o cliente numa única requisição. Usando WMLScript, as seleções e entradas do usuário podem ser retidas e distribuídas entre os cartões já carregados, eliminando então, transações excessivas com os servidores remotos. Claro que com limitações para o usuário. Dependendo da capacidade de memória do cliente, pode ser necessário dividir vários cartões e vários Decks para evitar que um deck fique demasiadamente grande.

Variáveis

Como vários cartões podem estar contidos em um deck, alguns mecanismos precisam estar posicionados para reter dados, enquanto o usuário navega de cartão em cartão. Este mecanismo é fornecido pelas variáveis WML. As variáveis podem ser criadas e usadas utilizando-se diversos métodos diferentes. Por exemplo:

- Usando o elemento `<setvar>` como resultado do usuário que executa uma tarefa. O elemento `<setvar>` pode ser usado para "setar" uma variável para os seguintes elementos: *go*, *prev* e *refresh*. O elemento abaixo criará uma variável "v1" com um valor "meu":

```
<setvar name="v1" value="meu"/>
```

- As Variáveis também são "setadas" através dos elementos de entrada (input, select, option, etc.). Uma variável é automaticamente criada correspondendo a um atributo de um elemento de entrada. Por exemplo, o elemento abaixo criará uma variável de nome "v1":

4

```
<select name="v1" title="v1 Value:">
```

É bom mencionar que quando o WML e o WMLScript trabalham juntos, eles compartilham as mesmas variáveis.

Criando um Deck WML

Neste exemplo, nós criaremos um Deck WML que nos permite primeiramente selecionar um *username* da lista, entrar com uma senha, então temos nossas seleções repetidas. Basicamente será mostrado a manipulação de entrada, dos eventos e das variáveis num único Deck que utiliza vários cartões (cards).

Fonte1: ([faça o download de fonte1wml.wml](#))

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

<wml>

    <card id="Login" title="Login">
        <do type="accept" label="Senha">
            <go href="#Senha"/>
        </do>
        <p>
            Usuário:
            <select name="nome" title="Nome:">
                <option value="joão">João</option>
                <option value="Sergio">Sergio</option>
                <option value="Maria">Maria</option>
                <option value="Claudio">Claudio</option>
            </select>
        </p>
    </card>
</wml>
```

5

```
        </p>
    </card>

    <card id="Senha" title="Senha:">
        <do type="accept" label="Resultado">
            <go href="#Resultado"/>
        </do>
        <p>
            Senha: <input type="text" name="senha"/>
        </p>
    </card>

    <card id="Resultado" title="Resultado:">
        <p>
            Entrada:<br/>
            Nome: $(nome)<br/>
            Senha: $(senha)<br/>
        </p>
    </card>
</wml>
```

Como você pode ver, o *prolog* deste documento contém a versão do XML. Depois deste, vem o elemento do documento WML, o Deck, que contém tres cartões: Login, Senha e Resultado. Cada um destes cartões é definido usando um elemento `<card>`. Os cartões do Login e da Senha também definem Eventos, eles utilizam o elemento `<do type = "accept">` para definir o evento a ser chamado.

Quando o elemento "accept" é encontrado, ele é mostrado no display do celular (ou outro dispositivo) como uma opção.

Selecionando uma destas opções, o elemento `<go>` será analisado.

Se você está familiarizado com a tag *anchor* `<a>` no HTML, e você conhece as especificações do atributo *href*, então se pode fazer um link no browser para uma âncora selecionada . O elemento WML `<go>` e o atributo *href* trabalham da mesma

6

maneira. Com HTML, para fazer um link para outro cartão no mesmo documento, você simplesmente insere o símbolo # antes do link. Por exemplo, um link para o cartão *Resultado*, é só definir o seguinte elemento:

```
<go href="#Results"/>
```

Este cartão *resultado* emprega variáveis recuperando e exibindo o conteúdo das variáveis *Nome* e *Senha*. Lembre-se que as variáveis são substituídas em um cartão ou deck usando a seguinte sintaxe:

```
$(nome_da_variável)
```

Acessando um Servidor de Scripts

Sem a habilidade de executar transações no servidor, o WML serviria somente para fornecer uma maneira padronizada de exibir um texto ao cliente. Com a habilidade de se conectar dinamicamente a um servidor remoto, aí então se abre, aos dispositivos WAP, a possibilidade transação de mensagens, base de dados das empresas e até mesmo o e-commerce. Os dispositivos WAP interagem com estas origens de dados através da Gateway WAP. Este Gateway deve se conectar a um portador, como: CDMA, GSM2, GPRS. Entretanto, é possível instalar e testar produtos do Gateway conjuntamente com servidores da WEB, tais como o IIS4 (Microsoft) ou o Apache.

Para rodar WML e WMLScript, configure os MIME types para:

```
WML text/vnd.wap.wml wml
```

```
WMLScript text/vnd.wap.wmlscript wmls
```

Feito isso, será criado um exemplo simples que permite que o usuário selecione uma opção e então recupere dados de um servidor baseado na opção escolhida. Neste exemplo, utilizaremos o Active Server Pages (ASP) para o script server-side e, naturalmente nosso servidor deverá suportá-lo. Pode-se usar outros servidores para Servlets Java, JavaScript ou Perl. Fonte2 exibe o código fonte em WML para um novo deck. Basicamente ele contém:

```
<select>
```

Elemento que dá ao usuário algumas opções de recuperação

Tutorial WML – Páginas para Celular

Neilon Márcio

7

<go>

Elemento para a lista de seleções que chama um servidor de scripts com os argumentos apropriados.

Fonte2: ([faça o download de fonte2wml.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="Lazer" title="Selecione">
        <p>
            <select name="Opcao" title="Opcao">
                <option value="Futebol">Futebol</option>
                <option value="Volei">Volei</option>
                <option value="Tenis">Tenis</option>
                <option value="Ciclismo">Ciclismo</option>
            </select>
        </p>
        <do type="accept" label="Execute">
            <go href="fonte3wml.asp" method="post">
                <postfield name="Opcao" value="$(Opcao)"/>
            </go>
        </do>
    </card>

</wml>
```

O servidor de script (mostrado em Fonte3) examina a entrada e produz uma saída WML para ser mostrada no dispositivo WAP.

Fonte3: ([faça o download de fonte3wml.asp](#))

8

```
<%  
Dim conteudo  
  
If Request.Form("Opcao") = "Futebol" Then  
    conteudo = "Você selecionou Futebol"  
Elseif Request.Form("Opcao") = "Volei" Then  
    conteudo = "Você selecionou Volei"  
Elseif Request.Form("Opcao") = "Tenis" Then  
    conteudo = "Você selecionou Tenis"  
Elseif Request.Form("Opcao") = "Ciclismo" Then  
    conteudo = "Você selecionou Ciclismo"  
End If  
  
Response.ContentType = "text/vnd.wap.wml"%>  
  
<?xml version="1.0"?>  
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"  
"http://www.wapforum.org/DTD/wml_1.1.xml">  
<wml>  
<card>  
<p>  
  
<%Response.write(conteudo)%>  
  
</p>  
</card>  
</wml>
```

Algumas coisas devem ser mencionadas para que este exemplo funcione em seu servidor web local. Você deve registrar os tipos apropriados do MIME type no seu servidor web de modo que o conteúdo WML possa ser corretamente exibido. Os dois MIME types que devem ser registrados são:

9

.wml

text/vnd.wap.wml

.wmls text/vnd.wap.wmlscript

Para usar imagens bitmap wireless (WBMP - o formato da imagem suportado para WAP), adicione também:

.wbmp image/vnd.wap.wbmp

Tags WML

<a>
<access>
<anchor>

<big>

<card>
<do>

<fieldset>
<go>
<head>
<i>

<input>
<meta>
<noop>
<onevent>
<optgroup>
<option>
<p>
<postfield>
<prev>
<refresh>
<select>
<setvar>
<small>

<table>
<td>
<template>
<timer>
<tr>
<u>
<wml>

CONCEITOS

<a>

Especifica o texto dentro das tags um hiperlink. O destino de uma ligação é especificado como uma URL: O endereço ou o ID de um outro tag. Os autores devem usar esta tag ao invés da tag anchor.

Esta tag é a forma curta de anchor e é limitado a tarefas que não utilizem variáveis.

<access>

Especifica a informação controle do acesso para todo deck. É um erro utilizar mais de um elemento access em um deck. Se um deck não incluir um elemento access, o controle de acesso é desabilitado. Quando o controle de acesso é desabilitado, os cartões em toda o deck podem acessar este deck.

Os domínios e os atributos path especificam quais são os outros decks que se pode acessar. Quando o browser navega de um deck para outro, executa verificações do controle de acesso para determinar se o deck do destino permite o acesso do deck atual.

Se um deck tiver um domínio ou atributo path, a URL consultada deve combinar os valores dos atributos. A

combinação é feita como se segue: o sufixo do domínio do access é combinado com o prefixo do domínio da URL consultada e o path do access tem seu prefixo combinado a parcela do path da URL consultada.

Para combinar o sufixo do domínio use o elemento inteiro de cada domínio-secundário e combine cada elemento exatamente.

<anchor>

Especifica ao texto dentro das tags um hiperlink. O destino de uma ligação é especificado como uma URL: O endereço ou o ID de um outro tag. Os autores devem usar a tag "a" ao invés de anchor.

A tag "a" é a forma curta de anchor e é limitado a tarefas que não utilizem variáveis.

Indica que o texto dentro das tags deve estar com formato bold (negrito). Os autores devem tentar usar os tags strong em lugar da tag b, i e u.

<big>

Indica que o texto dentro das tags deve estar em um formato grande

**
**

Termina a linha atual e começa uma linha nova. Também é suportado dentro de tabelas.

<card>

Um deck de WML contém uma coleção de cards (cartões). O elemento card delimita elementos texto e permite a entrada de uma grande variedade de dispositivos.

Um card pode conter markup, os campos de entrada e os elementos que indicam a estrutura do card.

A identificação de um card pode ser usada como um fragmento de "anchor".

<do>

Esta tag fornece um mecanismo para que o usuário acesse acima do card atual.

A representação da tag depende do dispositivo. A tag 'do' deve aparecer no card e no deck.

Indica que o texto dentro das tags devem ser tratados com ênfase. Os autores devem tentar usar o "em" e o tag "strong" em lugar das tags "b", "i", "u".

<fieldset>

O elemento de fieldset permite o agrupamento de campos relacionados e texto. Este agrupamento permite otimizar o layout e navegação.

Elementos Fieldset podem aninhar e podem proporcionar para o autor meios para especificar comportamentos para uma grande variedade dispositivos.

<go>

Declara uma tag 'go' indicando a navegação para uma nova URL. Se o nome da URL é um cartão WML ou deck, a execução da tarefa exibirá o conteúdo indicado. Esta tarefa executa uma operação 'push' no histórico do browser

<head>

A tag head contém a informação relacionando o cabeçalho ao todo, inclusive meta-data e tags de controle de acesso.

<i>

Indica que o texto dentro das tags deverá ser exibido em modo itálico. Os autores deveriam tentar usar a tag strong em lugar de tags como b, i e u.

A tag img indica que uma imagem será incluída no fluxo do texto. O layout da imagem é feito dentro do contexto do layout do texto.

<input>

O elemento input especifica a entrada de um objeto. O input do usuário é restrito pelo atributo Format (opcional).

<meta>

O elemento meta contém o meta-information genérico relativo ao deck do WML. É especificado um meta-information com nomes de propriedades e valores. Esta especificação não define nenhuma propriedade, nem define como os browsers têm que interpretar meta-data.

<noop>

A tag noop especifica que nada deveria ser feito - quer dizer, "nenhuma operação". Isto pode ser usado em um cartão para cancelar uma tarefa que foi especificada em um template à nível de deck.

<onevent>

O elemento onevent liga uma tarefa a um evento particular para o elemento imediatamente incluído. Por exemplo, especificando um elemento de onevent dentro de um elemento card, este associa um evento intrínseco que o liga com aquele elemento card. Os browsers ignorarão qualquer elemento onevent que especifica um tipo que não corresponda a um evento intrínseco legal para o elemento imediatamente incluído.

<optgroup>

O elemento optgroup permite ao autor agrupar elementos option relacionados em uma hierarquia. Os browsers podem usar esta hierarquia para facilitar o layout de apresentação em uma grande variedade de dispositivos.

<option>

Este elemento especifica uma única opção de escolha em um elemento selecionado.

<p>

O elemento p estabelece uma linha em branco e parâmetros de alinhamento para um parágrafo. Se o

alinhamento do texto não é especificado, o padrão passa ser o alinhamento à esquerda. Se o modo de linha de texto não é especificado, ficará idêntico ao modo de linha de texto do parágrafo anterior no cartão atual. Parágrafos vazios (um elemento vazio ou um elemento com um insignificante espaço branco) será considerado como insignificante e será ignorado nos browsers. Se o primeiro elemento p em um cartão não especificar uma linha de texto ou modo de alinhamento, o padrão será o do modo inicial do cartão.

Os browsers inserirão uma linha no fluxo de texto entre parágrafos significantes.

<postfield>

O elemento postfield 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.

<prev>

A tag prev indica a navegação para a URL anterior no histórico do browser.

<refresh>

A tag refresh indica uma atualização da tela e contexto do dispositivo como especificado pela tag setvar, por exemplo. Efeitos do lado do usuário (por exemplo, uma mudança na exibição de tela) poderão acontecer durante o processo de atualização.

<setvar>

O elemento setvar 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.

<small>

Indica que o texto dentro das tags deverá ser exibido com uma fonte pequena.

Indica que o texto dentro das tags deverá ser exibido com uma fonte forte. Autores deveriam tentar usar as tags "strong" e "em" em lugar das tags b, i e u.

<table>

Usado junto com as tags tr e td para criar jogos de colunas alinhadas de texto e imagens em um cartão. As tags table determinam a estrutura das colunas. As etiquetas separam conteúdo em colunas, mas não especifica a largura da coluna ou entre as colunas.

O número de colunas deve ser especificado pelo atributo de colunas. Se o número atual de colunas for menor que o valor especificado pelo atributo de colunas, a fila será arrumada efetivamente com colunas vazias. A orientação da tabela depende do idioma. São adicionadas colunas do lado direito de uma fila para arrumar tabelas da esquerda

para direita. São adicionadas colunas do lado esquerdo de uma fila para arrumar tabelas da direita para esquerda.

Se o número atual de colunas for maior que o valor especificado pelo atributo de colunas, serão adicionadas colunas extras da fila na última coluna, tal que a fila contenha o número de colunas exatamente especificadas. Um único espaço será inserido entre duas células que estão sendo adicionadas.

A tabela será feita tão estreita quanto possível, em função de seus conteúdos.

<td>

O elemento td é usado como um delimitador para segurar os dados de uma célula de uma tabela dentro de uma linha da tabela. Células de dados de uma tabela podem estar vazias. Células vazias são significantes, e não devem ser ignoradas. Os browsers tentarão lidar com múltiplas linhas de células de dados que podem ser o resultado de usar imagens ou de linhas quebradas.

<template>

O elemento template declara um modelo para cartões. Eventos de ligação especificados no elemento template (por exemplo, do ou onevent) aplicados a todos os cartões naquele deck, embora um elemento de cartão possa anular o comportamento especificado no elemento template . Em particular:

* O elemento DO especificado em um elemento TEMPLATE pode ser anulado em cartões individuais.

* Eventos de ligações especificados no elemento `TEMPLATE` podem ser anulados pela especificação de um evento que o liga a um elemento `CARD`.

<timer>

O elemento `timer` expõe um meio de inatividade no processo ou tempo de inatividade. O `timer` é inicializado e é "startado" na entrada do cartão e é parado quando o cartão sai. Entrada de cartão é qualquer tarefa ou ação de um usuário que resulte no cartão que é ativado, por exemplo, navegando num cartão. Saída de cartão é definida como a execução de qualquer tarefa. O valor de um `timer` vai tendo um decremento a partir de um valor inicial e ativa a entrega de um evento `ontimer` intrínseco quando alcança o valor zero. Note que aquela resolução de `timer` e a interação do `timer` com a interface do browser de um usuário ou outro dispositivo `time-based` depende da implementação.

É um erro ter mais de um elemento de `timer` num cartão.

O valor de intervalo de um `timer` é especificado em unidades de um décimo (1/10) de um segundo. O autor não deveria esperar uma resolução do `timer` particular e deveria

proporcionar para o usuário outros meios para chamar a tarefa `timer`. Um valor de intervalo zero incapacita o `timer`.

Chamar o `refresh` é considerado uma saída. A tarefa para o `timer`, atualiza o valor inicial do `timer`, e atualiza o browser adequadamente. A conclusão é que o `refresh` é considerado uma entrada para o cartão.

<tr>

O elemento `tr` é usado para delimitar uma *linha* em uma *tabela*. As *linhas* podem estar vazias. Mesmo as *linhas* vazias não devem ser ignoradas

<u>

Indica que o texto delimitado pelas *tags* deverá ser exibido **sublinhado**. Os Autores devem tentar usar o *strong* em lugar de *tags* como *b*, *i* e *u*

<wml>

A tag `wml` define um *deck* e inclui alí, todas as informações e *cards*.

LISTA DE EXERCÍCIOS PARA WML

tag <go> - Navegando entre os cards de um deck

tag <p> - Exibe "Oi Mundo Wireless"

tag <p> e atributo (align) - Alinhando texto

tag <select> e <option> - Uma lista de opções

tag - Ressaltando Texto

tag <table> - Tabela de preços

tag <timer> - Temporizando um card

EXERCÍCIO 1

tag <go> - Navegando entre os cards de um deck

Navegando entre os cards com facilidade. Este script inicia-se uma exibição do cartão "Inicio" e aguarda a chamada do cartão "Segundo" e vice-versa. Desta forma pode-se verificar como é simples a navegação entre os cartões em um mesmo deck.

```
<?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="inicio">
    <do type="accept" label="Segundo">
      <go href="#segundo"/>
    </do>
    <p>Card "inicio"<br/><br/>
    Clique em "Segundo" para acessar o card
    "Segundo".</p>
  </card>
  <card id="segundo">
    <do type="accept" label="Inicio">
      <go href="#inicio"/>
    </do>
    <p>Card "Segundo"<br/><br/>
    Clique em Inicio para voltar ao primeiro
    Cartao "Inicio".</p>
  </card>
</wml>
```


EXERCÍCIO 2

tag <p> - Exibe "Oi Mundo Wireless"

Exibindo a frase "Oi Mundo Wireless" em seu celular, utilizando apenas um cartão (card) e o elemento <p>, num script muito simples.

```
<?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="mundo" title="Oi Mundo!">
    <p>
      Oi Mundo Wireless!
    </p>
  </card>
</wml>
```

EXERCÍCIO 3

tag <p> e atributo (align) - Alinhando texto

Alinhamento do texto utilizando o atributo (align), veja como é simples utilizá-lo no exemplo abaixo.

```
<?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="alinhamento">
    <p align="left"> Esquerda
  </p>
    <p align="center"> Centro
  </p>
    <p align="right"> Direita
  </p>
  </card>
</wml>
```

EXERCÍCIO 4

tag <select> e <option> - Uma lista de opções

Utilizando os elementos <select> e <option> pode-se criar uma lista de opções. Após selecionada a opção desejada, será exibido ao visitante o cartão correspondente e seu conteúdo.

```
<?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="bank" title="bank">
    <p mode="nowrap">
      <select>
        <option onpick="#b1">UFIR
        </option>
        <option onpick="#b2">Dolar
        </option>
        <option onpick="#b3">DowJones
        </option>
      </select>
    </p>
  </card>
  <card id="b1">
    <p mode="nowrap">
      Sua Escolha foi: <br/><br/>
      UFIR<br/>
    </p>
  </card>
  <card id="b2">
    <p mode="nowrap">
      Sua Escolha foi: <br/><br/>
      Dolar<br/>
    </p>
  </card>
  <card id="b3">
    <p mode="nowrap">
      Sua Escolha foi: <br/><br/>
      DowJones<br/>
    </p>
  </card>
</wml>
```

EXERCÍCIO 5

tag - Ressaltando Texto

Através do elemento (strong) a palavra "OK" foi ressaltada do resto do texto. O ideal é que se priorize, quando possível, o uso do elemento (strong) para ressaltar texto, deixando os elementos (b), (i), (u) para casos específicos.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <card>
    <p>Click em
    <strong> OK </strong>
    para continuar.
    </p>
  </card>
</wml>
```

EXERCÍCIO 6

tag <table> - Tabela de preços

Utilizando as tags (table), (td) e (tr) coloca-se informações em tabelas para que possam ser exibidas em grupos e alinhadas.

```
<?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="Feira" title="Tabela de precos">
  <p>
  <table columns="2" align="center">
    <tr>
      <td>Suco</td>
      <td>Valor</td>
    </tr>
    <tr>
      <td>Laranja</td>
      <td>2,20</td>
    </tr>
  </table>
  </p>
</card>
```

```

<tr>
  <td>Caja</td>
  <td>3,30</td>
</tr>
<tr>
  <td>Morango</td>
  <td>4,40</td>
</tr>
<tr>
  <td>Ameixa</td>
  <td>2,10</td>
</tr>
<tr>
  <td>Abacaxi</td>
  <td>1,90</td>
</tr>
<tr>
  <td>Cereja</td>
  <td>3,00</td>
</tr>
</table>
</p>
</card>

</wml>

```

EXERCÍCIO 7

tag <timer> - Temporizando um card

Utilizando o elemento de tempo <timer>, o card2 somente será exibido após 7 segundos do início da navegação.

```

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <card ontimer="#card2">
    <timer name="time" value="70"/>
    <p> Aguarde... Este card mudará em
      5 segundos!
    </p>
  </card>
  <card id="card2">

```

```
                <p> Ok! Card2 após 5 segundo está  
                sendo exibido.  
            </p>  
        </card>  
</wml>
```