

Código del programa del Ejercicio 3

```
//-----  
#include <vcl.h>  
#pragma hdrstop  
#include <math.h> UNICA BIBLIOTECA A AGREGAR PARA EL USO DE POTENCIAS  
#include "Unit1.h"  
//-----  
#pragma package(smart_init)  
#pragma link "LPComponent"  
#pragma link "LPDrawLayers"  
#pragma link "LPTransparentColor"  
#pragma link "SLComponentCollection" SON BIBLIOTECAS Y ARCHIVOS QUE EL  
#pragma link "ULBasicControl" BUILDER AGREGA EN NUESTRO PROGRAMA  
#pragma link "ULImage"  
#pragma link "pies"  
#pragma resource "*.dfm"  
TForm1 *Form1;  
//-----  
__fastcall TForm1::TForm1(TComponent* Owner)  
: TForm(Owner)  
{  
}  
  
//-----  
void __fastcall TForm1::Button1Click(TObject *Sender) COMIENZO ACCIONES BOTON CALCULAR  
{  
//Aquí comienza el calculo  
float R1,R2,RTS,RTP,PR1,PR2,V,I,I1,I2,PMAX; DECLARACIÓN DE VARIABLES USADAS  
  
//Hacemos color normal los Edit  
  
Edit1->Color=clWindow;  
Edit2->Color=clWindow; SE PINTAN LOS EDIT DEL COLOR NORMAL DE WINDOWS  
  
//Hacemos invisibles las imágenes de explosión  
  
Image5->Visible=false;  
Image6->Visible=false; LAS EXPLOSIONES SON IMÁGENES QUE SIEMPRE ESTAN SOBRE  
Image7->Visible=false; LAS RESISTENCIAS, SOLO QUE SE HACEN INVISIBLES CUANDO  
Image8->Visible=false; SON NECESARIAS.  
  
//Asignamos valores a las variables  
  
R1 =StrToFloat(Edit1->Text);  
R2 =StrToFloat(Edit2->Text);  
V =StrToFloat(Edit5->Text); SE CARGAN DESDE LOS EDIT LOS VALORES PARA R1, R2 Y V  
  
//Cargamos potencia RMAX permitida de acuerdo a la opción elegida por usuario  
  
if(Form1->RadioGroup1->ItemIndex==0)  
{  
PMAX=0.125;  
}  
  
if(Form1->RadioGroup1->ItemIndex==1)  
{  
PMAX=0.250;  
}  
  
if(Form1->RadioGroup1->ItemIndex==2)
```

```
{
PMAX=0.5;
}
```

```
if(Form1->RadioGroup1->ItemIndex==3)
```

```
{
PMAX=1;
}
```

```
if(Form1->RadioGroup1->ItemIndex==4)
```

```
{
PMAX=2;
}
```

SE CARGA POTENCIA MAXIMA DE LAS RESISTENCIAS DE ACUERDO AL ITEM RADIOGROUP QUE SELECCIONE EL USUARIO.

//Se controla que ningún valor sea nulo

```
if((R1!=0)&&(R2!=0))
```

```
{
```

//instrucciones se ejecutan si condición se cumple

//Preguntamos si el circuito elegido es serie

```
if(RadioButton1->Checked == true) SI SE ELIGIO SERIE
```

```
{
```

//instrucciones se ejecutan si condición se cumple

```
Panel1->Visible=true;
```

```
Panel2->Visible=false;
```

```
RTS=R1+R2;
```

```
I=V/RTS;
```

```
PR1=R1*pow(double(I),double(2)); POW ES LA FUNCIÓN QUE CÁCULA POTENCIA
```

```
PR2=R2*pow(double(I),double(2));
```

```
Edit3->Text=FormatFloat("#0.###",RTS);
```

```
Edit7->Text=FormatFloat("#0.#####",I);
```

```
Edit6->Text=FormatFloat("#0.#####",PR1);
```

```
Edit8->Text=FormatFloat("#0.#####",PR2);
```

```
Memo1->Lines->Clear();
```

```
Memo1->Lines->Add("Cálculo realizado");
```

```
if(PR1>PMAX) SI HABIENDO ELEGIDO SERIE Y
SI POTENCIA DE R1 SUPERA LA MAXIMA DECLARADA PMAX
```

```
{
```

//instrucciones se ejecutan si condición se cumple

```
Memo1->Lines->Add("Potencia de R1 excedida.");
```

```
Image5->Visible=true;
```

```
}
```

```
if(PR2>PMAX) SI HABIENDO ELEGIDO SERIE Y
SI POTENCIA DE R2 SUPERA LA MAXIMA DECLARADA PMAX
```

```
{
```

```

//instrucciones se ejecutan si condición se cumple
Memo1->Lines->Add("Potencia de R2 excedida.");
Image6->Visible=true;

}

}

```

//Preguntamos si el circuito elegido es paralelo

if(RadioButton2->Checked == true) **SI SE ELIGIO PARALELO**

```

{
RTP= 1/((1/R1)+(1/R2));

I=V/RTP;
I1=V/R1;
I2=V/R2;

PR1=R1*pow(double(I1),double(2) );
PR2=R2*pow(double(I2),double(2) );

Edit9->Text=FormatFloat("#0.#####",I);
Edit10->Text=FormatFloat("#0.#####",I1);
Edit11->Text=FormatFloat("#0.#####",I2);

Edit12->Text=FormatFloat("#0.#####",PR1);
Edit13->Text=FormatFloat("#0.#####",PR2);

Edit4->Text=FormatFloat("#0.###",RTP);
Memo1->Lines->Clear();
Memo1->Lines->Add("Cálculo realizado");

```

if(PR1>P_{MAX}) **SI HABIENDO ELEGIDO PARALELO Y SI POTENCIA DE R1 SUPERA LA MAXIMA DECLARADA P_{MAX}**

```

{

//instrucciones se ejecutan si condición se cumple
Memo1->Lines->Add("Potencia de R1 excedida.");
Image7->Visible=true;

}

```

if(PR2>P_{MAX}) **SI HABIENDO ELEGIDO PARALELO Y SI POTENCIA DE R2 SUPERA LA MAXIMA DECLARADA P_{MAX}**

```

{

//instrucciones se ejecutan si condición se cumple
Memo1->Lines->Add("Potencia de R2 excedida.");
Image8->Visible=true;

}

}

}

```

```

else
{

//instrucciones se ejecutan si condición NO se cumple

Memo1->Lines->Clear();
Memo1->Lines->Add("Advertencia: Los valores no pueden ser nulos");

//Revisamos si se debe colocar en rojo algun Edit
if(R1==0)
    Edit1->Color=clRed;

if(R2==0)
    Edit2->Color=clRed;

}

```

} AQUI ES DONDE FINALIZA ACCIONES BOTÓN CALCULAR

```

//-----
void __fastcall TForm1::Salir1Click(TObject *Sender)
{
    Application->Terminate(); SALIR DEL PROGRAMA – MENU PALABRA SALIR
}
//-----
void __fastcall TForm1::Acercade1Click(TObject *Sender)
{
    MessageDlg("(C) Prof: Bolaños D.", mtInformation, TMsgDlgButtons() << mbOK, 0);
    MENSAJE EMERGENTE
}
//-----
void __fastcall TForm1::t1Click(TObject *Sender)
{
    ShellExecute(NULL,"open","RESISTENCIAS.pdf",NULL,"",SW_SHOWNORMAL);
    ABRE EL PDF – MENU PALABRA TEORIA
}
//-----

```

```

void __fastcall TForm1::RadioButton1Click(TObject *Sender)
{
    Panel1->Visible=true; HACE VISIBLE EL PANEL1
    Panel2->Visible=false; HACE INVISIBLE EL PANEL2
    Image4->Visible=false; HACE INVISIBLE EL ROBOT
    //Fijamos item 1 del RadioGroup
    RadioGroup1->ItemIndex=0; SE ELIGE POTENCIA 1/8 EN RADIOGROUP
}
//-----

```

```

void __fastcall TForm1::RadioButton2Click(TObject *Sender)
{
    Panel2->Visible=true; HACE VISIBLE EL PANEL2
    Panel1->Visible=false; HACE INVISIBLE EL PANEL1
    Image4->Visible=false; HACE INVISIBLE EL ROBOT

    //Fijamos item 1 del RadioGroup
    RadioGroup1->ItemIndex=0; SE ELIGE POTENCIA 1/8 EN RADIOGROUP
}
//-----

```

FIN DEL PROGRAMA