

Criando uma agenda com o Lazarus

Este artigo mostra como criar uma agenda simples com o Lazarus, uma IDE para FreePascal. Além de ser uma boa dica para treinar programação em pascal, usando arquivo em memória, array, record, variáveis, procedure, etc e tudo em modo gráfico |'8^).

Leia o artigo do Cabelo sobre o Lazarus:

www.vivaolinux.com.br/artigos/verArtigo.php?codigo=176&pagina=1

1. Baixando e instalando

Baixe os pacotes necessários para instalar o Lazarus em <http://www.lazarus.freepascal.org>.

No menu selecione os links:

Download/binaries/linux/Current Lazarus Binary RPM e baixe os pacotes:

fpc-1.0.11-031022.i386.rpm (ou a última versão estável)

lazarus-0.9.0.6-fpc_1.0.11-031022.i386.rpm (lazarus para o freepascal escolhido)

Download/resources/libraries/gtk2_bindings e baixe o pacote:

gtk2fpc-1.0.2-fpc_1.0.7_laz.20030301.i386.rpm

Agora basta instalar os pacotes:

```
rpm -ivh <nome_pacote>
```

Na área de download do Lazarus você pode encontrar os pacotes binários para instalar no Windows

2. Criando o formulário da agenda

Execute o Lazarus

Como deu para perceber o Lazarus é uma IDE como o Delphi ou Kylix, com Menu, Componentes (botões, caixa de texto, menu, painéis etc), Object Inspector (propriedades e eventos) (figura 1).

Selecione no menu File/New Form. Será criado um form chamado Form1 (figura 2). No Object Inspector, mude a propriedade Caption para "Agenda FreePASCAL".

Na barra de Componentes, (figura 3) selecione a aba Standard e clique no TLABEL (ícone com a letra "A"), agora clique no formulário para incluir um Label, posicione a mesma no canto superior esquerdo. No Object Inspector, mude a propriedade CAPTION para Código e Autosize para TRUE.

Agora vamos selecionar, com um clique, na aba Standard o componente TEDIT para receber o valor do código. Clique no formulário, logo a frente da label Código. No Object Inspector, apague o conteúdo da propriedade TEXT e modifique a propriedade NAME para EditCod.

Repita os passos dos dois parágrafos acima para os campos, nome, endereço, complemento, telefone (res), telefone (serv), celular, e-mail, mudando a propriedade para EditNome, EditEnde, EditComp, EditTelR, EditTelS, EditTelC, EditMail, respectivamente (figura 4). No exemplo alinhei as TLabel pela direita e os TEdit pela esquerda, ajustando o tamanho do último de acordo com a necessidade.

Ajuste as posições dos objetos dentro do formulário, o Lazarus tem uma facilidade muito interessante que é mostrar com linhas coloridas os posicionamentos horizontais e verticais, facilitando o alinhamento durante a construção do formulário.

Agora que você já ajustou os campos salve o projeto e o formulário, clicando em File/Save All (Ctrl+Sft+S), será solicitado nome do projeto e do form, digite proj_agenda.lpr para o projeto e agenda.pas para o formulário.

3. Colocando botões de controle

Já colocamos os campos, agora vamos colocar os botões de controle para a agenda. Clique sobre o componente TBOTTOM (ícone com botão "OK"), na aba Standard e depois sobre o formulário, na parte inferior. Renomeie o botão para BtPrimeiro, na propriedade Name do Object Inspector e modifique a propriedade Caption para " |< Primeiro ". Dica: use pipe e menor que, para fazer a seta de início. Repita estes passos para criar os botões "< Anterior", "Próximo >", "Último >|", "Novo" e "Confirmar", renomeia a propriedade Name para BtAnterior, BtProximo, BtUltimo, BtNovo e BtConfirmar respectivamente (figura 5).

4. Colocando dica do campos

O Lazarus permite que sejam criadas dicas para os campos, estas dicas ajudam os usuário durante o preenchimento dos dados no formulário. Clique sobre o TEDIT onde será digitado o nome. No Object Inspector, edite a propriedade HINT para "Digite o nome" (sem aspas) e a propriedade SHOWHINT para True. Execute o form através do menu em Run/Run, você pode executar também clicando no ícone com a seta verde ou teclando <F9>.

Passa o mouse sobre o campo Nome para ver o resultado. Feche o formulário em execução e repita o processo para os demais campos colocando as dicas de digitação.

5. Criando estrutura de registros para a agenda

Vamos criar duas estruturas de registros, uma para armazenar informações da "base de dados" e outra para armazenar os dados. No Lazarus Source Editor (área de edição do código do Lazarus), na área de Type digite:

```
Type
  tBase=record
    QtReg: integer;
    PsReg: integer;
  end;

  tAgenda=record
    Cod:string;
    Nome:string;
    Ende:string;
    Comp:string;
    TelR:string;
    TelS:string;
    TelC:string;
```

```
Mail:string;  
end;
```

Agora vamos criar a variável e o array para receber os registros. No Lazarus Source Editor, na área de Var digite:

```
Var  
  Base:tBase;  
  vAgenda:array[1..100] of tAgenda;
```

6. Colocando a mão na massa. Iniciando a procedure ACTIVATE.

O que estiver programado no FormACTIVATE será executado sempre que você carregar o formulário. Para fazer isso clique sobre o formulário (Form1:TForm1), no Object Inspector selecione a aba Events, a seguir clique sobre ONACTIVATE, aparecerá um campo para preenchimento e um botão a direita (com três pontinho), clique sobre o botão para criar a procedure TForm1.Form1ACTIVATE. Edite a procedure conforme as linhas abaixo:

```
procedure TForm1.Form1ACTIVATE(Sender: TObject);  
begin  
  Base.QtReg:=1; {inicia quantidade de registro}  
  Base.PsReg:=1; {define a posicao do registro como um}  
  EditCod.ReadOnly:=True; {nao permite mudança do codigo}  
  EditCod.text:=InttoStr(Base.PsReg); {converte para string e atribui a  
EditCod}  
end;
```

7. Criando evento para confirmar inclusão do registro no array vAgenda

Agora vamos criar o evento para o botão "Confirma", com a finalidade de gravar os dados no array. Primeiro selecione o botão Confirma, a seguir clique na aba Events do Object Inspector e selecione o evento ONCLICK (botão com três ponto, como no exemplo acima). Será criado a procedure TForm1.BtConfirmaCLICK. Edite a mesma conforme as linhas abaixo:

```
procedure TForm1.BtConfirmaCLICK(Sender: TObject);  
begin  
  {atribui ao array o conteudo de TEdit}  
  vAgenda[Base.PsReg].Cod:=EditCod.text;  
  vAgenda[Base.PsReg].Nome:=EditNome.text;  
  vAgenda[Base.PsReg].Ende:=EditEnde.text;  
  vAgenda[Base.PsReg].Comp:=EditComp.text;  
  vAgenda[Base.PsReg].TelR:=EditTelR.text;  
  vAgenda[Base.PsReg].TelS:=EditTelS.text;  
  vAgenda[Base.PsReg].TelC:=EditTelC.text;  
  vAgenda[Base.PsReg].Mail:=EditMail.text;  
end;
```

Cada linha tem a função de gravar os dados digitados nos campos TEdit para seu campo correspondente no array, na posição do código atual.

NOTA: o campo de código esta sendo usado como chave ligada ao índice do array apenas

para efeito educativo, considerando-se o fato de não haver exclusão nem gravação em disco dos registros para este exemplo.

8. Criando um novo registro

Neste evento, que será associado ao botão Novo, será criado um novo registro para a Agenda. Selecione o botão "Novo" e no Object Inspector selecione ONCLICK, e crie a procedure TForm1.BtNovoCLICK. Edite conforme as linhas abaixo.

```
procedure TForm1.BtNovoCLICK(Sender: TObject);
begin
    Base.QtReg:=Base.QtReg+1;
    Base.PsReg:=Base.QtReg;
    EditCod.text:=IntToStr(Base.QtReg);
    EditNome.text:='';
    EditEnde.text:='';
    EditComp.text:='';
    EditTelR.text:='';
    EditTelS.text:='';
    EditTelC.text:='';
    EditMail.text:='';
end;
```

9. Movimentando-se dentro do array

Agora vamos criar os eventos para movimentar uma posição para cima e para baixo dentro do array, desta forma navegando dentro do registro da agenda. Clique sobre o botão "< Anterior" e na aba Events crie a procedure TForm1.BtAnteriorCLICK. Editando-a confirme código abaixo:

```
procedure TForm1.BtAnteriorCLICK(Sender: TObject);
begin
    {se posicao for maior que um subtrai um}
    if Base.PsReg > 1 then
        {retorna uma posição no array}
        Base.PsReg:=Base.PsReg-1;
    {atribui ao TEdit conteudo do array}
    EditCod.text:=vAgenda[Base.PsReg].Cod;
    EditNome.text:=vAgenda[Base.PsReg].Nome;
    EditEnde.text:=vAgenda[Base.PsReg].Ende;
    EditComp.text:=vAgenda[Base.PsReg].Comp;
    EditTelR.text:=vAgenda[Base.PsReg].TelR;
    EditTelS.text:=vAgenda[Base.PsReg].TelS;
    EditTelC.text:=vAgenda[Base.PsReg].TelC;
    EditMail.text:=vAgenda[Base.PsReg].Mail;
end;
```

Faça o mesmo procedimento com o botão "Próximo >" e crie a procedure TForm1.BtProximoCLICK, conforme código abaixo:

```
procedure TForm1.BtProximoCLICK(Sender: TObject);
begin
```

```

{se posicao for menor que quantidade soma um}
if Base.PsReg < Base.QtReg then
  {avança uma posição no array}
  Base.PsReg:=Base.PsReg+1;
{atribui ao TEdit conteudo do array}
EditCod.text:=vAgenda[Base.PsReg].Cod;
EditNome.text:=vAgenda[Base.PsReg].Nome;
EditEnde.text:=vAgenda[Base.PsReg].Ende;
EditComp.text:=vAgenda[Base.PsReg].Comp;
EditTelR.text:=vAgenda[Base.PsReg].TelR;
EditTelS.text:=vAgenda[Base.PsReg].TelS;
EditTelC.text:=vAgenda[Base.PsReg].TelC;
EditMail.text:=vAgenda[Base.PsReg].Mail;
end;

```

Saltando para o primeiro registro. Crie a procedure TForm1.BtPrimeiroCLICK e edite o código:

```

procedure TForm1.BtPrimeiroCLICK(Sender: TObject);
begin
  {vai para posição 1 do array}
  Base.PsReg:=1;
  {atribui ao TEdit conteudo do array}
  EditCod.text:=vAgenda[Base.PsReg].Cod;
  EditNome.text:=vAgenda[Base.PsReg].Nome;
  EditEnde.text:=vAgenda[Base.PsReg].Ende;
  EditComp.text:=vAgenda[Base.PsReg].Comp;
  EditTelR.text:=vAgenda[Base.PsReg].TelR;
  EditTelS.text:=vAgenda[Base.PsReg].TelS;
  EditTelC.text:=vAgenda[Base.PsReg].TelC;
  EditMail.text:=vAgenda[Base.PsReg].Mail;
end;

```

Saltando para o último registro. Crie a procedure TForm1.BtUltimoCLICK e edite o código:

```

procedure TForm1.BtUltimoCLICK(Sender: TObject);
begin
  {vai para o ultimo registro}
  Base.PsReg:=Base.QtReg;
  {atribui ao TEdit conteudo do array}
  EditCod.text:=vAgenda[Base.PsReg].Cod;
  EditNome.text:=vAgenda[Base.PsReg].Nome;
  EditEnde.text:=vAgenda[Base.PsReg].Ende;
  EditComp.text:=vAgenda[Base.PsReg].Comp;
  EditTelR.text:=vAgenda[Base.PsReg].TelR;
  EditTelS.text:=vAgenda[Base.PsReg].TelS;
  EditTelC.text:=vAgenda[Base.PsReg].TelC;
  EditMail.text:=vAgenda[Base.PsReg].Mail;
end;

```