

# Gabarito da AP 1 de Programação I

Rio de Janeiro, 23 de setembro de 2011

1. **Faça o desenho dos formulários e declare as variáveis que serão utilizadas para armazenar os dados fornecidos: (3 pontos)**

A interface, como descrita no enunciado principal da questão, deve ser similar à apresentada abaixo:

Figura 1: Formulário Principal

Figura 2: Formulário de Sessão

Existe apenas uma variável que deve obrigatoriamente ser usada, que é a instância de cada formulário (pois sempre haverá dados nos formulários):

(...)

**var**

Form1 : TForm1;

(...)

Outras variáveis de armazenamento podem ser citadas, contanto que sejam citadas em pelo menos uma das descrições de evento de clique (lembre-se que a questão pede para listar as variáveis **que serão usadas**). No caso do exemplo de aplicação desse gabarito, mais uma variável foi declarada no formulário de sessão:

(...)

```

var
    // Variavel para guardar o credito inicial, em minutos
    MinutosDeInicio: Integer;

    (...)

```

## 2. Explique cada componente que será empregado no projeto de cada formulário. (2 pontos)

Os componentes usados para criar cada um dos formulários são:

Elemento	Tipo	Descrição
ButtonLimpar	<i>TButton</i>	Botão que limpa os dados da tela
ButtonIniciar	<i>TButton</i>	Botão que abre o segundo formulário
ButtonFechar	<i>TButton</i>	Botão que fecha a aplicação
ComboBoxComputador	<i>TComboBox</i>	Contém os computadores disponíveis
EditNome	<i>TEdit</i>	Campo de nome
GroupBoxAlocacao	<i>TGroupBox</i>	Agrupar os componentes calculados
LabelTitulo	<i>TLabel</i>	Rótulo “Lan House”
LabelNome	<i>TLabel</i>	Rótulo “Nome*.”
LabelComputador	<i>TLabel</i>	Rótulo “Computador*.”
LabelValor	<i>TLabel</i>	Rótulo “Valor*.”
LabelTempo	<i>TLabel</i>	Rótulo “Tempo*.”
LabelSaida	<i>TLabel</i>	Rótulo “Saída.”
LabelCredito	<i>TLabel</i>	Rótulo “Crédito.”
LabelObservacao	<i>TLabel</i>	Rótulo “Os campos ... são obrigatórios”
LabelSaidaValor	<i>TLabel</i>	Label que possuirá a hora de encerramento da sessão
LabelCreditoValor	<i>TLabel</i>	Label que possuirá a quantidade restante de créditos
MaskEditValor	<i>TMaskEdit</i>	Inserção do valor, com máscara “R\$ 000,00”
MaskEditTempo	<i>TMaskEdit</i>	Inserção do tempo de entrada, com máscara “00:00”
ValidateInput	<i>Function</i>	Função que verifica há campos vazios
RemoveLeadingZeroes	<i>Procedure</i>	Procedure que remove zeros à esquerda, exceto um
Split	<i>Procedure</i>	Procedure que divide uma <i>string</i> em partes

Tabela 1: Componentes do Formulário Principal

Elemento	Tipo	Descrição
ButtonEncerrarSessao	<i>TButton</i>	Botão de encerramento de sessão
LabelTitulo	<i>TLabel</i>	Rótulo “Controle de Sessão”
LabelNome	<i>TLabel</i>	Rótulo “Nome:”
LabelComputador	<i>TLabel</i>	Rótulo “Computador:”
LabelEntrada	<i>TLabel</i>	Rótulo “Entrada:”
LabelValor	<i>TLabel</i>	Rótulo “Valor:”
LabelCredito	<i>TLabel</i>	Rótulo “Crédito (em minutos):”
LabelNomeValor	<i>TLabel</i>	Label que contém o nome
LabelComputadorValor	<i>TLabel</i>	Label que contém o computador
LabelEntradaValor	<i>TLabel</i>	Label que contém a hora de entrada
LabelValorValor	<i>TLabel</i>	Label que contém o valor
LabelCreditoValor	<i>TLabel</i>	Label que contém o crédito calculado
TimerMinutes	<i>TTimer</i>	Temporizador de minutos

Tabela 2: Componentes do Formulário de Sessão

### 3. Explique os métodos utilizados nos eventos dos cliques de cada botão. (3 pontos)

- ButtonLimpar → limpa todos os campos do formulário principal. Segue abaixo o código referente a ele:

```

procedure TForm1.ButtonLimparClick(Sender: TObject);
begin
    // 1. Limpando os campos editaveis
    EditNome.Clear;
    MaskEditValor.Text := 'R$ 000,00';
    MaskEditTempo.Text := '00:00';
    ComboBoxComputador.ClearSelection;

    // 2. Limpando os campos calculados
    LabelCreditoValor.Caption := '—';
    LabelSaidaValor.Caption := '—';
end;

```

- ButtonFechar → Fecha a aplicação. Segue abaixo o código referente a ele:

```

procedure TForm1.ButtonFecharClick(Sender: TObject);
begin
    Application.Terminate;

```

**end;**

- ButtonIniciar → Abre o formulário de sessão, calcula o valor do seu campo “Crédito” e preenche seus outros campos com os valores correspondentes do formulário principal. Segue abaixo o código referente a ele:

```
procedure TForm1.ButtonIniciarClick(Sender: TObject);  
var  
    Reais: String;  
    Trechos: TStringList;  
begin  
    if(not ValidateInput) then exit;  
  
    Form2.LabelComputador.Caption := ComboBoxComputador.Text;  
    Form2.LabelNome.Caption := EditNome.Text;  
  
    Trechos := TStringList.Create;  
    Split(' ', MaskEditValor.EditText, Trechos);  
    Reais := Copy(Trechos.Strings[0], 4, Length(Trechos.Strings  
        [0]) - 3);  
  
    Form2.LabelValor.Caption := 'R$ ' + RemoveLeadingZeroes(Reais  
        ) + ', ' + Trechos.Strings[1];  
    Form2.LabelEntrada.Caption := MaskEditTempo.EditText;  
    Form2.LabelCredito.Caption := FloatToStr(trunc(StrToFloat(  
        RemoveLeadingZeroes(Reais + Trechos.Strings[1])) * 0.1));  
  
    Form2.TimerMinutes.Enabled := true;  
    Form2.ShowModal;  
end;  
  
function TForm1.ValidateInput : Boolean;  
var  
    Trechos: TStringList;  
    Reais: String;  
begin  
    Result := true;  
  
    if(trim(EditNome.Text) = '') then  
    begin  
        ShowMessage('O campo ''Nome'' é obrigatório.');
```

```

if (ComboBoxComputador.Text = '') then
begin
    ShowMessage('É obrigatória a escolha de uma máquina.');
```

Result := false;

**exit**;

**end**;

  

```

if ((trim(MaskEditValor.Text) = '') or (trim(MaskEditValor.
    Text) = '0')) then
begin
    ShowMessage('O campo ''Valor'' é obrigatório.');
```

Result := false;

**exit**;

**end**;

  

```

if ((trim(MaskEditTempo.Text) = '') or (trim(MaskEditTempo.
    Text) = '0')) then
begin
    ShowMessage('O campo ''Tempo'' é obrigatório.');
```

Result := false;

**exit**;

**end**;

  

```

Trechos := TStringList.Create;
Split(',', MaskEditValor.EditText, Trechos);
Reais := RemoveLeadingZeroes(Copy(Trechos.Strings[0], 4,
    Length(Trechos.Strings[0]) - 3));

if ((Reais = '0') and (StrToInt(Trechos.Strings[1]) < 10))
    then
begin
    ShowMessage('O valor inserido não é suficiente para
        iniciar a sessão.');
```

Result := false;

**exit**;

**end**;

**end**;

  

```

procedure TForm1.Split(const Delimiter: Char; Input: string; const
    Strings: TStringList);
begin
    Assert(Assigned(Strings));
    Strings.Clear;
    Strings.StrictDelimiter := true;
    Strings.Delimiter := Delimiter;
    Strings.DelimitedText := Input;
end;
```

```

function TForm1.RemoveLeadingZeroes(StringToRemove : String) :
    String;
begin
    Result := StringToRemove;

    while ((Copy(Result, 1, 1) = '0') and (Length(Result) > 1))
        do
            Delete(Result, 1, 1);
end;

```

- ButtonEncerrarSessao → Fecha o formulário de sessão e passa os valores atualizados para o formulário principal. Segue abaixo o código referente a ele:

```

procedure TForm2.ButtonEncerrarSessaoClick(Sender: TObject);
var
    DiferencaTempo: Integer;
    Hora, Trechos: TStringList;
    Reais, HoraStr, MinutoStr: String;
begin
    Trechos := TStringList.Create;
    Form1.Split(' ', LabelValor.Caption, Trechos);
    Reais := Copy(Trechos.Strings[0], 4, Length(Trechos.Strings
        [0]) - 3);
    if (Length(Reais) < 3) then
        Reais := DupeString('0', 3 - Length(Reais)) + Reais;

    DiferencaTempo := MinutosDeInicio - StrToInt(LabelCredito.
        Caption);
    Hora := TStringList.Create;
    Form1.Split(':', LabelEntrada.Caption, Hora);

    HoraStr := IntToStr(StrToInt(Hora.Strings[0]) + (
        DiferencaTempo div 60));
    MinutoStr := IntToStr(StrToInt(Hora.Strings[1]) + (
        DiferencaTempo mod 60));

    if (Length(HoraStr) < 2) then
        HoraStr := '0' + HoraStr;

    if (Length(MinutoStr) < 2) then
        MinutoStr := '0' + MinutoStr;

    Form1.MaskEditValor.Text := 'R$ ' + Reais + ', ' + Trechos.
        Strings[1];

```

```

Form1.LabelSaida.Caption := HoraStr + ':' + MinutoStr;
Form1.LabelCredito.Caption := LabelCredito.Caption;

TimerMinutes.Enabled := false;
Hide;
end;

```

## Observação:

- Para que os formulários se acessem, é necessário declarar na seção ‘uses’ de ambos os formulários as *units* de cada um. Segue abaixo um exemplo de inclusão das *units* dos formulários:

```
unit UnitLanHouse;
```

```
interface
```

```
(...)
```

```
implementation
```

```
uses UnitSessao;
```

```
(...)
```

---

```
unit UnitSessao;
```

```
interface
```

```
(...)
```

```
implementation
```

```
uses UnitLanHouse;
```

```
(...)
```

Foi usada a seção *implementation* para não ocorrer o erro de **dependência circular**.

