```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                      COMPENSACIÓN DE SISTEMAS ELÉCTRICOS DE POTENCIA
                                 Autor: Diego Villegas Govea
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function varargout = Compensacion(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Compensacion_OpeningFcn, ...
                   'gui_OutputFcn',  @Compensacion_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
function Compensacion_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;

guidata(hObject, handles);
function varargout = Compensacion_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function Calcular_Callback(hObject, eventdata, handles)
clc

set (handles.listaresultados, 'String', '');
fprintf('Cálculo de un Sistema Eléctrico de AC\n\n');
disp('+---------------+');
disp('|               |');
disp('|               R1');
disp('|               |');
disp('V1              |');
    circuito=get(handles.menucircuitos,'Value');
    switch circuito
        case 1
            disp('|               L1');
        case 2
            disp('|               C1');
    end
disp('|               |');
disp('|               |');
disp('+---------------+');
fprintf('\n');



%PARÁMETROS DE ENTRADA
if (isnan(str2double(get(handles.f,'String'))) ||
isnan(str2double(get(handles.V1,'String'))) ||
isnan(str2double(get(handles.angV1,'String'))) ||
isnan(str2double(get(handles.R1,'String'))) ||
isnan(str2double(get(handles.Xvalor,'String'))))
    errordlg('Error algún dato inválido  o vacío','Error')
else
    f=eval(get(handles.f,'String'));

    V1=eval(get(handles.V1,'String'));
    angV1=eval(get(handles.angV1,'String'));

    R1=eval(get(handles.R1,'String'));

   %
    inductor=get(handles.Xreac,'value');
    reactancia=get(handles.Xvi,'value');
    if(inductor==1)
        Xreac=eval(get(handles.Xvalor,'String'));
    elseif (reactancia==1)
        Xvi=eval(get(handles.Xvalor,'String'));
    end

    fprintf('\n');
    disp('+---------------+');
    disp('|               |');
```

```
disp('|                 |');
disp('|                 |');
disp('V1                Z1');
disp('|                 |');
disp('|                 |');
disp('|                 |');
disp('+-----------------+');
fprintf('\n');

%IMPEDANCIA
circuito=get(handles.menucircuitos,'Value');
switch circuito
    case 1
        if(inductor==1)
            Z1=complex(R1,Xreac);
        elseif (reactancia==1)
            %Circuito RL
            Xvi=Xvi*10^-3;
            XL=2*pi*f*Xvi;
            Z1=complex(R1,XL);
        end
    case 2
        if(inductor==1)
            Z1=complex(R1,-Xreac);
        elseif (reactancia==1)
            %Circuito RC
            c=Xvi*10^-6;
            Xc=1/(2*pi*f*c);
            Z1=complex(R1,-Xc);
        end
end



[realV1, imagV1]=pol2cart((angV1*pi/180),V1);
disp(['V1 = ' num2str(complex(realV1,imagV1))]);
disp(['V1 = ' num2str(V1) ' |_ ' num2str(angV1) '° V']);
fprintf('\n');
res1=['V1 = ' num2str(complex(realV1,imagV1))];
res2=['V1 = ' num2str(V1) ' |_ ' num2str(angV1) '° V'];

disp(['Z1 = ' num2str(Z1)]);
res3=['Z1 = ' num2str(Z1)];
[angZ1,magZ1] = cart2pol(real(Z1),imag(Z1));
angZ1=angZ1*180/pi;
disp(['Z1 = ' num2str(magZ1) ' |_ ' num2str(angZ1) '° ohms']);
res4=['Z1 = ' num2str(magZ1) ' |_ ' num2str(angZ1) '° ohms'];
fprintf('\n');

disp(['La impedancia equivalente es: ' num2str(Z1)]);
disp(['Zeq = ' num2str(magZ1) ' |_ ' num2str(angZ1) '° ohms']);
res5=['La impedancia equivalente es: ' num2str(Z1)];
res6=['Zeq = ' num2str(magZ1) ' |_ ' num2str(angZ1) '° ohms'];
fprintf('\n');

%CORRIENTE
[realV1, imagV1]=pol2cart((angV1*pi/180),V1);

I1=complex(realV1, imagV1)/Z1;
disp(['I1 = ' num2str(I1)]);
res7=['I1 = ' num2str(I1)];
[angI1,magI1] = cart2pol(real(I1),imag(I1));
angI1=angI1*180/pi;
disp(['I1 = ' num2str(magI1) ' |_ ' num2str(angI1) '° A']);
res8=['I1 = ' num2str(magI1) ' |_ ' num2str(angI1) '° A'];
fprintf('\n');

%POTENCIA APARENTE
S=V1*magI1;
disp(['La potencia aparente es: ' num2str(S) ' VA']);
res9=['La potencia aparente es: ' num2str(S) ' VA'];

%POTENCIA ACTIVA
P=S*cos(angI1*pi/180);
disp(['La potencia activa es: ' num2str(P) ' W']);
res10=['La potencia activa es: ' num2str(P) ' W'];

%POTENCIA REACTIVA
Q=S*sin(angI1*pi/180);
if Q<0
    disp(['La potencia reactiva es: ' num2str(-Q) ' VAR (atraso)']);
```

```
        res11=['La potencia reactiva es: ' num2str(-Q) ' VAR (atraso)'];
    else
        disp(['La potencia reactiva es: ' num2str(Q) ' VAR']);
        res11=['La potencia reactiva es: ' num2str(Q) ' VAR'];
    end
    fprintf('\n');

    %FACTOR DE POTENCIA
    FP=P/S;
    disp(['El factor de potencia es: ' num2str(FP)]);
    res12=['El factor de potencia es: ' num2str(FP)];

    %SEÑALES SENOIDALES
    w=2*pi*f;
    if(f==0)
        f=60;
        t=0:0.0001:(1/f)*5;
    else
        t=0:0.0001:(1/f)*5;
    end



    set(handles.angI1,'String',num2str(angI1));
    set(handles.magI1,'String',num2str(magI1));

    Vm=V1*cos(w*t+angV1*pi/180);

    Im=magI1*cos(w*t+angI1*pi/180);



    %GRÁFICOS
    %Gráfica de comparación de voltaje vs corriente
    [hAx,hLine1,hLine2] = plotyy(handles.grafica,t,Vm,t,Im,'plot');
    title('Sistema Eléctrico No Compensado');
    xlabel('Tiempo');
    ylabel(hAx(1),'Voltaje (V)');
    ylabel(hAx(2),'Corriente (A)')
    grid(handles.grafica, 'on')
    V=['V = ' num2str(V1) ' V'];
    I=['I = ' num2str(round(magI1,2)) ' A'];
    set(hLine1, 'Color', [21/255, 67/255, 96/255]);
    set(hLine1, 'LineWidth',2);
    set(hLine2, 'Color', [129/255, 0, 0]);
    set(hLine2, 'LineWidth',2);
    set(hAx, {'ycolor'}, {[21/255, 67/255, 96/255]; [129/255, 0, 0]})
    legend(V, I);

    %Gráfica de potencias
    x=[0 100];
    yS=[S S];
    plot(handles.aparente,x,yS,'LineWidth',2, 'Color', [255/255, 87/255, 51/255]);
    title(handles.aparente,'Potencia Aparente'); xlabel(handles.aparente,'Tiempo');
ylabel(handles.aparente,'VA')
    grid(handles.aparente, 'on')
    Etiq=['S = ' num2str(round(S,2)) ' VA'];
    legend(handles.aparente, Etiq);

    yP=[P P];
    plot(handles.activa,x,yP,'LineWidth',2, 'Color', [250/255, 210/255, 50/255]);
    title(handles.activa,'Potencia Activa'); xlabel(handles.activa,'Tiempo');
ylabel(handles.activa,'W')
    grid(handles.activa, 'on')
    Etiq=['P = ' num2str(round(P,2)) ' W'];
    legend(handles.activa,Etiq);

    yQ=[Q Q];
    plot(handles.reactiva,x,yQ,'LineWidth',2, 'Color', [60/255, 150/255, 0]);
    title(handles.reactiva,'Potencia Reactiva'); xlabel(handles.reactiva,'Tiempo');
ylabel(handles.reactiva,'VAR')
    grid(handles.reactiva, 'on')
    Etiq=['Q = ' num2str(round(Q,2)) ' VAR'];

    legend(handles.reactiva,Etiq);
    %ListBox
    datos1 = {'Sistema Eléctrico No Compensado';'';mat2str(res1);''; mat2str(res2);'';
mat2str(res3);''; mat2str(res4);''; mat2str(res5);''; mat2str(res6);''; mat2str(res7);'';
mat2str(res8);''; mat2str(res9);''; mat2str(res10);''; mat2str(res11);'';
mat2str(res12)};
    set (handles.listaresultados, 'String', datos1);
```

```
    set(handles.barra,'Enable','off');
    set(handles.edit, 'String', '');
end

function f_Callback(hObject, eventdata, handles)

function f_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function V1_Callback(hObject, eventdata, handles)

function V1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function R1_Callback(hObject, eventdata, handles)

function R1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function angV1_Callback(hObject, eventdata, handles)

function angV1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%      SLIDER      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function barra_Callback(hObject, eventdata, handles)



clc

set (handles.listaresultados, 'String', '');
fprintf('Cálculo de un Sistema Eléctrico de AC\n\n');
disp('+---------------+');
disp('|               |');
disp('|               R1');
disp('|               |');
disp('V1              |');
    circuito=get(handles.menucircuitos,'Value');
    switch circuito
        case 1
            disp('|               L1');
        case 2
            disp('|               C1');
    end
disp('|               |');
disp('|               |');
disp('+---------------+');
fprintf('\n');

if (isnan(str2double(get(handles.f,'String'))) ||
isnan(str2double(get(handles.V1,'String'))) ||
isnan(str2double(get(handles.angV1,'String'))) ||
isnan(str2double(get(handles.R1,'String'))) ||
isnan(str2double(get(handles.Xvalor,'String'))))
    errordlg('Error algún dato inválido  o vacío','Error')
else

    %PARÁMETROS DE ENTRADA
    f=eval(get(handles.f,'String'));

    V1=eval(get(handles.V1,'String'));
    angV1=eval(get(handles.angV1,'String'));
```

```
    R1=eval(get(handles.R1,'String'));

    inductor=get(handles.Xreac,'value');
    reactancia=get(handles.Xvi,'value');

    if(inductor==1)
        Xreac=eval(get(handles.Xvalor,'String'));
    elseif (reactancia==1)
        Xvi=eval(get(handles.Xvalor,'String'));
    end

    fprintf('\n');
    disp('+----------------+');
    disp('|                |');
    disp('|                |');
    disp('|                |');
    disp('V1               Z1');
    disp('|                |');
    disp('|                |');
    disp('|                |');
    disp('+----------------+');
    fprintf('\n');


%IMPEDANCIA
    circuito=get(handles.menucircuitos,'Value');
    switch circuito
        case 1
            if(inductor==1)
                Z1=complex(R1,Xreac);
            elseif (reactancia==1)
                %Circuito RL
                Xvi=Xvi*10^-3;
                XL=2*pi*f*Xvi;
                Z1=complex(R1,XL);
            end
        case 2
            if(inductor==1)
                Z1=complex(R1,-Xreac);
            elseif (reactancia==1)
                %Circuito RC
                c=Xvi*10^-6;
                Xc=1/(2*pi*f*c);
                Z1=complex(R1,-Xc);
            end
    end


    [realV1, imagV1]=pol2cart((angV1*pi/180),V1);
    disp(['V1 = ' num2str(complex(realV1,imagV1))]);
    disp(['V1 = ' num2str(V1) ' |_ ' num2str(angV1) '° V']);
    fprintf('\n');
    res1=['V1 = ' num2str(complex(realV1,imagV1))];
    res2=['V1 = ' num2str(V1) ' |_ ' num2str(angV1) '° V'];

    disp(['Z1 = ' num2str(Z1)]);
    res3=['Z1 = ' num2str(Z1)];
    [angZ1,magZ1] = cart2pol(real(Z1),imag(Z1));
    angZ1=angZ1*180/pi;
    disp(['Z1 = ' num2str(magZ1) ' |_ ' num2str(angZ1) '° ohms']);
    res4=['Z1 = ' num2str(magZ1) ' |_ ' num2str(angZ1) '° ohms'];
    fprintf('\n');

    disp(['La impedancia equivalente es: ' num2str(Z1)]);
    disp(['Zeq = ' num2str(magZ1) ' |_ ' num2str(angZ1) '° ohms']);
    res5=['La impedancia equivalente es: ' num2str(Z1)];
    res6=['Zeq = ' num2str(magZ1) ' |_ ' num2str(angZ1) '° ohms'];
    fprintf('\n');

    %CORRIENTE
    [realV1, imagV1]=pol2cart((angV1*pi/180),V1);

    I1=complex(realV1, imagV1)/Z1;
    disp(['I1 = ' num2str(I1)]);
    res7=['I1 = ' num2str(I1)];
    [angI1,magI1] = cart2pol(real(I1),imag(I1));
    angI1=angI1*180/pi;
    disp(['I1 = ' num2str(magI1) ' |_ ' num2str(angI1) '° A']);
    res8=['I1 = ' num2str(magI1) ' |_ ' num2str(angI1) '° A'];
    fprintf('\n');
```

```
    %POTENCIA APARENTE
    S=V1*magI1;
    disp(['La potencia aparente es: ' num2str(S) ' VA']);
    res9=['La potencia aparente es: ' num2str(S) ' VA'];

    %POTENCIA ACTIVA
    P=S*cos(angI1*pi/180);
    disp(['La potencia activa es: ' num2str(P) ' W']);
    res10=['La potencia activa es: ' num2str(P) ' W'];

    %POTENCIA REACTIVA
    Q=S*sin(angI1*pi/180);
    if Q<0
        disp(['La potencia reactiva es: ' num2str(-Q) ' VAR (atraso)']);
        res11=['La potencia reactiva es: ' num2str(-Q) ' VAR (atraso)'];
    else
        disp(['La potencia reactiva es: ' num2str(Q) ' VAR']);
        res11=['La potencia reactiva es: ' num2str(Q) ' VAR'];
    end
    fprintf('\n');

    %FACTOR DE POTENCIA
    FP=P/S;
    disp(['El factor de potencia es: ' num2str(FP)]);
    res12=['El factor de potencia es: ' num2str(FP)];


%%%%%%%%%%%%%%%%%%%%%%%%%%COMPENSACIÓN DEL SISTEMA%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    RES14=['Sistema Eléctrico Compensado'];
    disp('   +------------+-------------+');
    disp('   |            |             |');
    disp(['   |          ' num2str(R1) 'ohm       ' num2str(R1) 'ohm']);
    disp('   |            |             |');
    disp([num2str(V1) '<' num2str(angV1) '°V' '          |             |']);
    inductor=get(handles.Xreac,'value');
%IMPEDANCIA
    circuito=get(handles.menucircuitos,'Value');

    %%%%Barra
    barra=get(handles.barra,'Value');

    switch circuito
        case 1%Circuito RL
            if(inductor==1)%Reactancia
                Xreac=eval(get(handles.Xvalor,'String'));
                C1=-Xreac;
                %%%%%%%%%%%%%%%
                C1=barra*C1/100;
                set(handles.edit,'String',barra);
                %%%%%%%%%%%%%%%
                disp(['   |          '  num2str(round(C1,2)) 'i        ' num2str(Xreac)
'i']);
                Z2=Z1;
                Z1=complex(R1,C1);
                RES33=[''];
            else %Inductancia
                Xvi=eval(get(handles.Xvalor,'String'));
                XL=2*pi*f*Xvi*10^-3;
                XL=(100*XL)/(barra);
                C1=1/(2*pi*f*XL*10^-6);
                %%%%%%%%%%%%%%%
                set(handles.edit,'String',barra);
                %%%%%%%%%%%%%%%
                disp(['   |          ' num2str(round(C1,2)) 'uF        ' num2str(Xvi)
'mH']);
                Z2=Z1;
                Z1=complex(R1,-XL);
                 RES33=['El valor del capacitor para compensar el sistema eléctrico es
de: ' num2str(round(C1,2)) ' uF'];
            end
        case 2 %Circuito RC
            if(inductor==1)%Reactancia
                Xreac=eval(get(handles.Xvalor,'String'));
                C1=-Xreac;
                %%%%%%%%%%%%%%%
                C1=barra*C1/100;
                set(handles.edit,'String',barra);
                %%%%%%%%%%%%%%%
```

```
                disp(['   |         '  num2str(round(-C1,2)) 'i        ' num2str(-Xreac)
'i']);
                Z2=Z1;
                Z1=complex(R1,-C1);
                RES33=[''];
            else%Capacitancia
                %Circuito RC
                c=eval(get(handles.Xvalor,'String'));
                Xc=1/(2*pi*f*c*10^-6);
                L=Xc/(2*pi*f);
                %%%%%%%%%%%%%%
                L=barra*L/100;
                set(handles.edit,'String',barra);
                %%%%%%%%%%%%%%
                XL=(2*pi*f*L);

                Z1=complex(R1,XL);
                Z2=complex(R1,-XL);
                disp(['   |         ' num2str(round(L*10^3,2)) ' mH        ' num2str(Xvi)
' uF']);
                RES33=['El valor del inductor para compensar el sistema eléctrico es de:
' num2str(round(L*10^3,2)) ' mH'];
            end
    end


    disp('   |            |             |');
    disp('   |            |             |');
    disp('   +------------+-------------+');
    fprintf('\n');
    disp('+-------+-------+');
    disp('|       |       |');
    disp('|       |       |');
    disp('V1      Z1      Z2');
    disp('|       |       |');
    disp('|       |       |');
    disp('+-------+-------+');
    fprintf('\n');

    [realV1, imagV1]=pol2cart((angV1*pi/180),V1);
    disp(['V1 = ' num2str(complex(realV1,imagV1))]);
    RES17=['V1 = ' num2str(complex(realV1,imagV1))];
    disp(['V1 = ' num2str(V1) ' |_ ' num2str(angV1) '° V']);
    RES18=['V1 = ' num2str(V1) ' |_ ' num2str(angV1) '° V'];
    fprintf('\n');

    disp(['Z1 = ' num2str(Z1)]);
    RES19=['Z1 = ' num2str(Z1)];
    [angZ1,magZ1] = cart2pol(real(Z1),imag(Z1));
    angZ1=angZ1*180/pi;
    disp(['Z1 = ' num2str(magZ1) ' |_ ' num2str(angZ1) '° ohms']);
    RES20=['Z1 = ' num2str(magZ1) ' |_ ' num2str(angZ1) '° ohms'];
    fprintf('\n');

    disp(['Z2 = ' num2str(Z2)]);
    RES21=['Z2 = ' num2str(Z2)];
    [angZ2,magZ2] = cart2pol(real(Z2),imag(Z2));
    angZ2=angZ2*180/pi;
    disp(['Z2 = ' num2str(magZ2) ' |_ ' num2str(angZ2) '° ohms']);
    RES22=['Z2 = ' num2str(magZ2) ' |_ ' num2str(angZ2) '° ohms'];
    fprintf('\n');

    disp('+-------------+');
    disp('|             |');
    disp('|             |');
    disp('V1            Z3');
    disp('|             |');
    disp('|             |');
    disp('+-------------+');
    fprintf('\n');

    %Z3=Z1||Z2
    Z3=(Z1*Z2)/(Z1+Z2);

    disp(['Z3 = ' num2str(Z3)]);
    RES23=['Z3 = ' num2str(Z3)];
    [angZ3,magZ3] = cart2pol(real(Z3),imag(Z3));
    angZ3=angZ3*180/pi;
    disp(['Z3 = ' num2str(magZ3) ' |_ ' num2str(angZ3) '° ohms']);
    RES24=['Z3 = ' num2str(magZ3) ' |_ ' num2str(angZ3) '° ohms'];
```

```
    fprintf('\n');

    disp(['La impedancia equivalente es: ' num2str(Z3)]);
    RES25=['La impedancia equivalente es: ' num2str(Z3)];
    disp(['Zeq = ' num2str(magZ3) ' |_ ' num2str(angZ3) '° ohms']);
    RES26=['Zeq = ' num2str(magZ3) ' |_ ' num2str(angZ3) '° ohms'];
    fprintf('\n');

    %Corriente
    [realV1, imagV1]=pol2cart((angV1*pi/180),V1);

    I1=complex(realV1, imagV1)/Z3;
    disp(['I1 = ' num2str(I1)]);
    RES27=['I1 = ' num2str(I1)];
    [angI1,magI1] = cart2pol(real(I1),imag(I1));
    angI1=angI1*180/pi;
    disp(['I1 = ' num2str(magI1) ' |_ ' num2str(angI1) '° A']);
    RES28=['I1 = ' num2str(magI1) ' |_ ' num2str(angI1) '° A'];
    fprintf('\n');

    %POTENCIA APARENTE
    S=V1*magI1;
    disp(['La potencia aparente es: ' num2str(S) ' VA']);
    RES29=['La potencia aparente es: ' num2str(S) ' VA'];

    %POTENCIA ACTIVA
    P=S*cos(angI1*pi/180);
    disp(['La potencia activa es: ' num2str(P) ' W']);
    RES30=['La potencia activa es: ' num2str(P) ' W'];

    %POTENCIA REACTIVA
    Q=S*sin(angI1*pi/180);
    if Q<0
        disp(['La potencia reactiva es: ' num2str(-Q) ' VAR']);
        RES31=['La potencia reactiva es: ' num2str(-Q) ' VAR'];
    else
        disp(['La potencia reactiva es: ' num2str(Q) ' VAR']);
        RES31=['La potencia reactiva es: ' num2str(Q) ' VAR'];
    end
    fprintf('\n');

    %FACTOR DE POTENCIA
    FP=P/S;
    disp(['El factor de potencia es: ' num2str(FP)]);
    RES32=['El factor de potencia es: ' num2str(FP)];

    %ListBox
    datos = {'Sistema Eléctrico No Compensado';'';mat2str(res1);''; mat2str(res2);'';
mat2str(res3);''; mat2str(res4);''; mat2str(res5);''; mat2str(res6);'';
        mat2str(res7);''; mat2str(res8);''; mat2str(res9);''; mat2str(res10);'';
mat2str(res11);''; mat2str(res12);'';
        mat2str(RES14);''; mat2str(RES17);''; mat2str(RES18);''; mat2str(RES19);'';
        mat2str(RES20);''; mat2str(RES21);''; mat2str(RES22);''; mat2str(RES23);'';
mat2str(RES24);''; mat2str(RES25);'';
        mat2str(RES26);''; mat2str(RES27);''; mat2str(RES28);''; mat2str(RES29);'';
mat2str(RES30);''; mat2str(RES31);
        ''; mat2str(RES32); ''; mat2str(RES33)};
    set (handles.listaresultados, 'String', datos);

    %SEÑALES SENOIDALES
    w=2*pi*f;
    t=0:0.0001:(1/f)*5;

    Vm=V1*cos(w*t+angV1*pi/180);
    Im=magI1*cos(w*t+angI1*pi/180);

    %GRÁFICOS
    %Gráfica de comparación de voltaje vs corriente
    [hAx,hLine1,hLine2] = plotyy(handles.grafica,t,Vm,t,Im,'plot');
    title('Sistema Eléctrico Compensado');
    xlabel('Tiempo');
    ylabel(hAx(1),'Voltaje (V)');
    ylabel(hAx(2),'Corriente (A)')
    grid(handles.grafica, 'on')
    V=['V = ' num2str(V1) ' V'];
    I=['I = ' num2str(round(magI1,2)) ' A'];
    set(hLine1, 'Color', [21/255, 67/255, 96/255]);
    set(hLine1, 'LineWidth',2);
    set(hLine2, 'Color', [129/255, 0, 0]);
    set(hLine2, 'LineWidth',2);
```

```
    set(hAx, {'ycolor'}, {[21/255, 67/255, 96/255]; [129/255, 0, 0]})
    legend(V, I);

    %Gráfica de potencias
    x=[0 100];
    yS=[S S];
    plot(handles.aparente,x,yS,'LineWidth',2, 'Color', [255/255, 87/255, 51/255]);
    title(handles.aparente,'Potencia Aparente'); xlabel(handles.aparente,'Tiempo');
ylabel(handles.aparente,'VA')
    grid(handles.aparente, 'on')
    Etiq=['S = ' num2str(round(S,2)) ' VA'];
    legend(handles.aparente, Etiq);

    yP=[P P];
    plot(handles.activa,x,yP,'LineWidth',2, 'Color', [250/255, 210/255, 50/255]);
    title(handles.activa,'Potencia Activa'); xlabel(handles.activa,'Tiempo');
ylabel(handles.activa,'W')
    grid(handles.activa, 'on')
    Etiq=['P = ' num2str(round(P,2)) ' W'];
    legend(handles.activa,Etiq);

    yQ=[Q Q];
    plot(handles.reactiva,x,yQ,'LineWidth',2, 'Color', [60/255, 150/255, 0]);
    title(handles.reactiva,'Potencia Reactiva'); xlabel(handles.reactiva,'Tiempo');
ylabel(handles.reactiva,'VAR')
    grid(handles.reactiva, 'on')
    Etiq=['Q = ' num2str(round(Q,2)) ' VAR'];
    legend(handles.reactiva,Etiq);

    set(handles.barra,'Enable','on');
end

function barra_CreateFcn(hObject, eventdata, handles)

if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

function Xvalor_Callback(hObject, eventdata, handles)

function Xvalor_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%%%%%%%%%%%%%%%%%%%%%%%%% Limpiar Campos %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Limpiar_Callback(hObject, eventdata, handles)

set(handles.f,'String','');
set(handles.V1,'String','');
set(handles.angV1,'String','');
set(handles.R1,'String','');
set(handles.Xvalor,'String','');
set (handles.listaresultados, 'String','');

set(handles.barra,'Enable','off');
set(handles.barra,'Value',100);
set(handles.edit, 'String', '');

cla(handles.grafica,'reset');
cla(handles.aparente,'reset');
cla(handles.activa,'reset');
cla(handles.reactiva,'reset');


function Xvi_Callback(hObject, eventdata, handles)

function Xreac_Callback(hObject, eventdata, handles)

function result_Callback(hObject, eventdata, handles)

function result_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor','white');
end

function listaresultados_Callback(hObject, eventdata, handles)

function listaresultados_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%%%%%%%%%%%%%%%%%%%%%%% Boton Compensar %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function pushbuttonCompensar_Callback(hObject, eventdata, handles)

clc
set(handles.barra,'Enable','off');
set(handles.barra,'Value',100);
set(handles.edit, 'String', '');
maxSliderValue = get(handles.barra, 'Max');
minSliderValue = get(handles.barra, 'Min');
theRange = maxSliderValue - minSliderValue;
steps = [10/theRange, 100/theRange];
set(handles.barra, 'SliderStep', steps);

set (handles.listaresultados, 'String', '');
fprintf('Cálculo de un Sistema Eléctrico de AC\n\n');
disp('+---------------+');
disp('|               |');
disp('|              R1');
disp('|               |');
disp('V1              |');
    circuito=get(handles.menucircuitos,'Value');
    switch circuito
        case 1
            disp('|               L1');
        case 2
            disp('|               C1');
    end
disp('|               |');
disp('|               |');
disp('+---------------+');
fprintf('\n');

if (isnan(str2double(get(handles.f,'String'))) ||
isnan(str2double(get(handles.V1,'String'))) ||
isnan(str2double(get(handles.angV1,'String'))) ||
isnan(str2double(get(handles.R1,'String'))) ||
isnan(str2double(get(handles.Xvalor,'String'))))
    errordlg('Error algún dato inválido  o vacío','Error')
else

    %PARÁMETROS DE ENTRADA
    f=eval(get(handles.f,'String'));

    V1=eval(get(handles.V1,'String'));
    angV1=eval(get(handles.angV1,'String'));

    R1=eval(get(handles.R1,'String'));

    inductor=get(handles.Xreac,'value');
    reactancia=get(handles.Xvi,'value');

    if(inductor==1)
        Xreac=eval(get(handles.Xvalor,'String'));
    elseif (reactancia==1)
        Xvi=eval(get(handles.Xvalor,'String'));
    end

    fprintf('\n');
    disp('+----------------+');
    disp('|                |');
    disp('|                |');
    disp('|                |');
    disp('V1              Z1');
    disp('|                |');
    disp('|                |');
    disp('|                |');
    disp('+----------------+');
    fprintf('\n');
```

```
%IMPEDANCIA
    circuito=get(handles.menucircuitos,'Value');
    switch circuito
        case 1
            if(inductor==1)
                Z1=complex(R1,Xreac);
            elseif (reactancia==1)
                %Circuito RL
                Xvi=Xvi*10^-3;
                XL=2*pi*f*Xvi;
                Z1=complex(R1,XL);
            end
        case 2
            if(inductor==1)
                Z1=complex(R1,-Xreac);
            elseif (reactancia==1)
                %Circuito RC
                c=Xvi*10^-6;
                Xc=1/(2*pi*f*c);
                Z1=complex(R1,-Xc);
            end
    end

    [realV1, imagV1]=pol2cart((angV1*pi/180),V1);
    disp(['V1 = ' num2str(complex(realV1,imagV1))]);
    disp(['V1 = ' num2str(V1) ' |_ ' num2str(angV1) '° V']);
    fprintf('\n');
    res1=['V1 = ' num2str(complex(realV1,imagV1))];
    res2=['V1 = ' num2str(V1) ' |_ ' num2str(angV1) '° V'];

    disp(['Z1 = ' num2str(Z1)]);
    res3=['Z1 = ' num2str(Z1)];
    [angZ1,magZ1] = cart2pol(real(Z1),imag(Z1));
    angZ1=angZ1*180/pi;
    disp(['Z1 = ' num2str(magZ1) ' |_ ' num2str(angZ1) '° ohms']);
    res4=['Z1 = ' num2str(magZ1) ' |_ ' num2str(angZ1) '° ohms'];
    fprintf('\n');

    disp(['La impedancia equivalente es: ' num2str(Z1)]);
    disp(['Zeq = ' num2str(magZ1) ' |_ ' num2str(angZ1) '° ohms']);
    res5=['La impedancia equivalente es: ' num2str(Z1)];
    res6=['Zeq = ' num2str(magZ1) ' |_ ' num2str(angZ1) '° ohms'];
    fprintf('\n');

    %CORRIENTE
    [realV1, imagV1]=pol2cart((angV1*pi/180),V1);

    I1=complex(realV1, imagV1)/Z1;
    disp(['I1 = ' num2str(I1)]);
    res7=['I1 = ' num2str(I1)];
    [angI1,magI1] = cart2pol(real(I1),imag(I1));
    angI1=angI1*180/pi;
    disp(['I1 = ' num2str(magI1) ' |_ ' num2str(angI1) '° A']);
    res8=['I1 = ' num2str(magI1) ' |_ ' num2str(angI1) '° A'];
    fprintf('\n');

    %POTENCIA APARENTE
    S=V1*magI1;
    disp(['La potencia aparente es: ' num2str(S) ' VA']);
    res9=['La potencia aparente es: ' num2str(S) ' VA'];

    %POTENCIA ACTIVA
    P=S*cos(angI1*pi/180);
    disp(['La potencia activa es: ' num2str(P) ' W']);
    res10=['La potencia activa es: ' num2str(P) ' W'];

    %POTENCIA REACTIVA
    Q=S*sin(angI1*pi/180);
    if Q<0
        disp(['La potencia reactiva es: ' num2str(-Q) ' VAR (atraso)']);
        res11=['La potencia reactiva es: ' num2str(-Q) ' VAR (atraso)'];
    else
        disp(['La potencia reactiva es: ' num2str(Q) ' VAR']);
        res11=['La potencia reactiva es: ' num2str(Q) ' VAR'];
    end
    fprintf('\n');

    %FACTOR DE POTENCIA
    FP=P/S;
```

```
    disp(['El factor de potencia es: ' num2str(FP)]);
    res12=['El factor de potencia es: ' num2str(FP)];


%%%%%%%%%%%%%%%%%%%%%%%%%%COMPENSACIÓN DEL SISTEMA%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    RES14=['Sistema Eléctrico Compensado'];
    disp('   +------------+-------------+');
    disp('   |            |             |');
    disp(['   |          ' num2str(R1) 'ohm       ' num2str(R1) 'ohm']);
    disp('   |            |             |');
    disp([num2str(V1) '<' num2str(angV1) '°V' '         |             |']);
    inductor=get(handles.Xreac,'value');

%IMPEDANCIA
    circuito=get(handles.menucircuitos,'Value');
    switch circuito
        case 1
            if(inductor==1)
                Xreac=eval(get(handles.Xvalor,'String'));
                C1=-Xreac;
                %set(handles.edit,'String',C1);
                disp(['   |         '  num2str(round(C1,2)) 'i        ' num2str(Xreac)
'i']);
                Z2=Z1;
                Z1=complex(R1,C1);
                RES33=[''];
            else
                Xvi=eval(get(handles.Xvalor,'String'));
                XL=2*pi*f*Xvi*10^-3;
                C1=1/(2*pi*f*XL*10^-6);
                %set(handles.edit,'String',C1);
                disp(['   |         ' num2str(round(C1,2)) 'uF        ' num2str(Xvi)
'mH']);
                Z2=Z1;
                Z1=complex(R1,-XL);
                 RES33=['El valor del capacitor para compensar el sistema eléctrico es
de: ' num2str(round(C1,2)) ' uF'];
            end
        case 2
            if(inductor==1)
                Xreac=eval(get(handles.Xvalor,'String'));
                C1=-Xreac;
                %set(handles.edit,'String',C1);
                disp(['   |         '  num2str(round(-C1,2)) 'i        ' num2str(-Xreac)
'i']);
                Z2=Z1;
                Z1=complex(R1,-C1);
                RES33=[''];
            else
                %Circuito RC
                c=eval(get(handles.Xvalor,'String'));
                Xc=1/(2*pi*f*c*10^-6);
                L=Xc/(2*pi*f);
                %set(handles.edit,'String',L);
                XL=(2*pi*f*L);

                Z1=complex(R1,XL);
                Z2=complex(R1,-XL);
                disp(['   |         ' num2str(round(L*10^3,2)) ' mH        ' num2str(Xvi)
' uF']);
                RES33=['El valor del inductor para compensar el sistema eléctrico es de:
' num2str(round(L*10^3,2)) ' mH'];
            end
    end


    disp('   |            |             |');
    disp('   |            |             |');
    disp('   +------------+-------------+');
    fprintf('\n');
    disp('+-------+-------+');
    disp('|       |       |');
    disp('|       |       |');
    disp('V1      Z1      Z2');
    disp('|       |       |');
    disp('|       |       |');
    disp('+-------+-------+');
    fprintf('\n');

    [realV1, imagV1]=pol2cart((angV1*pi/180),V1);
```

```
disp(['V1 = ' num2str(complex(realV1,imagV1))]);
RES17=['V1 = ' num2str(complex(realV1,imagV1))];
disp(['V1 = ' num2str(V1) ' |_ ' num2str(angV1) '° V']);
RES18=['V1 = ' num2str(V1) ' |_ ' num2str(angV1) '° V'];
fprintf('\n');

disp(['Z1 = ' num2str(Z1)]);
RES19=['Z1 = ' num2str(Z1)];
[angZ1,magZ1] = cart2pol(real(Z1),imag(Z1));
angZ1=angZ1*180/pi;
disp(['Z1 = ' num2str(magZ1) ' |_ ' num2str(angZ1) '° ohms']);
RES20=['Z1 = ' num2str(magZ1) ' |_ ' num2str(angZ1) '° ohms'];
fprintf('\n');

disp(['Z2 = ' num2str(Z2)]);
RES21=['Z2 = ' num2str(Z2)];
[angZ2,magZ2] = cart2pol(real(Z2),imag(Z2));
angZ2=angZ2*180/pi;
disp(['Z2 = ' num2str(magZ2) ' |_ ' num2str(angZ2) '° ohms']);
RES22=['Z2 = ' num2str(magZ2) ' |_ ' num2str(angZ2) '° ohms'];
fprintf('\n');

disp('+-------------+');
disp('|             |');
disp('|             |');
disp('V1            Z3');
disp('|             |');
disp('|             |');
disp('+-------------+');
fprintf('\n');

%Z3=Z1||Z2
Z3=(Z1*Z2)/(Z1+Z2);

disp(['Z3 = ' num2str(Z3)]);
RES23=['Z3 = ' num2str(Z3)];
[angZ3,magZ3] = cart2pol(real(Z3),imag(Z3));
angZ3=angZ3*180/pi;
disp(['Z3 = ' num2str(magZ3) ' |_ ' num2str(angZ3) '° ohms']);
RES24=['Z3 = ' num2str(magZ3) ' |_ ' num2str(angZ3) '° ohms'];
fprintf('\n');

disp(['La impedancia equivalente es: ' num2str(Z3)]);
RES25=['La impedancia equivalente es: ' num2str(Z3)];
disp(['Zeq = ' num2str(magZ3) ' |_ ' num2str(angZ3) '° ohms']);
RES26=['Zeq = ' num2str(magZ3) ' |_ ' num2str(angZ3) '° ohms'];
fprintf('\n');

%Corriente
[realV1, imagV1]=pol2cart((angV1*pi/180),V1);

I1=complex(realV1, imagV1)/Z3;
disp(['I1 = ' num2str(I1)]);
RES27=['I1 = ' num2str(I1)];
[angI1,magI1] = cart2pol(real(I1),imag(I1));
angI1=angI1*180/pi;
disp(['I1 = ' num2str(magI1) ' |_ ' num2str(angI1) '° A']);
RES28=['I1 = ' num2str(magI1) ' |_ ' num2str(angI1) '° A'];
fprintf('\n');

%POTENCIA APARENTE
S=V1*magI1;
disp(['La potencia aparente es: ' num2str(S) ' VA']);
RES29=['La potencia aparente es: ' num2str(S) ' VA'];

%POTENCIA ACTIVA
P=S*cos(angI1*pi/180);
disp(['La potencia activa es: ' num2str(P) ' W']);
RES30=['La potencia activa es: ' num2str(P) ' W'];

%POTENCIA REACTIVA
Q=S*sin(angI1*pi/180);
if Q<0
    disp(['La potencia reactiva es: ' num2str(-Q) ' VAR']);
    RES31=['La potencia reactiva es: ' num2str(-Q) ' VAR'];
else
    disp(['La potencia reactiva es: ' num2str(Q) ' VAR']);
    RES31=['La potencia reactiva es: ' num2str(Q) ' VAR'];
end
fprintf('\n');
```

```
    %FACTOR DE POTENCIA
    FP=P/S;
    disp(['El factor de potencia es: ' num2str(FP)]);
    RES32=['El factor de potencia es: ' num2str(FP)];

    %ListBox
    datos = {'Sistema Eléctrico No Compensado';'';mat2str(res1);''; mat2str(res2);'';
mat2str(res3);''; mat2str(res4);''; mat2str(res5);''; mat2str(res6);'';
        mat2str(res7);''; mat2str(res8);''; mat2str(res9);''; mat2str(res10);'';
mat2str(res11);''; mat2str(res12);'';
        mat2str(RES14);''; mat2str(RES17);''; mat2str(RES18);''; mat2str(RES19);'';
        mat2str(RES20);''; mat2str(RES21);''; mat2str(RES22);''; mat2str(RES23);'';
mat2str(RES24);''; mat2str(RES25);'';
        mat2str(RES26);''; mat2str(RES27);''; mat2str(RES28);''; mat2str(RES29);'';
mat2str(RES30);''; mat2str(RES31);
        ''; mat2str(RES32); ''; mat2str(RES33)};
    set (handles.listaresultados, 'String', datos);


    %SEÑALES SENOIDALES
    w=2*pi*f;
    t=0:0.0001:(1/f)*5;

    Vm=V1*cos(w*t+angV1*pi/180);
    Im=magI1*cos(w*t+angI1*pi/180);

    %GRÁFICOS
    %Gráfica de comparación de voltaje vs corriente
    [hAx,hLine1,hLine2] = plotyy(handles.grafica,t,Vm,t,Im,'plot');
    title('Sistema Eléctrico Compensado');
    xlabel('Tiempo');
    ylabel(hAx(1),'Voltaje (V)');
    ylabel(hAx(2),'Corriente (A)')
    grid(handles.grafica, 'on')
    V=['V = ' num2str(V1) ' V'];
    I=['I = ' num2str(round(magI1,2)) ' A'];
    set(hLine1, 'Color', [21/255, 67/255, 96/255]);
    set(hLine1, 'LineWidth',2);
    set(hLine2, 'Color', [129/255, 0, 0]);
    set(hLine2, 'LineWidth',2);
    set(hAx, {'ycolor'}, {[21/255, 67/255, 96/255]; [129/255, 0, 0]})
    legend(V, I);

    %Gráfica de potencias
    x=[0 100];
    yS=[S S];
    plot(handles.aparente,x,yS,'LineWidth',2, 'Color', [255/255, 87/255, 51/255]);
    title(handles.aparente,'Potencia Aparente'); xlabel(handles.aparente,'Tiempo');
ylabel(handles.aparente,'VA')
    grid(handles.aparente, 'on')
    Etiq=['S = ' num2str(round(S,2)) ' VA'];
    legend(handles.aparente, Etiq);

    yP=[P P];
    plot(handles.activa,x,yP,'LineWidth',2, 'Color', [250/255, 210/255, 50/255]);
    title(handles.activa,'Potencia Activa'); xlabel(handles.activa,'Tiempo');
ylabel(handles.activa,'W')
    grid(handles.activa, 'on')
    Etiq=['P = ' num2str(round(P,2)) ' W'];
    legend(handles.activa,Etiq);

    yQ=[Q Q];
    plot(handles.reactiva,x,yQ,'LineWidth',2, 'Color', [60/255, 150/255, 0]);
    title(handles.reactiva,'Potencia Reactiva'); xlabel(handles.reactiva,'Tiempo');
ylabel(handles.reactiva,'VAR')
    grid(handles.reactiva, 'on')
    Etiq=['Q = ' num2str(round(Q,2)) ' VAR'];
    legend(handles.reactiva,Etiq);

    set(handles.barra,'Enable','on');
end

function edit_Callback(hObject, eventdata, handles)

function edit_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

```
end

function w_Callback(hObject, eventdata, handles)

function w_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function magI1_Callback(hObject, eventdata, handles)

function magI1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function angI1_Callback(hObject, eventdata, handles)

function angI1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Menu de Circuitos %%%%%%%%%%%%%%%%%%%%%%%%%%

function menucircuitos_Callback(hObject, eventdata, handles)

circuito=get(handles.menucircuitos,'Value');
switch circuito
    case 1
        set(handles.elementos,'Title','Inductor:');
        set(handles.Xvi,'String','Inductancia (mH)');
    case 2
        set(handles.elementos,'Title','Capacitor:');
        set(handles.Xvi,'String','Capacitancia (uF)');
end

function menucircuitos_CreateFcn(hObject, eventdata, handles)
% hObject    handle to menucircuitos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function elementos_CreateFcn(hObject, eventdata, handles)
```