INSTITUTO FEDERAL MINAS GERAIS - *Campus Bambuí*
Lazarus
Prof. Marcos Roberto Ribeiro

Lista de Exercícios 03

Observação: Os formulários dos aplicativos e todos os seus componentes devem ser nomeados e terem suas propriedades indicadas corretamente.

---

1. Refaça todos os exemplos e exercícios da aula sobre "Principais Componentes do Lazarus seus Eventos".

2. Informe para que são usados os seguintes componentes:

   (a) Menu; (b) Button; (c) Label; (d) Edit; (e) Memo; (f) CheckBox; (g) ListBox; (h) ComboBox; (i) GroupBox; (j) Panel;

3. Crie um formulário com um menu igual ao exibido na Figura 1. Codifique os menus da seguinte maneira:

   Arquivo > **Novo** > **Documento** Exibir a mensagem "Novo documento criado";
   Arquivo > **Novo** > **Planilha** Exibir a mensagem "Nova planilha criada";
   Arquivo > **Abrir** Exibir a mensagem "Abrindo arquivo";
   Arquivo > **Salvar** Exibir a mensagem "Salvando arquivo";
   Arquivo > **Sair** Finalizar o programa;

Figura 1: Menu

4. Modifique o programa do exercício anterior, acrescentando os botões:

   - Novo documento;
   - Nova planilha;
   - Abrir;
   - Salvar;
   - Sair;

   Os botões devem realizar as mesmas tarefas dos menus já codificados, porém isto deve ser feito sem acrescentar novas linhas de código.

5. Crie um formulário com quatro botões que o movimente das direções cima, baixo, esquerda e direita.

6. Crie um programa semelhante ao do exercício anterior. Porém, não utilize botões para movimentar o formulário, utilize as setas do teclado ($\leftarrow, \uparrow, \rightarrow, \downarrow$). Dica utilize o evento *KeyDown*.

7. Implemente um programa com um formulário e um **Label**. Este *Label* deve se movimentar pelo formulário de acordo com a posição do mouse. Além disto, o **Label** deve exibir a posição horizontal e vertical do mouse.

8. Crie um programa que contenha quatro *Edits*:

   (a) Os três primeiros *Edits* podem ser editados, mas o quatro deve ser somente leitura;

   (b) A medida que os três primeiros *Edits* forem alterados, o conteúdo do quarto *Edit* deve ser a concatenação do conteúdo dos três primeiros separados por vírgula;

   (c) Exiba o número de caracteres do quarto *Edit* em um *Label*.

9. Crie um formulários com 10 botões numerados de 0 a 9 e um *Edit* em branco. Acrescente também um botão "Limpar". Os botões numerados devem acrescentar seus respectivos dígitos ao conteúdo do *Edit*. Já o botão "Limpar" deve apagar todo o conteúdo do *Edit*.

10. Projete um formulário com um *Memo* chamado MemoLinhas e um *Edit* chamado EditPalavra. Acrescente também os seguintes botões:

    (a) Adicionar o texto do EditPalavra no final do MemoLinhas;

    (b) Limpar o conteúdo do MemoLinhas;

    (c) Exibir uma mensagem com o número de linhas do MemoLinhas;

    (d) Inserir o texto do EditPalavra na primeira linha do MemoLinhas;

    (e) Verificar se o texto do EditPalavra existe dentro o MemoLinhas.

11. Construa um formulário com 10 *Checkboxes* cada um com um nome de cidade diferente. Codifique o programa para que quando houver alterações na marcação das cidades seja exibida uma mensagem com o número de cidades marcadas.

12. Crie um formulário igual ao mostrado na Figura 2. Codifique o botão ">" para que mova o elemento selecionado na lista 1 para a lista 2 e o botão "<" para que mova o elemento selecionado da lista 2 para a lista 1.

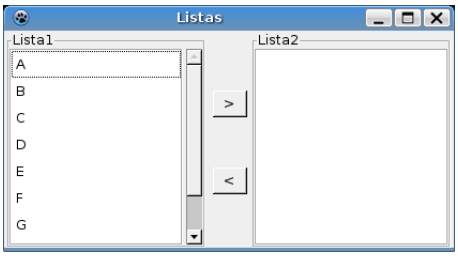

Figura 2: Menu

13. Implemente um programa no Lazarus com um *CheckGroup* contendo todos os estados brasileiros e cinco *Labels* um para cada região do Brasil. A medida que o usuário marcar ou desmarcar os estados, os *Labels* devem mostrar o número de estados marcados em cada região.

14. Crie um formulário igual ao mostrado na Figura 3. Observações:

    - O botão "Adicionar deve verificar se o número é válido através da função *Val()*. Se o número for inválido deve ser exibida uma mensagem para o usuário informando esta situação, caso contrário o número é acrescentado a lista da direita;
    - O botão "Remover" deve remover o item selecionado na lista;
    - O botão "Limpar" deve remover todos os números da lista;
    - Além disto, sempre que houver uma alteração na lista o programa deve calcular a soma dos números e exibi-la no painel inferior.

15. Crie programa com três *Edits* para receber os termos A, B e C de uma equação de segundo grau no formato $Ax^2 + Bx + C = 0$. O programa deve calcular as raízes da equação e exibi-las em um painel. Observações:

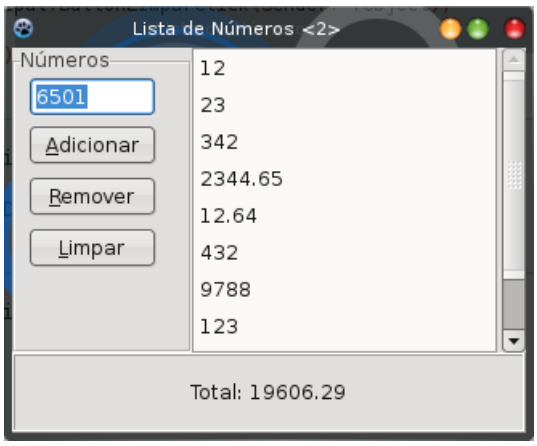

Figura 3: Lista de Números

- Se o termo $A = 0$ então a equação não é uma equação de segundo grau;
- Seja $\Delta = B^2 - 4 \times A \times C$. Se $\Delta < 0$ então a equação não possui raízes. Se $\Delta = 0$ então a equação possui apenas uma raiz.

16. Crie um formulário igual ao mostrado na Figura 4. Observações:

    - O botão "Adicionar" deve verificar se as notas são válidas através da função *Val()*. Se as notas forem válidas, o nome do aluno deve ser inserido na primeira lista, a primeira nota na segunda lista, a segunda nota na terceira lista e na quarta lista é inserida a soma da nota 1 mais a nota 2;
    - O botão "Remover" deve remover o alunos selecionado e suas respectivas notas;
    - O botão "Alterar" deve fazer a mesma verificação do botão "Adicionar" e modificar os dados selecionados nas listas;
    - Quando o usuário selecionar um item de qualquer uma das listas, o programa deve selecionar os itens da mesma posição nas demais listas;
    - Dica: crie uma sob-rotina chamada **Seleciona(Posicao: Integer)** que selecione em todas as listas.

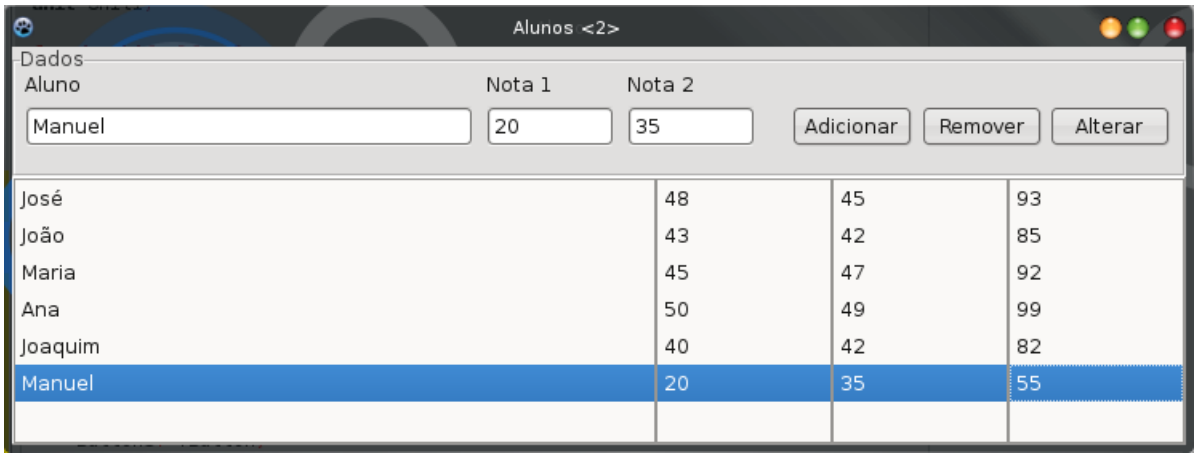

Figura 4: Alunos

17. Modifique o programa do exercício anterior para que não seja permitido a adição de alunos já existentes.

18. Desenvolva um programa com um formulário similar ao da Figura 5. Observações:

    - O programa deve ter uma lista oculta para armazenar os significados das palavras;

- O botão "Adicionar deve verificar se a palavra e seu significado são válidos antes de inseri-los;
- As palavras devem ser listadas em ordem alfabética;
- O botão "Alterar" deve solicitar um novo significado para a palavra, validá-lo e modificá-lo;
- O botão "Remover" deve confirmar a remoção da palavra e então apagá-la junto com seu significado;
- O programa deve salvar as palavras e seus respectivos significados em arquivo.

Dicas:

- Utilize as sub-rotinas **InputBox()** e **MessageDlg()** para comunicar com o usuário;
- Pesquise sobre as sub-rotinas **Trim()** e **AnsiLowerCase()**. Descubra como elas podem ajudar neste exercício.

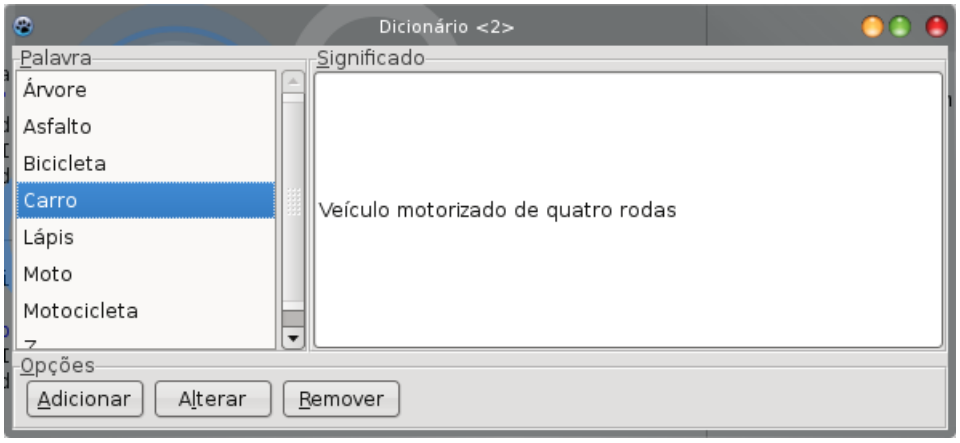

Figura 5: Dicionário

19. Projete um programa no Lazarus para cálculo de MDC entre dois números de forma que tal cálculo possa ser exibido para o usuário passo a passo.

20. Desenvolva uma semelhante àquela mostrada na Figura 6. Observações:

- A calculadora deve suportar as operações +, -, * e /;
- O botão "Calcular" deve realizar a operação selecionada entre os dois números e acrescentá-la no texto da parte inferior;
- Os números devem ser valiados com a função *Val()* antes de realizar a operação;
- O botão "Limpar" deve limpar todas as operações já registradas no texto inferior.

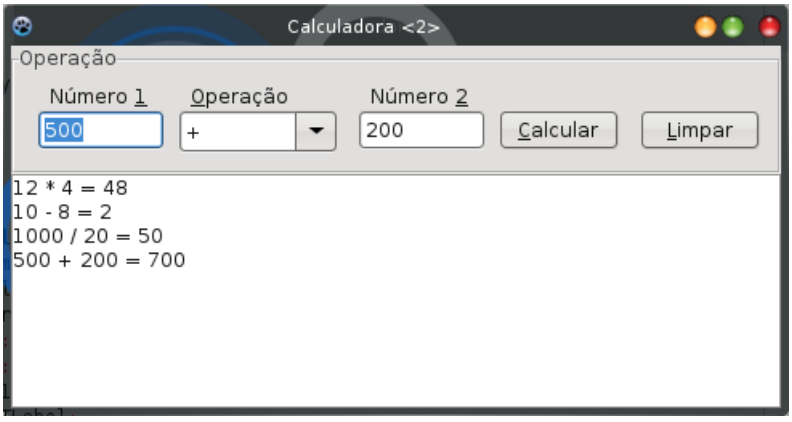

Figura 6: Calculadora