

Nesta seção você encontra artigos intermediários sobre Delphi Win 32 e Delphi for .NET

Controle Wizard no .net 2.0

Criando assistentes passo a passo com o novo controle Wizard

Nesse artigo veremos

- Controle Wizard;
- Criando Wizard em aplicações ASP.NET;
- Configurando WizardSteps.

Qual a finalidade

Conhecer o controle Wizard para criação de assistentes passo a passo em Web Sites;

Quais situações utilizam esses recursos?

Com o controle Wizard é possível criar assistentes passo a passo para as mais diversas tarefas on-line, como cadastros, por exemplo.

É muito comum encontrarmos interfaces de *wizard* em aplicações *Web*, ou seja, aplicações que orientam o usuário a realizar determinadas tarefas passo a passo. Os famosos *Next*, *Next*, *Finish*. Assim como na instalação de um software, temos vários passos a seguir, com a possibilidade de avançar ou retornar ao passo anterior.

Tradicionalmente, poderíamos criar uma interface *wizard* utilizando diversas páginas, onde cada uma é um passo do processo. Mas eis que na versão 2.0 do ASP.NET temos um controle específico para desenvolvimentos de interfaces *wizard*, facilitando em muito o trabalho de criar esse tipo de solução. Neste artigo veremos como utilizar e configurar o controle *Wizard* do ASP.NET 2.0.



Adriano Santos

falecom@adrianosantos.pro.br

É desenvolvedor Delphi desde 1998, professor e programador PHP, bacharel em Comunicação Social pela Universidade Cruzeiro do Sul, SP. É Editor Chefe das revistas ClubeDelphi e Web-Mobile. Gerente de TI na SoftPark, parceira Borland, membro fundador do DUG-SP (www.dug-sp.com) e mantenedor do blog Delphi to Delphi (www.delphitodelphi.blogspot.com) com dicas, informações e tudo sobre desenvolvimento Delphi.



Resumo do DevMan

Todos nós conhecemos o famoso Next, Next, Finish em instaladores de programas e até mesmo de sistemas operacionais. Mas como fazer isso na internet e por quê? O uso mais comum de assistentes na internet acontece em cadastros e alterações de preferências de sistema, como é o caso do Orkut.

Nesse artigo, faremos uso do controles Wizard do .net 2.0 com o RAD Studio 2007. Veremos como utilizá-lo na prática para criar uma pequena tela de cadastro de usuário.

Incluindo o controle Wizard na página

Vamos direto a um exemplo prático. Abra o seu *RAD Studio 2007* e crie um site usando o menu *File>New>ASP.NET Web Application – Delphi for .NET* e salve o projeto com o nome *Wizards*. Veja que na *Tool Palette* em *Web Controls*, temos um controle chamado *Wizard* (**Figura 1**).

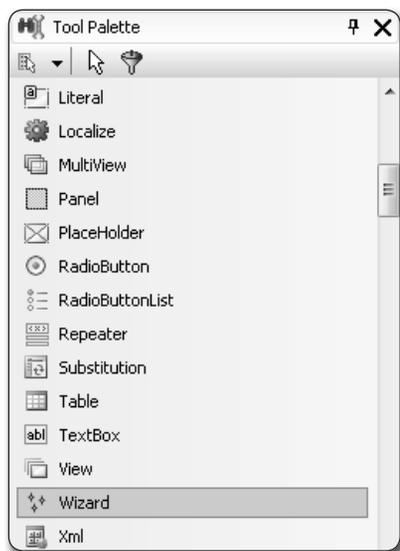


Figura 1. Controle Wizard na Tool Palette

Arraste um controle deste tipo para a página *Default.aspx* do projeto que acabamos de criar. Veja que ao incluir o controle na página, temos acesso ao *Wizard Tasks* (**Figura 2**).

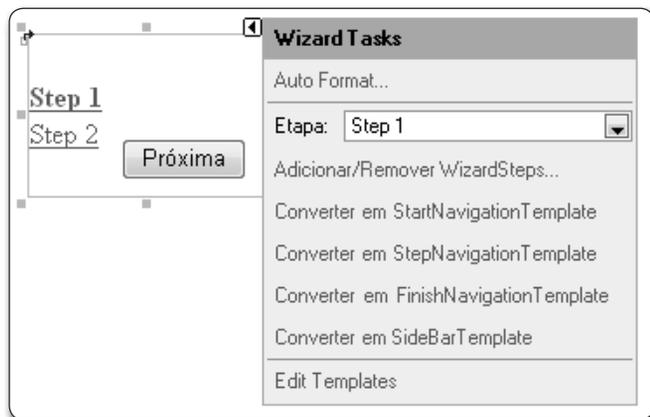


Figura 2. Formatando layout do controle Wizard

Nele, a primeira opção é o *Auto Format*. Clique no *Auto Format* e troque a formatação para *Professional*.

Definindo os passos do Wizard

Um *wizard* pode ter inúmeras aplicações, o preenchimento de informações cadastrais para efetuar uma compra, o processo de configuração de um sistema, etc. Neste exemplo vamos criar um *wizard* que irá capturar as informações necessárias para cadastro em um site de relacionamento (tipo o “Orkut”).

A primeira configuração que devemos fazer em um controle

Wizard, é definir quais serão os passos (*steps*) da tarefa a ser executada. Para isso, nas *Tasks* do controle, clique na opção *Add/Remove WizardSteps* e será aberto um editor.

Esse editor é responsável pela manutenção dos *WizardSteps*, que representa um passo do *wizard*. Utilizando o botão *Remove*, exclua todos os *WizardSteps* que aparecem. Clique no botão *Add* para incluir o nosso primeiro *WizardStep*. Ele se chamará “Bem vindo!”, que será uma introdução de boas-vindas do *wizard*.

Para configurarmos o nome do *WizardStep* é só modificar a propriedade *Title*, que fica à direita. Inclua mais três *Steps*, que se chamarão respectivamente: *Dados Pessoais*, *Dados Profissionais* e *Conclusão* (**Figura 3**). Feche o editor para voltar ao controle *Wizard*.

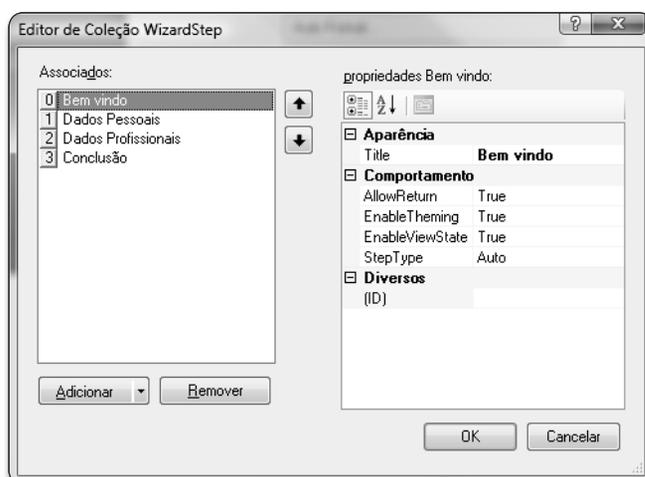


Figura 3. Configurando os passos do Wizard



Nota do DevMan

Você deve ter observado que cada *Step* possui um ou mais botões. Através das *Tasks* do *Wizard*, utilizando a opção *Step*, você pode modificar qual *WizardStep* está aparecendo em nosso design. Veja que o primeiro *Step* possui o botão *Next*, o último possui os botões *Finish* e *Previous*. Os demais *Steps* possuem os botões *Next* e *Previous*. Temos a possibilidade de modificar os textos desses botões para trocarmos pelas respectivas palavras em português: *Próximo*, *Anterior* e *Concluir*. Para isso, vá nas propriedades do *Wizard* e modifique o valor das propriedades listadas a seguir:

- *FinishCompleteButtonText*: Concluir;
- *FinishPreviousButtonText*: Anterior;
- *StartNextButtonText*: Próximo;
- *StepPreviousButtonText*: Anterior;

Além de trocar o *Text* dos botões, através das propriedades, podemos realizar diversas customizações, incluindo configurações de estilos, formatações de layout, imagens para os botões, etc.

Incluindo texto no Bem Vindo

Cada *Step* de um *Wizard* funciona como um *Container*, onde podemos incluir controles e desenharmos assim cada passo do processo. No primeiro *Step* vamos apenas incluir um texto de boas-vindas.

Para isso, selecione o *Bem Vindo* através do Wizard *Tasks* clique na área vazia logo acima do botão *Próximo*. Inclua e formate um texto de Boas-Vindas nesse espaço. Veja na **Figura 4** como o primeiro *Step* do controle *Wizard* ficará.



Figura 4. Incluindo texto no StepWizard

Criando um formulário no Wizard

Vamos agora criar um formulário para inclusão dos dados pessoais do usuário. Selecione o passo *Dados Pessoais* e clique na área vazia do *Step*. O cursor irá aparecer dentro do *Step*, vá então a barra de ferramentas superior do Delphi e clique no botão *Insert Table*. Inclua uma tabela com 2 colunas e 5 linhas. Veja na **Figura 5** com deve ficar o formulário para preenchimento das informações pessoais do usuário.



Figura 5. Criando o formulário de dados pessoais

Veja que utilizamos vários *TextBoxes* e dois *RadioButtons* para o campo *Sexo*. O importante aqui é que qualquer controle pode ser utilizado dentro de um controle *Wizard*. Devemos encarar cada *Step* como uma página comum, onde desenvolvemos nossos formulários.

Também foi incluído um controle *RequiredFieldValidator* para validar o preenchimento do nome do usuário. Nas propriedades do *RequiredFieldValidator*, em *ControlToValidate* informamos o ID do *Text-Box* referente ao nome do usuário. E na propriedade

ErrorMessage incluímos o texto: "O Nome é obrigatório".

Para o passo *Dados Profissionais* vamos fazer a mesma coisa, crie uma tabela e inclua os controles necessários para construir um formulário, semelhante ao mostrado na **Figura 6**.



Figura 6. Criando um formulário de dados profissionais

Finalizando o Wizard

Em nosso *Step* de conclusão, vamos apenas incluir um texto de finalização. O importante aqui é que ação tomar quando o usuário clicar em *Concluir*. Nas propriedades do controle *Wizard*, acesse os

PENSE...

QUANTO TEMPO
VOCÊ GASTARIA
PARA DESENVOLVER
COBRANÇA COM BOLETOS
BANCÁRIOS PARA
APENAS UM BANCO
NO SEU SOFTWARE

COBREBEMX

- 56 BANCOS E MAIS DE 430 CARTEIRAS DE COBRANÇA PARA IMPRESSÃO E/OU ENVIO DE BOLETO BANCÁRIO POR EMAIL;
- GERAÇÃO DE BOLETOS ON LINE;
- GERAÇÃO E LEITURA DE ARQUIVOS (REMESSA/RETORNO) NOS PADRÕES FEBRABAN E CNAB;
- MAIS DE 40 EXEMPLOS EM DIVERSAS LINGUAGENS DE PROGRAMAÇÃO

Tecnologia

DOWNLOADS E INFORMAÇÕES EM WWW.COBREBEM.COM

eventos do mesmo. Veja que podemos codificar diversos eventos possíveis, um deles é o *FinishButtonClick*. Dê um duplo clique no *FinishButtonClick* e veja que seremos redirecionados ao evento, pronto para ser codificado.

Aqui podemos tomar a ação que desejarmos com os dados. Se a nossa aplicação fosse real, o correto seria persistir os dados preenchidos em um banco de dados. Para efeito de testes vamos apenas mostrar uma mensagem com o nome do usuário. Veja na **Listagem 1** a codificação necessária para dispararmos uma mensagem de finalização ao usuário. O que deve ficar claro, é que nesse evento, podemos dar o destino que quisermos aos dados preenchidos.

Listagem 1. Mensagem de conclusão do Wizard

```
01 procedure TDefault.Wizard1_FinishButtonClick1(sender: System.
02   Object; e:
03   System.Web.UI.WebControls.WizardNavigationEventArgs);
04 var
05   Texto : string;
06 begin
07   Texto := 'Olá ' + TextBox(Wizard1.FindControl('txtNome')).Text +
08     ' seu cadastro foi efetuado com sucesso!';
09   Response.Write(
10     '<script language='\"'\"'javascript'\"'\"'>window.alert('\"'\"' +
11     Texto + '\"'\"') + '</script>');
12 end;
```

Testando a Aplicação

Deixe o *Welcome* selecionado para que seja o primeiro *Step* a aparecer para o usuário. Salve, compile e execute o projeto. Na tela *Welcome* clique no botão *Próximo*. Veja que o *Wizard* passa para o *Step* seguinte. Preencha as informações da tela *Dados Pessoais*.

Tente passar para o próximo *Step* sem preencher o campo Nome. Veja que o controle *Wizard* só permite que você passe ao próximo passo se o controle de validação não acusar nenhuma inconsistência. Veja também que é possível navegar pelo *Wizard* através dos links que temos na barra lateral esquerda e não apenas pelos botões.

Dica: Para ocultar os links no painel esquerdo e permitir a navegação somente pelos botões, configure para *False* a propriedade *DisplaySideBar*.

Clique em *Próximo*, preencha os *Dados Profissionais* e clique novamente no botão *Próximo*. No final, clique no botão *Concluir* e veja que a mensagem de conclusão do controle *Wizard* será mostrada. Veja um exemplo na **Figura 7**, lembrando que aqui poderíamos persistir os dados informados em um banco de dados e redirecionar o usuário para uma outra página do *site*.

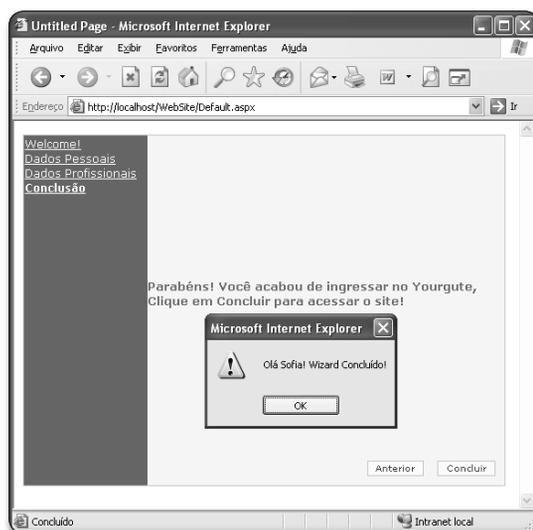


Figura 7. Concluindo Wizard



Nota do DevMan

Um *Workflow*, ou simplesmente fluxo de trabalho é, segundo o Wikipédia, a seqüência de passos necessários para que se possa atingir a automação de processos de um negócio, de acordo com um conjunto de regras definidas, envolvendo a noção de processos, permitindo que estes possam ser transmitidos de uma pessoa para outra de acordo com algumas regras. Um exemplo de *Workflow* está no processo de compra online. Desde o ponto onde você confirma sua compra até você receber o produto, existe uma série de passos que devem ser realizados, como solicitação do produto ao setor de estoque e este na seqüência envia o produto para o setor de empacotamento, que por sua vez envia ao setor de distribuição onde será então entregue à uma transportadora e finalmente você recebe o produto.

Conclusão

Novamente pudemos conferir como o *ASP.NET 2.0* facilita o trabalho do desenvolvedor. Criamos uma estrutura de *wizard* com praticamente nenhum código. O único código utilizado aqui foi para dispararmos uma mensagem ao final.

Podemos criar soluções mais avançadas de *wizard*, como por exemplo, modificar os *Setps* em tempo de execução. Imagine uma solução de *Workflow*, onde cada etapa do processo dependa de informações inseridas pelo usuário. Explorando as propriedades e eventos do controle, podemos aplicar o conceito de *wizard* da maneira mais apropriada aos nossos requisitos. ●

Dê seu feedback sobre esta edição!

A Clubedelphi tem que ser feita ao seu gosto. Para isso, precisamos saber o que você, leitor, acha da revista!

Dê seu voto sobre este artigo, através do link:

www.devmedia.com.br/clubedelphi/feedback

