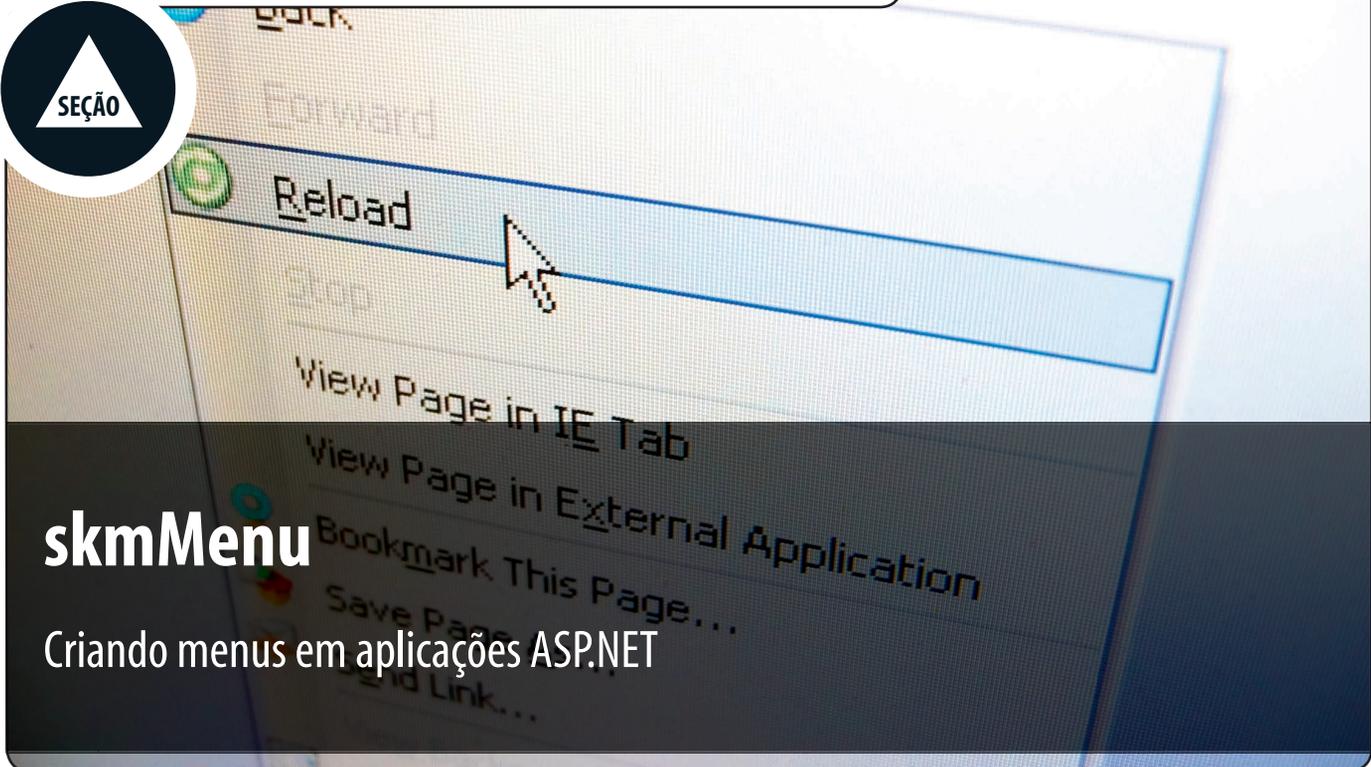


Artigos intermediários sobre Delphi Win32 e Delphi .NET


 SEÇÃO



skmMenu

Criando menus em aplicações ASP.NET

Utilizar menus é uma forma muito prática e eficiente de organizar informações e opções dentro de uma aplicação. Neste artigo, ensinaremos como criar menus facilmente em aplicações ASP.NET com o Delphi 2006 através de um componente chamado *skmMenu*.

Abra o Delphi 2006 e crie um novo projeto através do menu *File>New>ASP.NET Web Application - Delphi for .NET*. O nome do projeto será "Menus" e utilizaremos o IIS como servidor Web. Criado o projeto, proceda então com a instalação do *skmMenu*.

O componente

O *skmMenu* é um componente *open source* utilizado para a criação dos menus em aplicações ASP.NET. Primeiramente, baixe o componente, através do site do projeto (www.skmmenu.com), usando o menu *Getting Started>Download*. Baixe a versão *Binary*, que já vem compilada. Até a data de criação deste artigo, a última

versão era a 2.2. Para fazer o download, clique em *skmMenu Version 2.2 Binaries*. Salve o arquivo em algum local no seu computador e descompacte-o.

Feito isso, proceda agora com a instalação do componente no Delphi e sua adição na aplicação de exemplo. Para isso, no Delphi, vá até o menu *Components>Installed .NET Components* e na aba *.NET Components* clique em *Select an Assembly*. Na caixa que se abrirá, vá até o diretório onde o arquivo foi salvo e selecione o *skmMenu.dll*.

Agora, encontre o componente na lista e assegure-se que a caixa de seleção a sua frente está marcada, como na **Figura 1**. Caso essa caixa de seleção estiver desmarcada, o componente não aparecerá na paleta de componentes.

Feito isso, clique em OK. Nesse momento, na *Tool Palette*, haverá um componente chamado *Menu*, como pode-se observar na **Figura 2**. Esse é o componente que acabou de ser instalado e que será utilizado



Luís Fernando Quitaiski

(lfernando@gmail.com)

é Técnico em Informática formado pelo Colégio Frederico Jorge Logemann de Horizontina/RS. Desenvolvedor de aplicações Windows e ASP.NET com Delphi 2006 e Firebird.

para a criação dos menus. Clique no *Menu* e arraste-o para o formulário.

Principais propriedades

Veja agora as principais propriedades do componente, que visam definir, entre outras coisas, a sua aparência e o comportamento:

- *Layout*: define o *layout* horizontal ou vertical;
- *Font*: define o estilo da fonte que será utilizada nos menus, como nome da fonte, negrito, itálico e outros efeitos;
- *ForeColor*: define a cor da letra dos menus;
- *BackColor*: define a cor do fundo dos menus;
- *EnableViewState*: define se o estado do componente deve ser guardado entre uma requisição e outra;
- *ClickToOpen*: Define se, para abrir um menu, será necessário clicar sobre ele (*True*) ou somente passar com o mouse (*False*);
- *ID*: Define o nome do componente.

Destaca-se ainda a propriedade *SelectedItemStyle*, onde podemos definir uma aparência diferente para o item que o usuário está apontando com o mouse no momento, diferenciando-o dos outros itens do menu. Isso cria um efeito muito interessante, semelhante ao produzido pelos menus utilizados em aplicações para Windows. Para este exemplo, configure as propriedades do componente da seguinte maneira:

- *BackColor*: "#E0E0E0";
- *Font>Name*: Arial;
- *EnableViewState*: *False* (já que não será necessário guardar o estado do componente);
- *Layout*: *Horizontal*;
- *SelectedItemStyle>BackColor*: "#FFC080".

Para efeitos didáticos, adicione ainda um *RadioButtonList* e altere a propriedade *ID* para "rbTipodeUsuario", *AutoPostBack* para *True* e *EnableViewState* para *False*. Acesse a propriedade *Items* e aparecerá a caixa que permite que sejam adicionados itens ao *RadioButtonList*.

Clique em *Add* para adicionar um item e altere sua propriedade *Selected*, *Text* e *Value* para, respectivamente, *True*, "Administrador" e "Administrador". Adicione outro item clicando no *Add* e defina suas

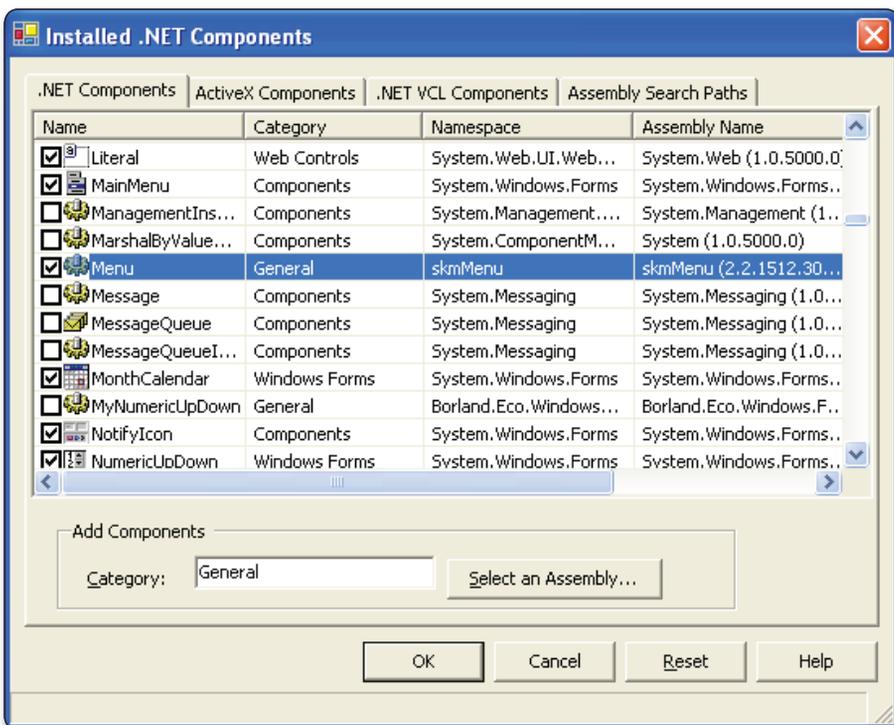


Figura 1. Instalação do skmMenu

propriedades para, respectivamente, *False*, "Usuário" e "Usuario". Adicione também um *Label* e altere sua propriedade *Text* para "1". Seguindo todos esses passos, a aparência da página deverá ser parecida com a Figura 3.

Adicione mais uma página ao projeto, clicando no menu *File>New>Other* e na caixa que aparecer, dentro da categoria *Delphi for .NET Projects*, acesse *New ASP.NET Files*. Selecione *New ASP.NET Page* e dê um OK.

Nessa página adicione o *Menu* e configure as suas propriedades para os mesmos valores que foram escolhidos anteriormente (você pode copiar e colar o componente). Dê um ENTER após o componente para ir para a próxima linha da página e digite "Página 2".

Criando a estrutura dos menus

Configurada a aparência do *Menu*, será criado agora a sua estrutura, ou seja, os seus itens e submenus. Essas informações são armazenadas em um arquivo XML guardado dentro do diretório da aplicação.

Como um arquivo XML é composto apenas por texto, para criá-lo podemos utilizar qualquer programa que trabalhe com textos simples, como o próprio Bloco



Figura 2. Componente instalado na Tool Palette

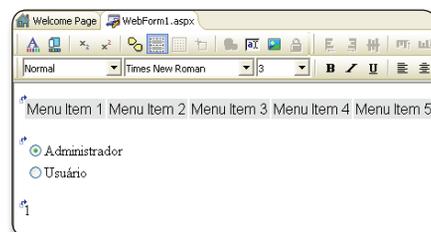
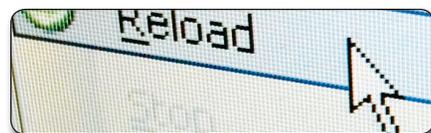


Figura 3. Aparência da aplicação após inseridos os componentes



de Notas do Windows. Neste artigo, será utilizado o Notepad++ (um Bloco de Notas “turbinado”, disponível em *notepad-plus.sourceforge.net*) para a criação desse arquivo (Figura 4).

Abra, então, um editor de texto simples e salve, no diretório de sua aplicação (geralmente em *C:\Inetpub\wwwroot\Menus*), um arquivo chamado “menu.xml”. Lembre-se que um arquivo XML é *case sensitive*, ou seja, há diferenças entre uma palavra escrita com letras maiúsculas e com letras minúsculas. Por exemplo: *text* é diferente de *Text*.

Em um arquivo XML, a primeira linha é uma identificação, que guarda a versão do XML que foi utilizada (nesse caso, a 1.0) e a codificação utilizada pelos caracteres do texto (a codificação escolhida afeta como são interpretados os caracteres que possuem sinais gráficos, como os acentos e o cedilha, e ela depende das configurações do servidor).

Recomenda-se que sejam testados os valores *iso-8859-1* e *iso-8859-2* e que seja escolhido o que se comportar corretamente. A linha será a seguinte: `<?xml version="1.0" encoding="iso-8859-1"?>`. Logo após a linha de identificação, tem-se uma tag que deverá englobar todos os itens do menu. Essa tag chama-se *menu* e é aberta logo após a identificação do arquivo e é fechada na última linha do mesmo.

Para criar um novo menu, utilizamos a tag *menuItem*. Criada a tag *menuItem*, vem a parte mais interessante: a criação dos textos do menu e a configuração da ação que será executada quando o usuário clicar sobre aquele item. Toda essa configuração também é feita por tags, e as principais estão listadas a seguir:

- *text*: texto que aparecerá no menu para o usuário;
- *image*: permite que o menu mostre, ao invés de texto, a imagem especificada;
- *tooltip*: dica referente ao menu, que será mostrada ao usuário se ele mantiver o mouse sobre aquele menu por algum tempo;
- *url*: página para onde o usuário será direcionado ao clicar no menu selecionado. Pode ser uma página da própria aplicação ou ser uma página da internet;
- *javascriptcommand*: comando *JavaScript* que será executado quando o usuário

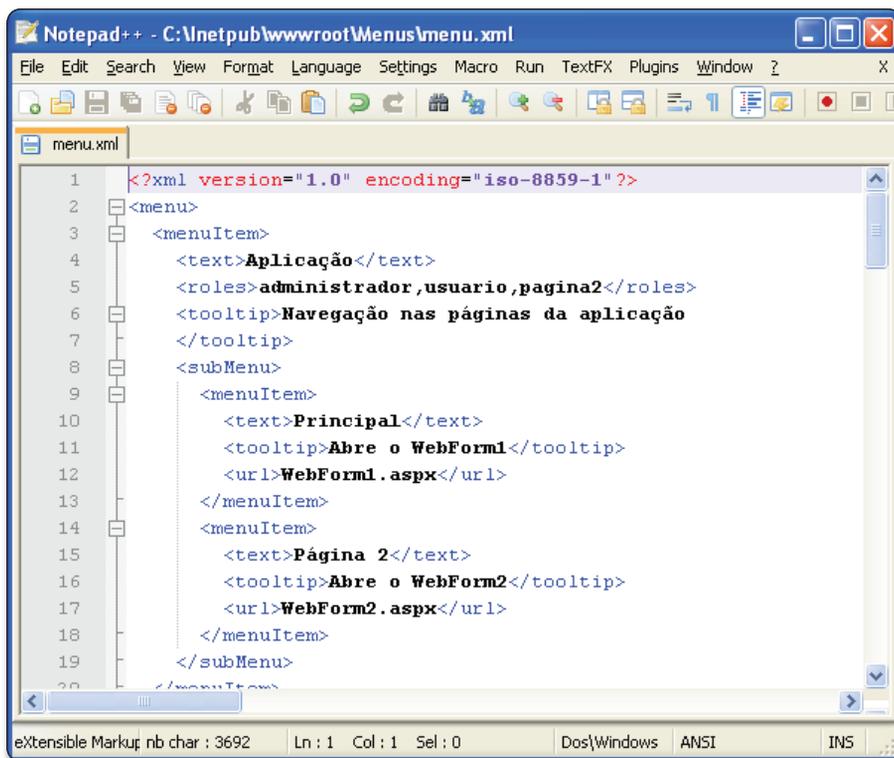


Figura 4. Exemplo de arquivo XML

clicar no menu selecionado;

- *commandname*: utilizado se quisermos que, ao clicar no menu, seja executada uma ação codificada no sistema. As outras tags citadas, como *url* ou *javascriptcommand* são executadas pelo browser do cliente, como ir para outra página ou executar um *JavaScript*. O conteúdo dessa tag será passado à aplicação, onde o programador poderá executar a ação desejada utilizando-a como um identificador;

- *leftimage*: adiciona a figura especificada à esquerda do texto;
- *rightimage*: adiciona a figura especificada à direita do texto;
- *roles*: os usuários terão acesso àquele menu, separados por vírgula e sem espaço entre eles.

Nota: Se não forem especificadas as permissões de acesso para um submenu, elas serão herdadas do menu imediatamente superior.

Outra tag muito importante é a *subMenu*, que permite, como o nome sugere, que os menus possuam submenus, ou seja, que um menu surja quando o usuário passar com o mouse sobre outro

menu. Seu uso é muito simples: abre-se a tag *subMenu*, cria-se quantos menus quisermos utilizando a tag *menuItem* e depois fechamos a tag *submenu*.

Para a criação de espaçadores existe a tag *menuSpacer*, onde podemos utilizar, dentro dela, as tags *spacermarkup* e *height*, utilizadas para, respectivamente, definir o caractere que aparecerá no espaçamento e definir o tamanho vertical da célula que conterá o caractere especificado pela tag *spacermarkup*. Na **Listagem 1** temos o XML utilizado nos menus do nosso exemplo.

Codificações necessárias

Até esse momento, os menus já estão configurados e sua estrutura já está criada. Prossiga, então, colocando o código da **Listagem 2** no *Page_Load* do *WebForm1*.

Ele é responsável por passar o arquivo *menu.xml* para o componente que, baseado nas informações desse arquivo, criará os menus na tela. Ele também é responsável por passar ao componente, através do *UserRoles.Add*, os usuários que terão acesso àquele menu. Lembrando que esse valor pode ser obtido de qualquer lugar, como de uma tabela no banco de dados ou de uma variável de sessão, por exemplo.

Listagem 1. Conteúdo do arquivo menu.xml

```

<?xml version="1.0" encoding="iso-8859-1"?>
<menu>
  <menuItem>
    <text>Aplicação</text>
    <roles>administrador,usuario,pagina2</roles>
    <tooltip>Navegação nas páginas da aplicação
    </tooltip>

    <submenu>
      <menuItem>
        <text>Principal</text>
        <tooltip>Abre o WebForm1</tooltip>
        <url>WebForm1.aspx</url>
      </menuItem>

      <menuItem>
        <text>Página 2</text>
        <tooltip>Abre o WebForm2</tooltip>
        <url>WebForm2.aspx</url>
      </menuItem>
    </submenu>
  </menuItem>

  <menuItem>
    <text>Páginas</text>
    <roles>administrador,usuario,pagina2</roles>
    <tooltip>Todos conseguirão ver esse menu</tooltip>

    <submenu>
      <menuItem>
        <text>DevMedia</text>
        <tooltip>Abre a página da Editora DevMedia
        </tooltip>
        <url>http://www.devmedia.com.br</url>
      </menuItem>

      <menuItem>
        <text>Google</text>
        <tooltip>Abre a página do Google</tooltip>
        <url>http://www.google.com.br</url>
      </menuItem>
    </submenu>
  </menuItem>

  <menuItem>
    <text>Email</text>
    <roles>administrador,pagina2</roles>
    <tooltip>Somente administradores verão esse menu
    </tooltip>

    <submenu>
      <menuItem>
        <text>Hotmail</text>
        <tooltip>Abre a página de login do Hotmail
        </tooltip>
        <url>http://www.hotmail.com</url>
      </menuItem>

      <menuItem>
        <text>GMail</text>
        <tooltip>Abre a página de login do Gmail
        </tooltip>
        <url>http://www.gmail.com</url>
      </menuItem>
    </submenu>
  </menuItem>

  <menuItem>
    <text>Outros</text>
    <roles>administrador,usuario</roles>
    <tooltip>Mostrar como utilizar comandos JavaScript
    e comandos que serão executados pela aplicação
    </tooltip>

    <submenu>
      <menuItem>
        <text>Exemplos JavaScript</text>
        <tooltip>Exemplos de como utilizar comandos
        JavaScript em um menu</tooltip>

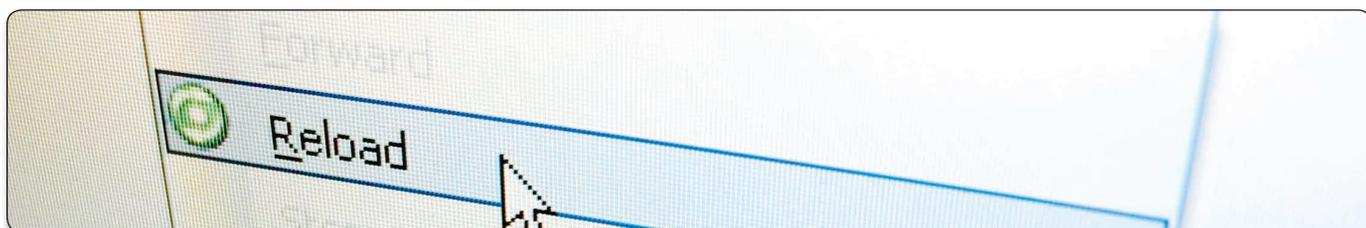
        <submenu>
          <menuItem>
            <text>Caixa de Mensagem</text>
            <tooltip>Mostra uma caixa de mensagens na
            tela utilizando JavaScript</tooltip>
            <javascriptcommand>javascript:alert (
            'Você clicou no menu Caixa de Mensagem');
            </javascriptcommand>
          </menuItem>

          <menuItem>
            <text>Janela Pop-Up</text>
            <tooltip>Abre, em uma janela pop-up,
            a página do Google</tooltip>
            <javascriptcommand>javascript:window.open (
            'http://www.google.com.br', '',
            'width=640px, height=480px');
            </javascriptcommand>
          </menuItem>
        </submenu>
      </menuItem>

      <menuItem>
        <text>Chamar comando na Aplicação</text>
        <tooltip>Exemplos de como executar comandos
        na aplicação através do menu</tooltip>
        <submenu>
          <menuItem>
            <text>Adicionar 1</text>
            <tooltip>Adiciona 1 unidade ao número que
            está aparecendo na tela</tooltip>
            <commandname>AdicionarUm</commandname>
          </menuItem>

          <menuItem>
            <text>Alterar Cor</text>
            <tooltip>Alterar a cor do número que está
            aparecendo na tela</tooltip>
            <commandname>AlterarCor</commandname>
          </menuItem>
        </submenu>
      </menuItem>
    </submenu>
  </menuItem>
</menu>

```



Coloque agora, no `MenuItemClick` do `Menu1` o código presente na **Listagem 3**. Esse evento é chamado quando o usuário clica em um item do menu que possui a tag `commandname`. Através da identificação passada pelo `commandname`, podemos executar comandos no sistema.

Coloque o código a seguir no `Page_Load` do `WebForm2`. Sua função também é passar o arquivo `menu.xml` como `DataSource` para o componente:

```
{ Somente mostrar os menus que possuam a role
"pagina2" }
Menu1.UserRoles.Add ('pagina2');
Menu1.DataSource:= Server.MapPath ('menu.
xml');
Menu1.DataBind;
```

Compile a aplicação e veja o menu em ação. Ao clicar em um dos itens do `RadioButton` pode-se ver a tag `roles` em funcionamento, permitindo que somente os usuários declarados na tag `roles` tenham acesso àquele menu.

No menu `Aplicação` temos disponíveis as páginas do nosso projeto, onde podemos ir para a página principal ou para a página 2. Em `Outros`, há exemplos do menu executando comandos `JavaScript` (mostrando uma caixa de mensagens e abrindo, em uma janela `popup`, a página do Google), e, também de como fazer para que um clique no menu resulte em uma ação da aplicação (no exemplo, incrementar em 1 unidade o número que aparece na tela ou alterar a sua cor de preto para vermelho ou vice-versa). A aplicação compilada deve ficar semelhante à **Figura 5**.

Conclusão

Vimos neste artigo que a criação de menus em aplicações ASP.NET através do `smkMenu` é muito simples. Tudo o que temos a fazer é configurar a sua aparência e criar um arquivo XML com a estrutura dos menus, tarefas relativamente fáceis de serem feitas e que geram um efeito visual muito legal, além de ser uma ótima maneira de organizar os recursos oferecidos pela sua aplicação. ●

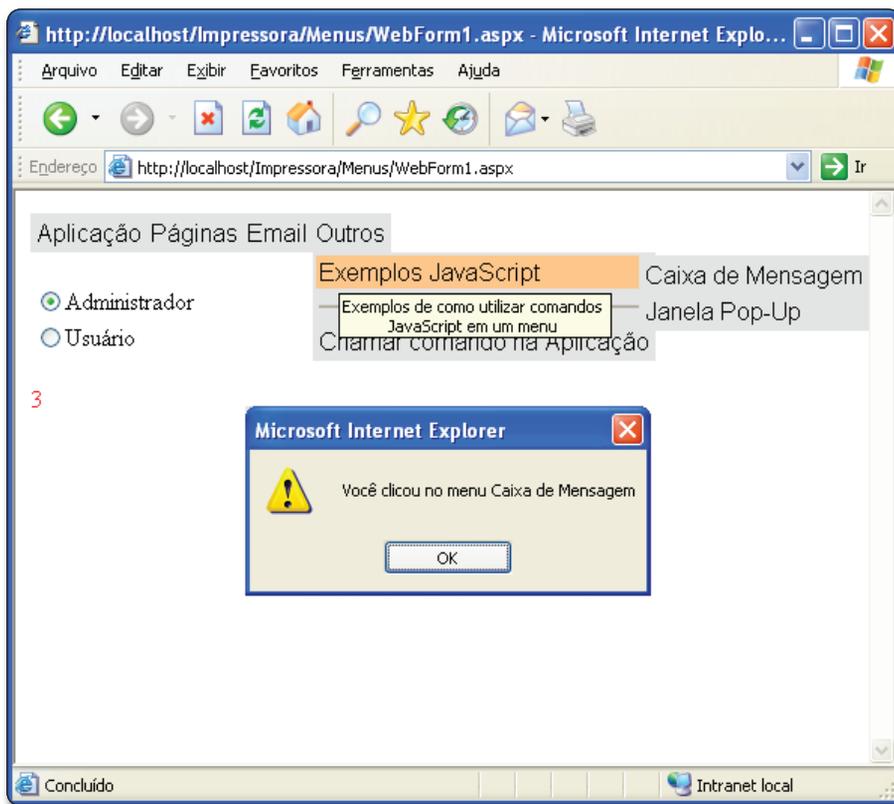


Figura 5. Aplicação em funcionamento

Listagem 2. Page_Load do WebForm1

```
{ Adiciona os usuários que terão permissão de
acesso à esse menu }
Menu1.UserRoles.Add (rbTipoDeUsuario.SelectedVale);
{ Passa o arquivo XML como DataSource para o menu e
preenchê-lo }
Menu1.DataSource := Server.MapPath ('menu.xml');
Menu1.DataBind;
```

Listagem 3. Evento MenuItemClick do Menu

```
{ Testa o menu clicado }
if e.CommandName= 'AdicionarUm' then
{ Comando executado se o item selecionado for
AdicionarUm, então adiciona 1 unidade }
Label1.Text:= Convert.ToString (Convert.ToInt16 (
Label1.Text) + 1)
else
{ Comando executado se o item selecionado for
AlterarCor }
begin
{ Verifica a cor atual e altera entre vermelho e
preto }
if Label1.ForeColor= Color.Red then
Label1.ForeColor:= Color.Black
else
Label1.ForeColor:= Color.Red;
end;
```

