**Tutorial de Utilização do Componente UserControl**

**Parte1**

**APRESENTAÇÃO**

http://www.planetadelphi.com.br/artigo/132/usercontrol---controle-os-usu%C3%A1rios-na-aplica%C3%A7%C3%A3o---parte-1/  
  
**UserControl** - Controle de Usuários: Uma milagre para sua aplicação, uma grande benefício para você  
  
**1 - Introdução**  
    Muitas vezes perdemos um tempo imenso projetando regras de acesso de usuários em nossos aplicativos. Além de procurar arranjar uma forma dos usuários acessarem o aplicativo, devemos estabelecer uma forma de como armazenar essas informações em um banco de dados, ou no registro do Windows. Para acabar com esse problema, surge o conjunto de componentes chamado UserControl. Nas linhas que se seguirão nesse artigo, estarei tentando mostrar em detalhes, o uso desse maravilhosos conjunto de componentes.  
  
**2 - Download e Instalação**  
    O download dos arquivos pode ser feito no endereço [www.usercontrol.net](http://www.usercontrol.net/) . No site, você deverá clicar no link Baixar (o link está presente no final da página principal). Ou [clique aqui](http://www.usercontrol.net/index.php?option=com_remository&Itemid=27&func=fileinfo&id=12) para acessar o link rapidamente. Após executar o download do arquivo, descompacte-o no diretório de sua preferência. O processo de instalação é simples. Você não conta com um instalador, mas pode contar com um arquivo de ajuda muito bem feito por  **Eduardo de Freitas Nunes**. [Pressione aqui](http://rapidshare.com/files/20834434/instala__o_User_Control.zip) para saltar para o link onde o tutorial de instalação poderá ser baixado. **Eu recomendo que você baixe esse tutorial**. Os passos para instalação apresentados abaixo, está de forma muito simplificada, mas funciona.  
  
**2.1 - Passos para Instalação**  
a) Execute o Delphi e clique no item de menu **Tools**; avance até o item **Enviroment Options**. Quando a janela se abrir, clique na aba **Library**. Na opção **Library Path**, clique no botão **Add** e adicione os caminhos das pastas: **Source** e **Package** dentro da pasta principal UserControl2.20RC2.  
OBS: Como recomendação, também adicione a pasta Indy que está localizada dentro da pasta **Delphi\Source\Indy**.  
b) Executados os detalhes no item anterior, vamos executar a instalação propriamente dita dos componentes. Você deverá instalar um pacote contendo componentes comuns, e deverá ao menos instalar um dos conectores disponíveis na pasta Connectors, ou instalar todos se assim achar necessário. Dentro da pasta UserControl, abra a pasta Packages e abra os arquivos \*.dpk (pckMD5.dpk, pckUCDataConnector.dpk,pckUserControl\_DT.dpk e pckUserControl\_RT.dpk) Compile primeiro o pckMD5.dpk, seguindo dopckUCDataConnector. Após compile o arquivo pckUserControl\_DT. Aqui poderá existir o problema de versão do conjunto de componentes Indy. É recomendável o uso da versão 10, porém se você não tiver, não tem problema. Tente compilar, se algum problema existir, retire da seção REQUIRES do arquivo pckUserControl\_RT.dpk as declarações do Indy e compile novamente; os pacotes corretos serão adicionados automaticamente. Depois de tudo compilado clique sobre o arquivo pckUserControl\_DT e pressione o botão **Install**.   
c) Finalizada a instalação dos componentes, você deverá instalar os conectores. Esses conectores é que ligam a sua base de dados ao componente UserControl. Abra a pasta que contém o conector que você usa e instale-o - para isso basta selecionar o arquivo \*.dpk e compilá-lo. Você poderá instalar todos os conectores ou somente aqueles que você mais usa.  
  
**3 - Primeiros passos - Criando a aplicação de Exemplo**  
    Após ter executado o processo de instalação do seu novo conjunto de componentes, você terá a sua disposição, duas abas na paleta de componentes do Delphi: **UC Main** e **UC Connectors**. Na paleta UC Main, existem 7 componentes, sendo eles respectivamente: UserControl, UCSettings, UCControls, UCXPStyle, UCAppMessage, UCIDle e MailUserControl). Na aba UC Connectors, você terá somente aqueles conectores que foram instalados.  
  
    Para que possamos entender o funcionamento desse maravilhoso conjunto de componentes, vamos criar uma pequena aplicação.  
**3.1 - Criando um Banco de Dados**

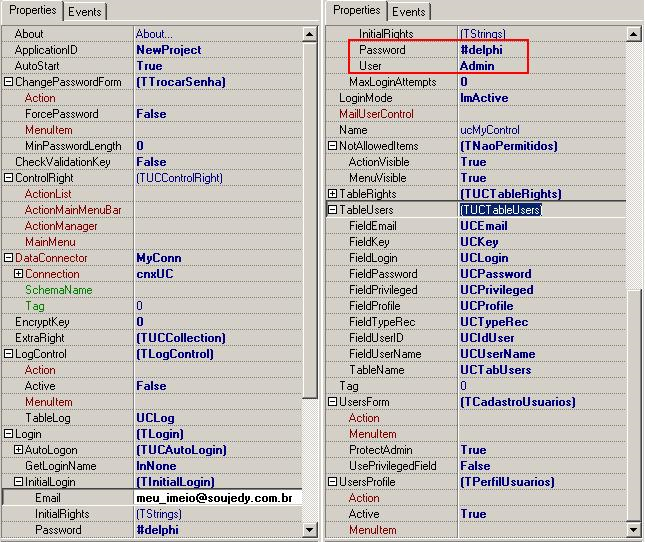
Você poderá criar o banco de dados usando o programa que preferir. Eu uso a versão do IBExpert Free. Esse procedimento eu não irei explicar. Mas você pode criar um banco de dados contendo a seguinte estrutura:  
  
[Clique aqui para baixar o arquivo contendo a estrutura do banco de dados](http://www.delphi.eti.br/colunistas/bitzero_000/CriandoaBaseDeDados.rar)  
  
peço desculpas a vocês, mas quando criei o arquivo esqueci-me de adcionar essa parte:  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
/\*\*\*\*                             Primary Keys                             \*\*\*\*/  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

ALTER TABLE BANCOS ADD CONSTRAINT PK\_BANCOS PRIMARY KEY (CODIGO\_B);  
ALTER TABLE CIDADES ADD CONSTRAINT PK\_CIDADES PRIMARY KEY (CODIGO);  
ALTER TABLE CLIENTES ADD CONSTRAINT PK\_CLIENTES PRIMARY KEY (CODIGO);  
ALTER TABLE PRODUTOS ADD CONSTRAINT PK\_PRODUTOS PRIMARY KEY (CODIGO\_P);  
  
**3.2 - Sistema de Exemplo** Para não perder muito tempo, [clique aqui](http://www.delphi.eti.br/colunistas/bitzero_000/Aplicat_UserControl.rar) para baixar o arquivo rar contendo os arquivos de banco de dados e também de aplicativo. Nesse pequeno aplicativo, foram adicionados um data module contendo um componente SQLConnection da paleta DbExpress com o Name cnxUC; um componente TUCDBXConn da paleta UCConnectors com a propriedade Name definida para MyConn e um componente UserControl da paleta UC Main, tendo a sua propriedade Name definida para ucMyControl.  
  
    Execute as configurações para o componente SQLConnection (isso não faz parte do escopo desse tutorial). Na propriedade Connection do componente TUCDBXConn, defina o SQLConnection (cnxUC). Clique no componente UserControl e na propriedade DataConnector defina o TUCDBXConn (MyConn).  
Executando esses procedimentos, você somente executou a parte mais simples de tudo. Agora, vamos começar a entender algumas propriedades do componente UserControl.  
  
**4 - Entendendo e Definindo as propriedade do Componente UserControl da Aplicação**

    Quando você adiciona um componente UserControl em sua aplicação, ele cria em sua base de dados algumas tabelas que armazenarão diversos dados - mensagens, logins, contas de usuários e direitos adicionais por exemplo. Isso tudo é criado automaticamente. Mas como isso é criado? É criado no primeiro momento em que o componente é executado. Ele criará uma conta de Administrador e uma Senha que poderão ser definidas por você, ou poderá ser o próprio Usuário e a Senha definidas como padrão no componente. Você primeiro se responsabilizará por efetuar a conexão normal com seu banco de dados, depois você pede que o UserControl entre em ação.  
  
Abaixo, estarão listadas as propriedades principais do componente UserControl e o que cada uma faz.

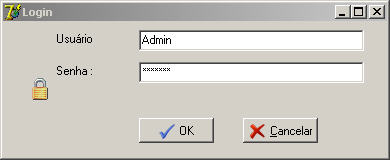
**About**: apresenta uma caixa contendo os agradecimentos e os créditos de desenvolvimento;  
  
**ApplicationID**: armazena o nome do projeto. Pode ser o mesmo nome do executável ou um nome definido pelo usuário;  
  
**AutoStart**: booleana. Quando True faz a conexão automática, e cria a estrutura inicial do UserControl;  
  
**DataConnector**: é o nome do conector. Um dos conectores disponívies na paleta UC Connectors.  
 **TableRights**: Onde você definirá os nomes dos campos e o nome da tabela de Direitos do Usuário.  
 **TableUsers**: Você definirá os nomes dos campos e o nome da tabela que armazenará os dados dos usuários que acessam o sistema.  
 **EncryptKey**: Quando as informações são registradas no banco de dados, elas sofrem uma encriptação. Caso ocorra algum acesso ao banco de dados por um usuário não-autorizado, ele não saberá quais são as senhas do usuários ou quais são os direitos de acesso. Nessa propriedade, você define um valor de 0 até 65535 complexa.  
 **LogControl**: contém diversas sub-propriedades. A propriedade **Active** quando definida para True, armazena o registro de Log do usuário na tabela de log, definida na propriedade **TableLog**. A sub-propriedade **InitialLogin** armazena o nome do usuário privilegiado (administrador), a senha e o endereço de e-mail. A propriedade **InitialRights** contém os direitos do administrador quando acessa o sistema.  
  
**Num momento mais apropriado, outros propriedades serão apresentadas.**

Veja nessa [figura](http://www.delphi.eti.br/colunistas/bitzero_000/propriedadesUC.JPG) quase todas as propriedades que foram expandidas do componente UserControl.



Voltando a nossa aplicação...  
**ATENÇÃO: Deixe a propriedade AutoStart definida para TRUE.**

no evento OnCreate do DataModule, adicione o comando de conexão:

**procedure** TdmUC.DataModuleCreate(Sender: TObject);  
**begin**  
   cnxUC.Connected := True;  
**end**;  
  
    Execute a aplicação, na primeia "rodada", o usercontrol lhe fornecerá uma caixa informando a senha de usuário privilegiado - aquela definida anteriormente em **InitialLogin**. Veja que ele também informa que você deverá fornecer as permissões para esse usuário. Pressione o botão Ok e a caixa de login aparecerá...  
  
Mensagem inicial  
  
  
  
Primeiro login  
  
  
  
    Digite a senha e usuário e pressione Ok. Num primeiro momento parece que nada aconteceu. Mas no fundo e de forma muito transparente, o UserControl criou todas as tabelas principais. Nada aconteceu no menu que criamos, porque não linkamos algumas propriedades no UserControl.  
   
**5 - Fazendo acontecer**

    No dataModule, clique sobre o componente UserControl (ucMyControl) e clique sobre a propriedade **UsersProfile**, no item **MenuItem** selecione **Form1.CadastrodePerfil1**. Clique sobre a propriedade **UsersForm** e no item **MenuItem** defina **Form1.CadastrodeUsurios1**. Na propriedade **ControlRight**, clique no item **MainMenu** e selecione o Menu principal do Form1 (Fo**rm1.MainMenu1**).

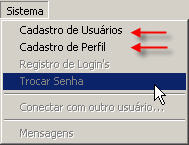
    Execute a aplicação, (provavelmente a janela informando a senha do usuário privilegiado seja apresentada novamente. Se isso acontecer, não se preocupe. Nas próximas conexões isso não ocorrerá novamente, até porque todos os itens do menu já estarão devidamente linkados ao componente UserControl) e no item de menu Sistema, clique sobre a opção: **Cadastro de Usuários**.

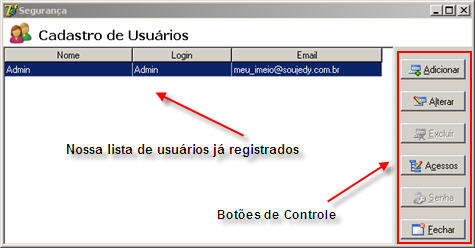
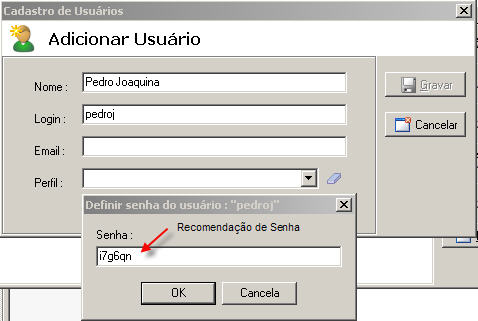
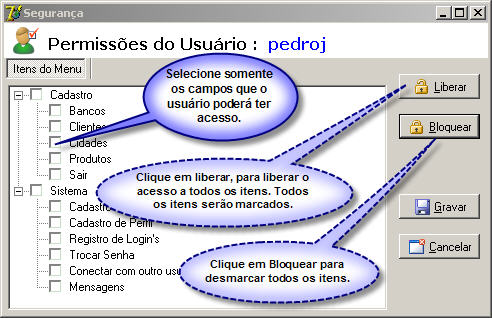
Por enquanto é só isso. Já no próximo tutorial, estaremos chamando as janelas de perfil de usuário, log de usuários, senha de usuários e janelas de mensagens internas.

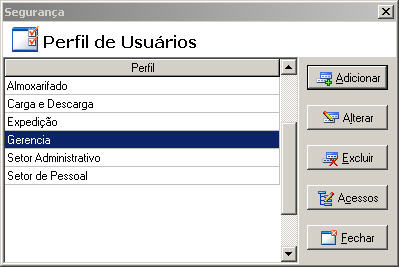
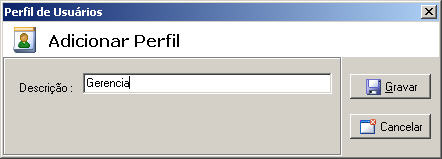
Abraços a todos e espero que gostem desse primeiro tutorial do UserControl. Não esqueçam de visitar o site e também contribuir com o desenvolvimento desses componentes.

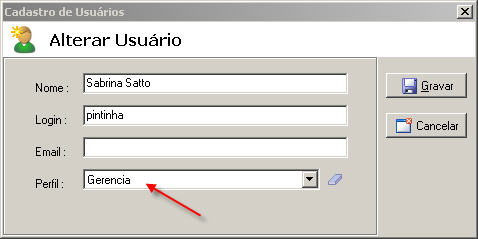
**Parte2**

http://www.planetadelphi.com.br/artigo/133/usercontrol---controlando-usu%C3%A1rios-e-definindo-o-perfil---parte-2/

No primeiro tutorial, nós estabelecemos as conexões com os componentes da paleta UserControl com o SQLConnection. Vimos também, que não tinhamos estabelecido as configurações no primeiro login e o que acontecia - tínhamos o acesso, mas não tinhamos direitos. Na mesma seqüência, preenchemos corretamente as propriedades do UserControl, para que o login pudesse ser executado e que tivéssemos acesso ao item de menu Sistema > Cadastro de Usuário.  
  
**PARA CONTINUAR O TUTORIAL, VOCÊ DEVERÁ EXCLUIR AS TABELAS DO USER CONTROL - DEIXE AS TABELAS BANCOS, CIDADES, CLIENTES E PRODUTOS SOMENTE. ISSO É PARA QUE POSSAMOS FICAR TODOS DENTRO DE UM PADRÃO DURANTE O TUTORIAL**.  
  
    Como deixei claro, você deve excluir as tabelas criadas pelo componente UserControl de nosso banco de dados. Conecte-se novamente, todo o procedimento explicando o nome de usuário e senha serão executados novamente. Isso porque estamos recriando as tabelas. Depois do login, podemos ver que temos os itens: Cadastro de Perfil e Usuários no menu Sistema de nosso exemplo, hablitados.  
  
  
   
Clique sobre o item Cadastro de Usuários.

**1 - Janela de Cadastro de Usuários**  
    Quando essa janela é ativada, podemos executar as tarefas de cadastros de usuários. Podemos alterar, registrar, excluir e disponibilizar os acessos através de seleção dos itens de menu, ou também, criar um perfil de usuário e atribuir esse perfil diretamente.  
  
  
  
**2 - Definindo Níveis de Acessos**  
    Para definir um nível de acesso para um usuário, primeiro faça o cadastro pressionando o botão **Adicionar**. Preencha os campos conforme solicitado. Quando pressionar o botão **Gravar**, o sistema solicitará que você entre com uma senha, mas o próprio sistema também lhe dará uma indicação. Esse recurso é muito interessante, porque você não precisa ficar meditando em projetar uma senha para o usuário. Mude a senha indicada pelo sistema ou entre com uma nova e pressione o botão **Ok**.  
  
  
  
    Finalizado o cadastro de usuário, clique sobre o nome do usuário que você quer dar o acesso e em seguida, pressione o botão **Acessos**. A **janela Permissões** é apresentada. Isso mesmo, você pressiona o botão Acessos e a janela Permissões será aberta. Não está errado. As permissões são dadas por acesso ao sistema. Você pode fazer qualquer tipo de configuração, bastando para isso, clicar sobre os itens de menu que o usuário poderá acessar. Isso se dá de uma forma individual. Você ainda conta com os botões **Liberar** e **Bloquear**. Quando o botão **Liberar** é pressionado, todos os itens existentes no menu são disponibilizados para o usuário. Se o botão **Bloquear** for pressionado, todos os itens do menu serão desmarcados. Feita a configuração ideal para o usuário, pressione o botão **Gravar**.  
  


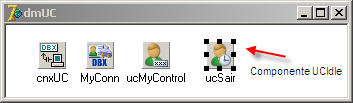
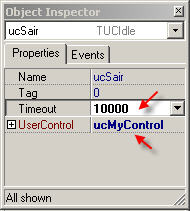
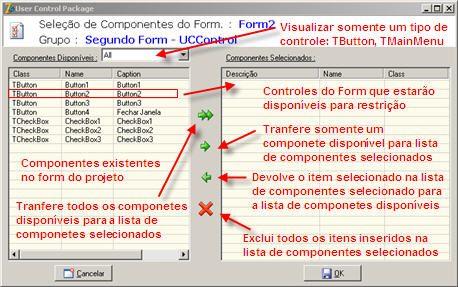
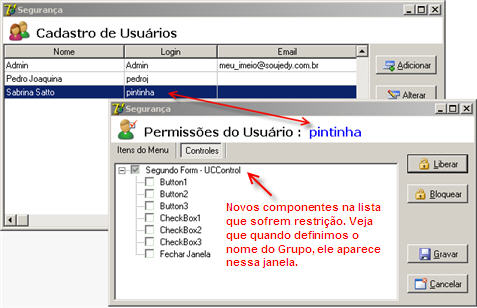
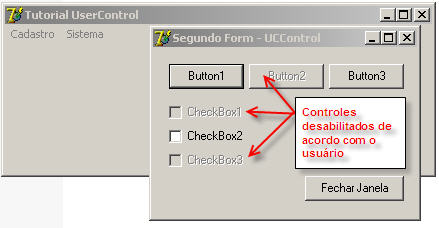
**3 - Perfil de Usuários**   
  
    O que é o recurso do Perfil? Vamos supor que você tenha três usuários no departamento de pessoal, geralmente esses setores acabam crescendo e não só o ambiente, mas também o material humano. Através do sistema de perfil, você poderá definir um padrão e gravar esse padrão. Quando um novo usuário entrar no setor, você só precisará selecionar um perfil para ele e pronto. Esse recurso de configuração através de perfil é realmente excepcional, ele facilita bastante a vida dos administradores do sistema. Para definir perfis de usuário, clique no menu Sistema no nosso exemplo e em seguida,  sobre o item **Cadastro de Perfil**. A janela é apresentada.  
  
  
  
    Clique sobre o botão **Adicionar**, e a janela Adicionar Perfil aparece. Escreva o nome do perfil e pressione o botão **Gravar**.  
  


Nessa mesma janela, pressione sobre o botão **Acessos**. A janela de Permissões do Perfil torna-se disponível para você definir o padrão de acesso. Marque os itens de menu desejados e pressione o botão **Gravar**.  
  
  
  
**4 - Registrando o usuário dentro de um perfil**  
    Feche a janela de perfil e clique novamente no item de menu Sistema de nosso exemplo. Clique sobre o item **Cadastro de Usuários**. Clique sobre um dos nomes de usuários da lista e em seguida, pressione sobre o botão **Alterar**. Clique no **ComboBox Perfil** e selecione um dos itens registrados e definidos anteriormente.  
  


Pressione novamente o botão **Gravar**. Se quiser adicionar direitos mais personalizados para um usuário em específico, mesmo que ele tenha recebido um perfil, clique novamente no botão Acessos na janela de Cadastro de Usuários. Até aqui, você já tem uma idéia do poder dos componentes UserControl, e isso porque só estamos vendo por enquanto o UserControl. Aguardem que vem muito mais por ai.  
Abraços – Exio

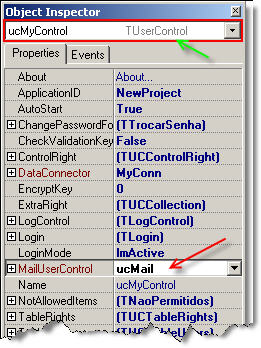
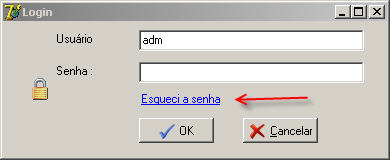
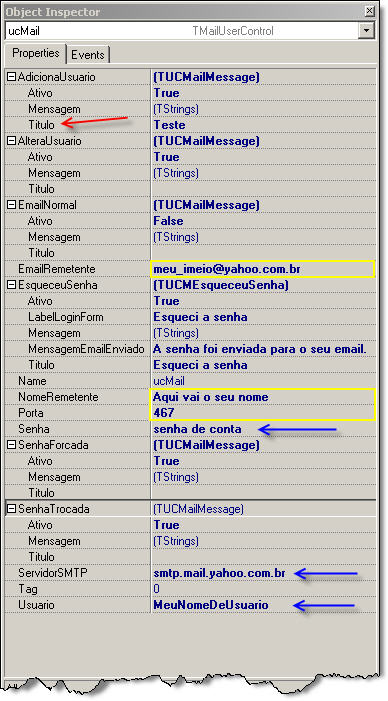
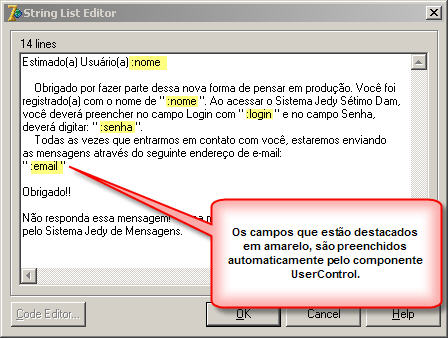
**Parte3**

**Apresentação**

http://www.planetadelphi.com.br/artigo/134/usercontrol---componentes---ucidle-e-uccontrol---parte-3/  
  
[**Baixe o código fonte completo do exemplo clicando aqui.**](http://www.delphi.eti.br/colunistas/bitzero_000/UserControl.rar)  
  
    Nessa parte de nosso tutorial, veremos dois novos componentes: **UCIdle e UCControl**,   naturalmente, os dois componentes estão presentes na paleta **UCMain**. O que faz cada um? O **UCIdle** controla o tempo que a aplicação fica inativa e o segundo componente, o **UCControl** é usado para definir quais controles um usuário poderá acessar num form que pode ou não ser o principal.  
  
**1 - UCIdle**  
    Adicione um componente UCIdle no datamodule. Uma coisa interessante é que esse componente não exige muitas configurações, ele é muito parecido com um componente Timer. No Object Inspector, defina a propriedade **UserControl** do componente UCIdle para **ucMyControl;** e a propriedade **TimeOut** defina um valor qualquer - eu estou definindo como exemplo, o valor 15000. O tempo é contado da mesma forma quando adicionamos um componente Timer na aplicação - para cada mil, um segundo. O que acontece quando adicionamos esse componente em nossa aplicação? Ele ficará contando o período de inatividade da aplicação, caso o tempo decorra sem qualquer interferência do usuário, a janela de login é chamada novamente.  
    Um detalhe importante deve ser apresentado, esse **componente não** **desativa a conexão** com o banco de dados, simplesmente, "desloga" o usuário, exigindo um novo login na aplicação. Esse é um daqueles componentes com bastante relevância. Ambientes com muitos usuários, exigem uma responsabilidade no registro de dados das aplicações. Ao saírem para almoço ou um lanche e ficarem logados, qualquer pessoa pode alterar dados sem o responsável saber, esse componente acaba com o problema de manter os usuários durante o perído integral de trabalho, logados no sistema.  
    Compile e execute o nosso pequeno sistema; depois de se logar, não toque mais no mouse. Aguarde a passagem do tempo e veja o que acontece.  
  
  
  
UCIdle - Inserido no DataModule de nosso sistema  
  
  
  
Propriedades do componente UCIdle - Simples e Rápido  
  
    Não há muito o que se falar do UCIdle, ele simplesmente tem uma função bem definida. Verifica se o usuário está usando a aplicação, se não estiver, desloga o usuário. É interessante ressaltar também que, você pode encerrar a conexão com o banco de dados no evento onLogOff do componente UserControl. Se a sua apicação fica realizando testes de conexão e de mensagens, isso contribuirá enormemente para a diminuição do tráfego.  
  
**2 - UCControl**  
  
   O UCControl é um componente mais especializado, a sua principal função é criar direitos de acessos em form's. Quando adicionado em um form, é permitido ao desenvolvedor, selecionar componentes que poderão ter o acesso controlado. Um exemplo simples e claro seria: um botão de imprimir poderia estar habilitado para um usuário x e não habilitado para o usuário y.  
  
**2.1 - Exemplo de controle de objetos em um form de nossa aplicação**  
    Adicione um novo Form em nosso exemplo. Adicione quatro botões, três checkbox e um componente UCControl. Salve o form. Deixe parecido com os da figura abaixo:  
  
  
  
   Vamos informar que o **form2** deverá usar o **datamodule** (pressione SHIFT + F11). Não esqueça de fazer a chamada ao **Form2** através do item de **menu principal**. Escolha qualquer um dos itens existente no menu Cadastro. Eu fiz o link com a opção **Bancos**. Clique sobre o componente UCControl e no Object Inspector definas as propriedades como:  
  
**GroupName** : Segundo Form - UCControl  
**UserControl** : dmUC.ucMyControl  
  
    A propriedade **NotAllowed** quando definida para **naInvisible**, faz com que os componetes que o usuário não tem direito de acesso fiquem invisíveis. Se definida para **naDisable**, os controles ficam desabilitados.  
  
  
      
    Dê dois cliques sobre o componentes UCControl, ou se preferir, poderá clicar na propriedade Components e em seguida, pressionar o botão de retiscências para que a janela de seleção de componentes seja aberta.  
  
http://www.delphi.eti.br/colunistas/bitzero_000/BotaoComponentes.jpg  
  
   A janela de Seleção de Componentes do Form torna-se disponível - Disponível somente em tempo de projeto. Nessa janela, iremos selecionar os componentes que sofrerão restrição quando a aplicação estiver sendo executada.  
  
  
  
  Voltando ao nosso exemplo. Selecione todos os componentes ou aqueles que desejar e transfira-os para a lista de componentes selecionados. Para confirmar as modificações pressione o botão Ok. Feito isso, parece que nada ocorreu de espetacular em nosso pequeno sistema. Porém, você terá uma surpresa. Compile nosso aplicativo e execute-o. Em tempo de execução, acesse a janela de Cadastro de Usuarios. Selecione um usuário e pressione no botão Acessos. Na janela de Permissões de Usuário, você terá além da opção Itens de Menu a opção Controles. Defina os direitos do usuário de acordo com a sua vontade. Em seguida, faça o login com o usuário para o qual foram definidas as novas restrições de acesso.  
  
  
  
  
Definindo o acesso para a usuária "pintinha"  
  
  
  
  
Acesso ao Segundo Form com o login da usuária "pintinha". UCControl teve a propriedade NotAllowed definida para **naDisabled**.   
  
   Aguardem um pouco mais que já estamos finalizando esse passeio pelo mundo dos controles de usuários.

**Parte4**

http://www.planetadelphi.com.br/artigo/135/usercontrol----esqueci-a-senha-%28enviando-e-mail%27s-para-o-usu%C3%A1rio%29-ultima-parte/

**Esqueci a Senha - Enviando E-Mail pelo UserControl - MailUserControl (UCMail)**  
**1 - Apresentação**  
    Estimados amigos desenvolvedores, chegamos a última parte de nosso tutorial sobre o UserControl. Para encerrar este tema, apresento o componente de mensagem. Vamos dar uma pequena passeada pelo código-fonte do componente, e destacar uma função importante que nos ajudará a montar o corpo do e-mail através de parâmetros. Então vamos começar a fazer o nosso trabalho.  
  
**2 - É importante analisar o Código-fonte?**  
    Qualquer componente que você tenha acesso ao código-fonte é sempre uma base de informações bem interessante. Muitos desenvolvedores aperfeiçoam as suas técnicas lendo e aprendendo, em seguida, acabam implementando ou personalizando esses componentes. o UserControl é uma suite de código-fonte aberto e com uma altíssima qualidade de programação. Se você abrir o código do **UCMail**, verá que não é muito extenso e que na verdade, é uma mistura de componentes Indy (SMTP) com algumas propriedades que são enviadas diretamente pelo componente UCMail. Abra o arquivo **UCMail.pas** e veja esse código:  
  
**function** **TMailUserControl.ParseMailMSG**(Nome, Login, Senha, Email, Perfil, txt : **String**): **String**;  
**begin**  
  Txt    := StringReplace(txt, ':nome', nome, [rfReplaceAll]);  
  Txt    := StringReplace(txt, ':login', login, [rfReplaceAll]);  
  Txt    := StringReplace(txt, ':senha', senha, [rfReplaceAll]);  
  Txt    := StringReplace(txt, ':email', email, [rfReplaceAll]);  
  Txt    := StringReplace(txt, ':perfil', perfil, [rfReplaceAll]);  
  Result := Txt;  
**end**;  
  
    Esse código é declarado como privado e é usado pelo próprio componente, para montar o corpo de texto dos e-mail's que serão enviados para o usuário.  
  
**3 - Começando o trabalho de configuração para o envio de e-mail**  
    Adicione ao Datamodule um componente **TMailUserControl(UCMail)** presente na paleta **UC Main**. Todos os componentes que adicionamos anteriormente faziam conexão com o UserControl, agora a situação se inverte; o UserControl e quem se conecta com o MailUserControl. Na propriedade **Name** do MailUserControl entre com **ucMail**. Clique sobre o componente **ucMyControl** e na propriedade **MailUserControl** preencha com **ucMail** conforme apresentado na figura abaixo.  
  
       
  
    Quando esta conexão dos dois componente é estabelecida, na janela de login do UserControl há um link que se torna visível e funcional depois que o usuário entra com o seu nome. Quando não existe a conexão, o label "Esqueci a Senha" não será apresentado aos usuários do seu sistema.  
  
  
  
**4 - Configurações básicas do MailUserControl**  
    Expanda todas as propriedades presentes no ucMail que adicionamos no datamodule. Os seguintes eventos ocorrem com:  
  
**Propriedade AdicionaUsuario**  
    Quando a propriedade Ativo estiver definida para TRUE, um e-mail é enviado para o usuário quando ele é registrado no sistema através através da janela de cadastro de usuários - visto no primeiro tutorial. A propriedade Titulo é usada para definir o Assunto.  
  
**Propriedade AlteraUsuario**  
    Ocorre quando os dados do usuário são alterados através da janela de cadastro de usuários. Só ocorre o envio do e-mail quando a propriedade Ativo estiver definida para True.  
  
**Propriedade EmailRemetente**  
    Defina essa propriedade contendo a conta de e-mail do usuário. Exemplo: se os e-mail's forem emitidos por uma conta do Yahoo chamada "[suporteMinhaEmpresa@yahoo.com.br](mailto:suporteMinhaEmpresa@yahoo.com.br)", preencha com essa conta.  
  
**Propriedade EsqueceuSenha**   
  
    Ocorre quando o login de usuário está sendo executado. Quando o usuário preenche o seu nome na janela de login, e que por ventura tenha esquecido da senha, quando clica no label "***Esqueci a senha***", o usercontrol efetuará um pesquisa procurando pelo endereço de e-mail registrado no banco de dados, no cadastro de usuários; se o nome de usuário for encontrando no sistema, o envio do e-mail e executado. A propriedade **LabelLoginForm** é o texto que aparece na janela de login para o usuário. Você pode mudar essa propriedade colocando qualquer outro texto que achar ser melhor. A propriedade **MensagemEmailEnviado**, é uma mensagem que é apresentada ao usuário depois do componente enviar com sucesso o e-mail.  
  
**Propriedade NomeRemetente**  
  
    É preenchido no campo "**De:**" do e-mail. Quando o e-mail é recebido, aparece **DE:** Sistema JEDY de E-mail, para: "Sabrina Sapeca" por exemplo.  
  
**Propriedade Porta**  
  
    É a porta padrão de envio de e-mail pelo protocolo SMTP (Simple Mail Transfer Protocol). O número dessa porta é 25 por Default, porém outros servidores de e-mail poderão exigir uma porta diferente. Um exemplo é para os que usam o servidor de e-mail do Yahoo que exige a porta 467 (se não estou enganado).   
  
**Propriedade Senha**  
  
    É a senha que você usar para acessar essa conta. Caso a sua senha seja "123456654321" é essa a senha que deve ser preenchida.  
  
**Propriedade SenhaForcada**  
    Quando alguém ou você tenta acessar o sistema tentando adivinhar a senha.  
  
**Propriedade SenhaTrocada**  
  
    O e-mail é enviado quando o usuário troca a senha através do comando de menu   
definido na propriedade ***ChangePasswordform > MenuItem*** do componente **UserControl**. Caso   
esta função tenha sido liberada para os usuários.   
  
**Propriedade ServidorSMTP**  
    Nessa propriedade deve ser especificado o servidor de SMTP. Exemplo: Se você usa o Yahoo como servidor de e-mail, defina "smtp.mail.yahoo.com.br". Se a sua empresa tem um domínio, então use o servidor do seu domínio, exemplo: "smtp.Jedy.com.br".  
  
**Propriedade Usuario**  
  
    Deve ser o nome do proprietário ou de login da conta de e-mail. Exemplo: Se você usa o servidor de E-mail Bol e seu login de usuário é SouJedySetimoDam; esse é o nome de usuário que deve ser registrado na propriedade.   
  
**Veja o Object Inspector com todas as propriedades expandidas**  
  
  
  
    Todos os comandos que acabam resultando no envio de um e-mail possuem a propriedade Mensagem. Essa propriedade pode ser preenchida com um texto apropriado a cada evento em que foi acionado.   
  
Exemplo: Se o usuário trocou a senha, você pode enviar um texto (corpo da mensagem) mais ou menos com esse dizeres: "Através do comando de troca de senha, você entrou com uma senha nova e que terá validade no próximo login. Essa mensagem não deve ser respondida. Ela e gerada automaticamente pelo seu sistema mais preferido - Sistemas Jedy Sétimo Dam. Obrigado! Passe bem!".   
  
    Porém, você lembra do que falei no início desse tutorial? Sobre aquela parte do código-fonte UCMail? Nós podemos nos beneficar criando uma mensagem mais ou menos assim, veja a figura abaixo:  
  
  
  
    Feito isso amigos, podemos fazer um teste digitando um nome válido de usuário registrado em nosso pequeno sistema e que tenha um e-mail válido é claro. Digite o nome do usuário e quando o campo "Esqueci a Senha" ficar habilitado, clique nele. Uma pequena janela se abrirá informando os passos do envio do e-mail. Se você estive usando o Yahoo ou qualquer outro provedor de serviços de mensagens (smtp), deverá consultar a ajuda dos provedores e obter as informações sobre o sistema de envio e recepção de mensagens usando o POP e SMTP. Se ocorrer uma pequena exceção exigindo o recurso de envio seguro, acesse o site [www.usercontrol.net](http://www.usercontrol.net) e lá você encontrará um post do usuário **bitzero\_000** (que sou eu) falando a forma de corrigir esse pequeno problema. Isso deverá acontecer principalmente com os serviços de mensagem do Yahoo, pelos testes que fiz.  
  
    Amigo, aqui me despeço (acho que é assim que se escreve) em relação ao tutorial dos componentes do UserControl. Foi muito bom saber que muitos usuários acessaram e estão procurando conhecer esse conjunto de componentes gratuítos. Todos aqueles que fizerem questão de participar e ampliar os recursos dessa maravilhosa suite, são bem-vindos. Só assim, poderemos melhorar ainda mais o que já temos.   
  
    Não esquecendo aqui também de ressaltar a importância do principal desenvolvedor e também quem disponibilizou os componentes - o [Rodrigo](http://www.usercontrol.net/index.php?option=com_contact&task=view&contact_id=2&Itemid=3) (Qmd), ao [Luiz Benevenuto](http://www.usercontrol.net/index.php?option=com_contact&task=view&contact_id=4&Itemid=3) - desenvolvedor e mantenedor do site, e [Fellipe Henrique](http://www.usercontrol.net/index.php?option=com_contact&task=view&contact_id=3&Itemid=3) (desenvolvedor); todos eles, desenvolvedores e com uma proposta aberta de contribuir com o mundo dos desenvolvedores sem cobrar nada em troca.  
  
Abraços – Exio

Propriedades expandidas do User Control:

