

Gabarito da AP 2 de Programação I

Rio de Janeiro, 3 de dezembro de 2010

1. Questão 1

Desenho dos formulários:

Controle de Passaportes

Nome:

Nº do passaporte:

Figura 1: Formulário 1 (Principal)

As Figuras 1, 2 e 3 correspondem, respectivamente, aos desenhos do Formulário Principal, do Formulário de Dados Pessoais e do Formulário de Dados da Viagem. Os campos de texto com fundo cinza são campos não-editáveis e os retângulos semi-preenchidos com fundo cinza e com texto representam botões.

Segue abaixo as declarações das variáveis responsáveis por armazenar os valores preenchidos pelo usuário do sistema, assim como as componentes de cada formulário:

- Formulário Principal:

`type`

```
TFormPrincipal = class(TForm)
```

Dados Pessoais

Nome:

Nº do Passaporte:

Endereço:

Telefone:

Nacionalidade: ▼

Estado civil: ☒ Solteiro(a)
☐ Casado(a)
☐ Viúvo(a)
☐ Divorciado(a)

Figura 2: Formulário 2 (Pessoal)

Dados da Viagem

Tipo da viagem: ▼

Países: >>

Brasil
EUA
Itália
Japão

Data de ida:

Data de volta:

Dias viajados:

Figura 3: Formulário 3 (Viagem)

```

    DadosPessoaisButton: TButton;
    DadosViagemButton: TButton;
    EditNome: TEdit;
    EditNumPassaporte: TEdit;
    LabelControlePassaportes: TLabel;
    LabelNome: TLabel;
    LabelNumPassaporte: TLabel;

    procedure DadosPessoaisButtonClick(Sender: TObject);
    procedure DadosViagemButtonClick(Sender: TObject);

    private

    public
end;

var
    FormPrincipal : TFormPrincipal;

// Nenhuma variável utilizada
end;
```

- Formulário de Dados Pessoais:

type

```
TFormDadosPessoais = class(TForm)
    SalvarButton: TButton;
    LimparButton: TButton;
    EditNome: TEdit;
    EditNumPassaporte: TEdit;
    EditEndereco: TEdit;
    EditNumTelefone: TEdit;
    ComboBoxNacionalidade: TComboBox;
    RadioButtonSolteiro: TRadioButton;
    RadioButtonCasado: TRadioButton;
    RadioButtonViuvo: TRadioButton;
    RadioButtonDivorciado: TRadioButton;
    LabelDadosPessoais: TLabel;
    LabelNome: TLabel;
    LabelNumPassaporte: TLabel;
    LabelEndereco: TLabel;
    LabelTelefone: TLabel;
    LabelNacionalidade: TLabel;
    LabelEstadoCivil: TLabel;

    procedure LimparButtonClick(Sender: TObject);
    procedure SalvarButtonClick(Sender: TObject);

private

public
end;

var
    i : Integer; // Indice usado durante o codigo de salvamento
    inserirPassageiro : Boolean; // Verificacao de insercao/atualizacao

end;
```

- Formulário de Dados da Viagem:

type

```
TFormDadosViagem = class(TForm)
    AdicionarButton: TButton;
    SalvarButton: TButton;
    LimparButton: TButton;
    EditPaíses: TEdit;
    EditDataIda: TEdit;
    EditDataVolta: TEdit;
    EditDiasViajados: TEdit;
    ComboBoxTipo: TComboBox;
    LabelDadosViagem: TLabel;
    LabelTipo: TLabel;
    LabelNumPassaporte: TLabel;
    LabelPaíses: TLabel;
    LabelDataIda: TLabel;
    LabelDataVolta: TLabel;
    LabelDiasViajados: TLabel;
    ListBoxPaíses: TListBox;

    procedure LimparButtonClick(Sender: TObject);
    procedure SalvarButtonClick(Sender: TObject);

    private

    public
end;

var
    i : Integer; // Índice usado durante o código de salvamento
    inserirCliente : Boolean; // Verificação de inserção/atualização
    listaPaísesBanco : TStringList; // Lista de países do passageiro

end;
```

2. Questão 2

- Formulário Principal:

Labels (sem valor funcional, apenas texto):

- LabelControlePassaportes: texto ‘Controle de Passaportes’;
- LabelNome: texto ‘Nome:’;
- LabelNumMat: texto ‘Nº do passaporte:’;

EditBoxes:

- EditNome: campo onde será preenchido o nome do passageiro;

validação: nenhuma.

funcionalidade: preencherá o campo ‘Nome’ do Formulário de Dados Pessoais.

- EditNumPassaporte: campo onde será preenchido o número do passaporte;

validação: nenhuma.

funcionalidade: preencherá o campo ‘Nº do Pasaporte’ do Formulário de Dados Pessoais.

Botões:

- DadosPessoaisButton: abrirá o formulário de dados pessoais, passando os valores em ‘Nome’ e ‘CPF’ para ele;
- DadosViagemButton: abrirá o formulário de dados da viagem;

- Formulário de Dados Pessoais:

Labels (sem valor funcional, apenas texto):

- LabelDadosPessoais: texto ‘Dados Pessoais’;
- LabelNome: texto ‘Nome:’;
- LabelNumPassaporte: texto ‘Nº do Passaporte:’;
- LabelEndereco: texto ‘Endereço:’;

- LabelTelefone: texto 'Telefone:';
- LabelNacionalidade: texto 'Nacionalidade:';
- LabelEstadoCivil: texto 'Estado Civil:';

EditBoxes:

- EditNome (somente leitura): campo que conterá o nome do passageiro, informado no Formulário Principal;

validação: nenhuma.

funcionalidade: nenhuma.

- EditNumPassaporte (somente leitura): campo que conterá o Número do passaporte do passageiro, informado no Formulário Principal;

validação: nenhuma.

funcionalidade: nenhuma.

- EditEndereco: campo onde será preenchido o endereço do passageiro;

validação: nenhuma.

funcionalidade: armazenar o endereço do passageiro, para possível atualização/inserção posterior no banco.

- EditTelefone: campo onde será preenchido o telefone do passageiro;

validação: (opcional) verificar se foram preenchidos 8 números (10, se for considerado DDD).

funcionalidade: armazenar o telefone do passageiro, para possível atualização/inserção posterior no banco.

ComboBoxes:

- ComboBoxNacionalidade: conterá as nacionalidades possíveis do passageiro;

validação: nenhuma.

funcionalidade: indicar qual a nacionalidade do passageiro, para futura atualização/inserção de seu cadastro.

RadioButtons:

- RadioButtonSolteiro: *radio* pertencente ao grupamento que define estado civil, opção ‘Solteiro(a)’;
- RadioButtonCasado: *radio* pertencente ao grupamento que define estado civil, opção ‘Casado(a)’;
- RadioButtonViuvo: *radio* pertencente ao grupamento que define estado civil, opção ‘Viúvo(a)’;
- RadioButtonDivorciado: *radio* pertencente ao grupamento que define estado civil, opção ‘Divorciado(a)’;

Botões:

- LimparButton: limpará os campos ‘Endereço’, ‘Telefone’, ‘Nacionalidade’ e ‘Estado Civil’;
- SalvarButton: salvará no banco todos os dados pessoais do passageiro. Caso ele já exista, os dados serão apenas atualizados.

● Formulário de Dados da Viagem:

Labels (sem valor funcional, apenas texto):

- LabelDadosViagem: texto ‘Dados da Viagem’;
- LabelTipo: texto ‘Tipo:’;
- LabelNumPassaporte: texto ‘Nº do passaporte:’;
- LabelDadosViagem: texto ‘Países:’;
- LabelTipo: texto ‘Data de Ida:’;
- LabelNumPassaporte: texto ‘Data de Volta:’;
- LabelDadosViagem: texto ‘Dias Viajados:’;

EditBoxes:

- EditPaíses: campo onde será preenchido o país a ser incluído na lista de países;

validação: nenhuma.

funcionalidade: adicionará países na *ListBox* de países.

- EditDataIda: campo onde será preenchida a data de ida da viagem;

validação: (opcional) verificar o preenchimento do formato 'DD/MM/YYYY'.

funcionalidade: armazenar a data de ida da viagem, para possível atualização/inserção posterior no banco.

- EditDataVolta: campo onde será preenchida a data de volta da viagem;

validação: (obrigatório) verificar o preenchimento do formato 'DD/MM/YYYY'.

funcionalidade: armazenar a data de volta da viagem, para possível atualização/inserção posterior no banco.

- EditDiasViajados (somente leitura): campo que conterá o resultado do cálculo dos dias viajados;

validação: nenhuma.

funcionalidade: armazenar o número de dias viajados, para possível atualização/inserção posterior no banco.

ComboBoxes:

- ComboBoxTipo: conterá os tipos possíveis de viagem;

validação: nenhuma.

funcionalidade: indicar qual o tipo da viagem, para futura atualização/inserção de seu cadastro.

ListBoxes:

- `ListBoxPaíses`: Conterá a listagem de países que serão visitados durante a viagem;

validação: nenhuma.

funcionalidade: indicar quais países serão adicionados/atualizados na tabela 'País' relacionada aos dados da viagem.

Botões:

- `AdicionarButton`: adicionará o país contido no *EditPaises*; na *ListEditPaises*. O seu atributo e o seu formato foram editados para a sua visualização ficar no formato apresentado na tela da questão 1: um pequeno quadrado com duas setas "»" ao lado do *TListBox*.
- `LimparButton`: limpará os campos 'Data de ida', 'Data de volta', 'Dias viajados' e a *ListBox* 'Países';
- `SalvarButton`: salvará no banco todos os dados da viagem do passageiro. Caso eles já existam, os dados serão apenas atualizados.

3. Questão 3

- Formulário Principal:

- procedure **DadosPessoaisButtonClick**(Sender: TObject):

Esse procedimento deve realizar duas ações: a transferência dos valores dos campos 'Nome' e 'Nº do passaporte' do Formulário Principal para o Formulário de Dados Pessoais e a abertura do Formulário de Dados Pessoais. A primeira ação consiste em atribuir os elementos citados do Formulário Principal ao o Formulário de Dados Pessoais e a segunda, usar o atributo 'Show' do Formulário de Dados Pessoais par exibi-lo.

```
EditNome.Text := '';  
(...)
```

Observação: é necessário importar a Unit contendo o formulário de dados pessoais, através da cláusula **Uses** da seção de **interface**.

- procedure **DadosViagemButtonClick**(Sender: TObject):

Esse procedimento deve abrir o Formulário de Dados da Viagem através do atributo ‘Show’ do Formulário de Dados da Viagem.

- Formulário de Dados Pessoais:

- procedure **LimparButtonClick**(Sender: TObject):

A limpeza dos campos ‘Endereço’ e ‘Telefone’ é feita através da atribuição de ‘’ ao texto de cada um desses campos.

- procedure **SalvarButtonClick**(Sender: TObject):

Esse procedimento realiza o salvamento dos dados preenchidos pelo usuário no Formulário de Dados Pessoais em um banco de dados à escolha do aluno. Pode ser usado qualquer banco, mas é *indispensável* que seja explicada a forma de uso do banco de dados no desenvolvimento do sistema e no salvamento dos dados. A título de exemplo, foi usado o banco de dados MySQL 5.1.

Existem diversas formas de se conectar a um banco de dados MySQL. Uma delas é apresentada na aula 14. Outra, a usada no gabarito e no código-fonte da AD2 (ambos se encontram na seção “Avaliações e Gabaritos” do site), usa as próprias componentes fornecidas pelo Lazarus: *TMySQL50Connection*, *TSQLQuery* e *TSQLTransaction*.

- *TMySQL50Connection*: responsável pela conexão do Lazarus a uma instância de servidor MySQL 5.X em execução;
- *TSQLQuery*: responsável pelas operações com o banco conectado à instância de *TMySQL50Connection* corrente. Quaisquer operações SQL podem ser executadas através dessa componente;
- *TSQLTransaction*: responsável pela integridade do banco (controla as transações, o sincronismo das componentes com o status atual do banco e armazena informações de sessão).

- Formulário de Dados da Viagem:

- procedure **AdicionarButtonClick**(Sender: TObject):

Essee botão adicionará o país contido na *EditPaíses*, **caso ele já não exista**, na *ListEditPaíses*.

- procedure **LimparButtonClick**(Sender: TObject):

A limpeza dos campos ‘Países’, ‘Data de Ida’, ‘Data de Volta’ e ‘Dias Viajados’ é feita através da atribuição de ‘’ ao texto de cada um desses campos.

- procedure **SalvarButtonClick**(Sender: TObject):

Esse procedimento realiza três ações: valida as datas digitadas (essa validação é obrigatória, devido ao cálculo dos dias viajados), calcula os dias viajados e atualiza/insere o registro de viagem do passageiro no banco de dados. Como dito no botão “Salvar” do formulário anterior, existem diversas formas de se comunicar com o banco de dados. Fica a critério do aluno.