

*GABARITO*

**1. Solução**

**Pede-se (cada item vale 2.5 pontos):**

- **Desenhe a interface em Lazarus para exibir o eMDC, nomeando os seus componentes.**

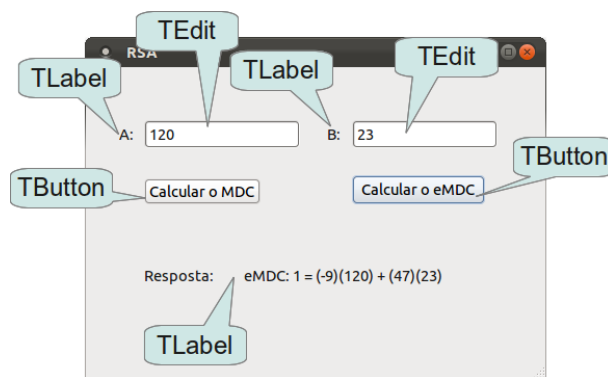


Figura 1: Interface.

- **A sua interface deve aceitar apenas valores positivos. Se for fornecido um valor não numérico, um segundo formulário deve ser aberto com uma mensagem de erro.**

Implementação do botão "Calcular o MDC".

```
1 procedure TRSAcript.MDCButtonClick(Sender: TObject);
2 var
3   x, y: Integer;
4 begin
5   Try
6     x := StrToInt(Edit1.caption);
7     y := StrToInt(Edit2.caption);
8   Except
9     On EConvertError do
10    begin
11      Form1.show; // abrindo formulário com msg de erro
12      exit;
13    end;
14  end;
15  if ((x > 0) and (y > 0)) then
16    gcdLabel.caption := ('MDC(' + IntToStr(x) + ', ' + IntToStr(y) + ') = ' +
      IntToStr(MDC(x, y)))
```

```

17  else
18      showmessage('Os valores de entrada devem ser positivos.');
```

---

```

19  end;
```

- Implemente uma função para retornar o MDC. Dica: cerca de 10 linhas de código e um único comando while.

**function MDC (a, b: integer): integer;**

```

1  function MDC(a, b: Integer): Integer;
2  var
3      resto: Integer;
4  begin
5      while b <> 0 do
6          begin
7              resto := a mod b;
8              a := b;
9              b := resto;
10         end;
11     MDC := a;
12 end;
```

- Implemente um função para retornar os coeficientes do eMDC, nos argumentos x e y, e o valor do MDC, no nome da função. Dica: cerca de 20 linhas de código e um único comando while.

**function eMDC (a, b: integer; var x, y: integer): integer;**

```

1  function eMDC (a,b: integer; Var lastx , lasty: integer): integer;
2  var x, y: integer;
3  var quotient, temp: integer;
4  begin
5      x := 0;
6      lastx := 1;
7      y := 1;
8      lasty := 0;
9      while b <> 0 do
10         begin
11             quotient := trunc (a/b);    // divisão inteira
12             temp := b;
13             b := a mod b;
14             a := temp;
15             temp := x;
16             x := lastx - quotient * x;
17             lastx := temp;
18             temp := y;
19             y := lasty - quotient * y;
20             lasty := temp;
21         end;
22     eMDC := a;
23 end;
```