```
function varargout = Programa(varargin)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%FERRAMENTA COMPUTACIONAL DE MODELAÇÃO DO DESEMPENHO DE PAINEIS
%FOTOVOLTAICOS / COMPORTAMENTO GERAL
%
%AUTOR: ROMEU SANTOS & PROF. PEDRO DINIS
%CURSO: ENGENHARIA ELECTROMECÂNICA
%
%UNIVERSIDADE DA BEIRA INTERIOR
%DATA: 19/09/2011
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
% PROGRAMA M-file for Programa.fig
%      PROGRAMA, by itself, creates a new PROGRAMA or raises the
existing
%      singleton*.
%
%      H = PROGRAMA returns the handle to a new PROGRAMA or the
handle to
%      the existing singleton*.
%
%      PROGRAMA('CALLBACK',hObject,eventData,handles,...) calls the
local
%      function named CALLBACK in PROGRAMA.M with the given input
arguments.
%
%      PROGRAMA('Property','Value',...) creates a new PROGRAMA or
raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before Programa_OpeningFcn gets called.
An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to Programa_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Programa

% Last Modified by GUIDE v2.5 17-Sep-2011 20:14:42

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Programa_OpeningFcn, ...
                   'gui_OutputFcn',  @Programa_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
% End initialization code - DO NOT EDIT

% --- Executes just before Programa is made visible.
function Programa_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Programa (see VARARGIN)

% Choose default command line output for Programa
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Programa wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Programa_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in Simular_pushbutton.
function Simular_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to Simular_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%%%%%%%%PRIMEIRA
PARTE%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%INPUTS:ENTRADA DOS PARÂMETROS DO FABRICANTE [STC]

%TENSÃO DE CIRCUITO ABERTO NAS CONDIÇÕES STC
Voc_stc = get(handles.Voc_in,'string');
Voc_stc = str2num(Voc_stc);

%CORRENTE DE CURTO-CIRCUITO NAS CONDIÇÕES STC
Isc_stc = get(handles.Isc_in,'string');
Isc_stc = str2num(Isc_stc);

%TENSÃO NO PONTO DE POTÊNCIA MÁXIMA (STC)
Vmp_stc = get(handles.Vmp_in,'string');
Vmp_stc = str2num(Vmp_stc);

%CORRENTE NO PONTO DE POTÊNCIA MÁXIMA (STC)
```

```
Imp_stc = get(handles.Imp_in,'string');
Imp_stc = str2num(Imp_stc);

%COEFICIENTE DE TEMPERATURA DE Isc
Ki = get(handles.ki_in,'string');
Ki = str2num(Ki);

%COEFICIENTE DE TEMPERATURA DE Voc
Kv = get(handles.kv_in,'string');
Kv = str2num(Kv);

%NÚMERO DE CÉLULAS
Ns = get(handles.Nc_in,'string');
Ns = str2num(Ns);
Nc=Ns;

%LARGURA DA CÉLULA
l = get(handles.l_in,'string');
l = str2num(l);

%COMPRIMENTO DA CÉLULA
c = get(handles.c_in,'string');
c = str2num(c);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
if  Voc_stc ==0 | Isc_stc==0 | Vmp_stc==0 | Imp_stc==0 | Ns==0;
errordlg('Introduza pelo menos os valores de Voc, Isc, Vmp, Imp,
N.Células','Erro');
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CONSTANTES
G_stc = 1;           %IRRADIANCIA DE REFERÊNCIA
T_stc = 298.15;      %TEMPERATURA DE REFERÊNCIA
q = 1.60217646e-19;  %CARGA DO ELECTRÃO
k = 1.3806503e-23;   %CONSTANTE DE BOLTZMANN
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%OUTROS FACTORES
a1 = 1;              %FACTOR DE IDEALIDADE DO DIODO 1
a2 = 1.2;            %FACTOR DE IDEALIDADE DO DIODO 2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CONDIÇÕES DE TESTE
T = 298.15;          %TEMPERATURA DO MÓDULO
G = 1;               %IRRADIÂNCIA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%INICIALIZAÇÃO
P_MAX_M = 0;         %INICIALIZAÇÃO DA POTÊNCIA MÁXIMA NUMÉRICA
RS_0 = 0.2;          %INICIALIZAÇÃO DA RESISTÊNCIA SÉRIE
RS = RS_0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%OUTRAS VARIAVEIS
DT = T - T_stc;        %DIFERENÇA ENTRE A TEMP. DO MÓDULO E A TEMP.
NOMINAL
PMAX_E = Vmp_stc .* Imp_stc;     %POTÊNCIA MÁXIMA EXPERIMENTAL
(DATASHEET)
Eps_POT = abs(P_MAX_M - PMAX_E); %CONDIÇÃO PARA CÁLCULO DO ERRO
TOL = 0.001;                     %TOLERÂNCIA
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CÁLCULO DAS TENSÕES TÉRMICAS
Vt_1 = ((a1 .* Ns .* k .* T) ./ q);
Vt_2 = ((a2 .* Ns .* k .* T) ./ q);
Vt = ((Ns .* k .* T) ./ q);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CÁLCULO DA CORRENTE FOTOVOLTAICA
Ipv_stc = Isc_stc;
Ipv = (Ipv_stc + (Ki .* DT)) .* (G ./ G_stc);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%OUTRAS VARIAVEIS PARA O CALCULO DE Eg
T_max = 348;
dt = T_max - T_stc;
Isc_Tmax = Isc_stc + Ki .* dt;
Voc_Tmax = Voc_stc + Kv .* dt;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CÁLCULO DAS CORRENTES DE SATURAÇÃO PARA OS DIODOS 1 E 2
%CALCULO DE EG
I_0_n = ((Isc_stc) ./ (exp((Voc_stc) ./ (Vt)) - 1));

Eg = (- log (((Isc_Tmax ./ I_0_n) .* ((T_stc ./ T_max)^3)) ./ ...
    (exp((q .* Voc_Tmax) ./ (a2 .* Ns .* k .*T_max)) - 1))) .* ...
    ((a2 .* k .* T_stc .* T_max) ./ ((q .* (T_stc - T_max))));

I_0 = (I_0_n) .* ((T_stc ./ T) ^ 3) .* (exp (((q .* Eg) ...
    ./ (k)) .* ((1 ./ T_stc) - (1./T))));

I01 = I_0;

I02 = I_0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%INICIO DO CICLO
x = 1;
y = 2;
limit = 1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
while (Eps_POT>TOL) & (limit >= 1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CALCULO DA RESISTÊNCIA SHUNT (RSH)
x1 = Vmp_stc + Imp_stc .* RS;
x2 = Ipv;
x3 = I01 .* (exp((Vmp_stc + Imp_stc .* RS) ./ (a1 .* Vt)) - 1);
x4 = I02 .* (exp((Vmp_stc + Imp_stc .* RS) ./ (a2 .* Vt)) - 1);
x5 = (PMAX_E ./ Vmp_stc);

RSH = x1 ./ (x2 - x3 - x4 - x5);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CALCULO DO a2 MAIS APROPRIADO (IDEALIDADE DO DIODO 2)

a2 = ((Vmp_stc + Imp_stc .* RS - Voc_stc) ./ (Vt .* (log (...
     (Isc_stc - (Vmp_stc ./ RSH) - Imp_stc)) - ...
     (log ((Isc_stc - (Voc_stc ./ RSH)))) ...
     + (Imp_stc ./ Isc_stc - (Voc_stc ./ RSH)))));
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CALCULO DE I APLICANDO O METODO
Va = 0;
Ia = 0;
i = 1;

while Ia>=0;

ID1 = (I01 .* (exp((Va + Ia .* RS )./ (a1 .* Vt_1)) - 1));

ID2 = (I02 .* (exp((Va + Ia .* RS )./ (a2 .* Vt_2)) - 1));

fIa = (Ipv - Ia - ID1 - ID2 - ((Va + Ia .* RS) ./ RSH));

fdIa =(-1 - I01 .* (RS ./ (a1 .* Vt_1)) .* exp ((Va + Ia .* RS)...
    ./ (a1 .* Vt_1)) - I02 .* (RS ./ (a2 .* Vt_2)) .* exp ((Va + Ia
.* RS)...
    ./ (a2 .* Vt_2)) - (RS ./ RSH));

Ia = Ia - ( fIa ./ fdIa);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
V(i) = Va;
I(i) = Ia;
i = i + 1;
Va = Va + 0.05;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CAlCULO DA POTENCIA
P = V .* I;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%Curva I-V QUE CARACTERIZA O MÓDULO
figure (1)
grid on
hold on
title('CURVA I-V - AJUSTAMENTO');
xlabel('V [V]');
ylabel('I [A]');
xlim([0 Voc_stc*1.1]);
ylim([0 Isc_stc*1.1]);
plot(V,I,'LineWidth',2,'Color','k')
plot([0 Vmp_stc Voc_stc],[Isc_stc Imp_stc
0],'o','LineWidth',2,'MarkerSize',5,'Color','k')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%POTÊNCIA VS TENSÃO
figure(2)
grid on
hold on
title('CURVA P-V - AJUSTAMENTO');
xlabel('V [V]');
ylabel('P [W]');
xlim([0 Voc_stc*1.1])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(V,P,'LineWidth',2,'Color','k')
plot([0 Vmp_stc Voc_stc],[0 Vmp_stc*Imp_stc
0],'o','LineWidth',2,'MarkerSize',5,'Color','k')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
Q = 1;
while P(Q) < P(Q+1);
Q = Q + 1;
end
P_MAX_M = P(Q);
V_MP = V(Q);
I_MP = I(Q);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%EPSILON
Eps_POT = abs((P_MAX_M - PMAX_E) ./ PMAX_E) .* 100;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%INCREMENTO DA RESISTENCIA SERIE
RS = RS + 0.01;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
A(x,1) = RSH;
A(x,2) = RS;
A(x,3) = P_MAX_M;
A(x,4) = V_MP;
A(x,5) = I_MP;
A(x,6) = Eps_POT;
A(x,7) = a2;
A;
B(1) = 10;
B(y) = Eps_POT;

if B(y-1) - B(y) >= 0
    limit = 1;
else
    limit = 0;
end
y = y + 1;
x = x + 1;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CONDIÇÕES DE REPRESENTAÇÃO
RSH = A(x-2,1);
RS = A(x-2,2);
Eps_POT = A(x-2,6);
a2 = A(x-2,7);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%FACTOR FORMA & CALCULO DE RENDIMENTO
A = l .* c .* Ns;
FF = (I_MP .* V_MP) ./ (Isc_stc .* Voc_stc);
PMAX_real = Voc_stc .* Isc_stc.* FF;
P_in = A .* 1000;
ren = ((I_MP .* V_MP)./ P_in) .* 100;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
% CURVA (I-V)/PARA
GUARDAR%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure (3)
grid on
hold on
title('Curva I-V [Condições STC]');
```

```
xlabel('V [V]');
ylabel('I [A]');
xlim([0 Voc_stc*1.1]);
ylim([0 Isc_stc*1.1]);
plot(V,I,'LineWidth',2,'Color','k')
saveas(figure (3),'CURVA I-V (STC)') ;
close
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
% CURVA (I-V)/PARA APARECER NA
FERRAMENTA%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axes(handles.I_V);
grid on
hold on
xlabel('V [V]');
ylabel('I [A]');
xlim([0 Voc_stc*1.1]);
ylim([0 Isc_stc*1.1]);
plot(V,I,'LineWidth',2,'Color','k')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
% CURVA P-V/PARA
GUARDAR%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure (4)
grid on
hold on
title('CURVA P-V [Condições STC]');
xlabel('V [V]');
ylabel('P [W]');
xlim([0 Voc_stc*1.1]);
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(V,P,'LineWidth',2,'Color','k')
saveas(figure (4),'CURVA P-V (STC)') ;
close
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
% CURVA P-V/PARA APARECER NA
FERRAMENTA%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axes(handles.P_V)
grid on
hold on
xlabel('V [V]');
ylabel('P [W]');
xlim([0 Voc_stc*1.1]);
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(V,P,'LineWidth',2,'Color','k')
%plot(Vexp,Pexp,'bo','LineWidth',2,'MarkerSize',6)%
%plot([0 V_MP Voc_stc ],[0 P_MAX_M 0
],'x','LineWidth',2,'MarkerSize',14,'Color','k')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
RSH = round(RSH.*100)/100;
RS = round(RS.*100)/100;
Eg = round(Eg.*100)/100;
a2 = round(a2.*100)/100;
Eps_POT = round(Eps_POT.*100)/100;
I_0 = round(I_0.*100)/100;
FF = round(FF.*100)/100;
PMAX_real = round(PMAX_real.*100)/100;
ren = round(ren.*100)/100;
Ipv = round(Ipv.*100)/100;
```

```
I_0 = I_0*1e8;

disp(sprintf('Dados de Entrada [STC]'));
disp(sprintf('Voc = %f[V]',Voc_stc ));
disp(sprintf('Isc_stc = %f[A]',Isc_stc));
disp(sprintf('Vmp_stc  = %f [V]',Vmp_stc ));
disp(sprintf('Imp_stc  = %f [A]',Imp_stc ));
disp(sprintf('Ki  = %f',Ki ));
disp(sprintf('Kv  = %f',Kv ));
disp(sprintf('Ns = %f[A]',Ns));
disp(sprintf('l  = %f',l ));
disp(sprintf('c  = %f[W]',c ));
disp(sprintf('\n\n'));
disp(sprintf('Dados de Saída [STC]'));
disp(sprintf('Rp = %f[Ohm]',RSH));
disp(sprintf('Rs = %f[Ohm]',RS));
disp(sprintf('Eg = %f',Eg));
disp(sprintf('a2 = %f',a2));
disp(sprintf('Eps_POT = %f',Eps_POT));
disp(sprintf('I01 = I02 = %f[x10^-8 A]',I_0));
disp(sprintf('Ipv = %f[A]',Ipv));
disp(sprintf('FF (STC) = %f',FF));
disp(sprintf('PMAX_real (STC) = %f[W]',PMAX_real));
disp(sprintf('ren (STC)= %f',ren));
disp(sprintf('\n\n'));

set(handles.Rsh_res,'String',RSH);
set(handles.Rs_res,'String',RS);
set(handles.Eg_res,'String',Eg);
set(handles.a2_res,'String',a2);
set(handles.Error_res,'String',Eps_POT);
set(handles.I0_res,'String',I_0);
set(handles.Ipv_res,'String',Ipv);
set(handles.FF_res,'String',FF);
set(handles.PMAX_res,'String',PMAX_real);
set(handles.Rend_res,'String',ren);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
for T = 273:25:348
V=0;
I=0;
G = 1; %IRRADIÂNCIA
T_stc = 298.15;
DT = T - T_stc;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CÁLCULO DAS TENSÕES TÉRMICAS
Vt_1 = ((a1 .* Ns .* k .* T) ./ q);
Vt_2 = ((a2 .* Ns .* k .* T) ./ q);
Vt = ((Ns .* k .* T) ./ q);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CÁLCULO DA CORRENTE FOTOVOLTAICA
Ipv_stc = Isc_stc;
Ipv = (Ipv_stc + (Ki .* DT)) .* (G ./ G_stc);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
```

```
%CÁLCULO DAS CORRENTES DE SATURAÇÃO PARA OS DIODOS 1 E 2 PARA A
SITUAÇÃO
%ONDE VOC É INFLUÊNCIADO POR Kv

I_0 = ((Isc_stc + Ki .* DT) ./ (exp((Voc_stc + Kv .* DT) ./ Vt)-1));

I01 = I_0;
I02 = I_0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
%INICIO DO CICLO
x = 1;
y = 2;
limit = 1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
%CALCULO DE I APLICANDO O METODO
Va = 0;
Ia = 0;
i = 1;
while Ia >= 0;
ID1 = (I01 .* (exp((Va + Ia .* RS )./ (a1 .* Vt_1)) - 1));
ID2 = (I02 .* (exp((Va + Ia .* RS )./ (a2 .* Vt_2)) - 1));
fIa = (Ipv - Ia - ID1 - ID2 - ((Va + Ia .* RS) ./ RSH));
fdIa =(-1 - I01 .* (RS ./ (a1 .* Vt)) .* exp ((Va + Ia .* RS)...
    ./ (a1 .* Vt)) - I02 .* (RS ./ (a2 .* Vt)) .* exp ((Va + Ia .*
RS)...
    ./ (a2 .* Vt)) - (RS ./ RSH));
Ia = Ia - ( fIa ./ fdIa);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
V(i) = Va;
I(i) = Ia;
i = i + 1;
Va = Va + 0.05;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
%CAlCULO DA POTENCIA
P = V .* I;
Q = 1;
while P(Q) < P(Q+1);
Q = Q + 1;
end
P_MAX_M =  P(Q);
V_M = V(Q);
I_MP = I(Q);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
%Curva I-V QUE CARACTERIZA O MÓDULO / PARA
GRAVAR%%%%%%%%%%%%%%%%%%%%%%%%%
figure (5);
grid on
hold on
title('CURVAS I-V [T: 75 -> 0°C / Irradiância: STC]');
xlabel('V [V]');
ylabel('I [A]');
xlim([0 Voc_stc*1.4]);
ylim([0 Ipv*1.1]);
plot(V,I,'LineWidth',2,'Color','k')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%Curvas I-V QUE CARACTERIZA O MÓDULO / PARA APARECER NA
FERRAMENTA%%%%%%%%%
axes(handles.CURVAS_I_V)
grid on
hold on
xlabel('V [V]');
ylabel('I [A]');
xlim([0 Voc_stc*1.4]);
ylim([0 Ipv*1.1]);
plot(V,I,'LineWidth',2,'Color','k')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%Curvas P-V QUE CARACTERIZA O MÓDULO / PARA
GRAVARR%%%%%%%%%%%%%%%%%%%%%%%%
figure (6);
grid on
hold on
title('CURVAS P-V [T: 75 -> 0°C / Irradiância: STC');
xlabel('V [V]');
ylabel('P [W]');
xlim([0 Voc_stc*1.5])
ylim([0 Vmp_stc*Imp_stc*1.4]);
plot(V,P,'LineWidth',2,'Color','k')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%Curvas I-V QUE CARACTERIZA O MÓDULO / PARA APARECER NA
FERRAMENTA%%%%%%%%%
axes(handles.CURVAS_P_V)
grid on
hold on
xlabel('V [V]');
ylabel('P [W]');
xlim([0 Voc_stc*1.5])
ylim([0 Vmp_stc*Imp_stc*1.4]);
plot(V,P,'LineWidth',2,'Color','k')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
end
saveas(figure (5),'CURVAS I-V.fig');
close figure 5
saveas(figure (6),'CURVAS P-V.fig');
close figure 6
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CONDIÇÕES DE TESTE
V = 0;
I = 0;
T = 298;       %TEMPERATURA DO MÓDULO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CÁLCULO DAS TENSÕES TÉRMICAS
Vt_1 = ((a1 .* Ns .* k .* T) ./ q);
Vt_2 = ((a2 .* Ns .* k .* T) ./ q);
Vt = ((Ns .* k .* T) ./ q);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
for G = 0.2:0.2:1
```

```
%CÁLCULO DA CORRENTE FOTOVOLTAICA
Ipv_stc = Isc_stc;
Ipv = (Ipv_stc + (Ki .* DT)) .* (G ./ G_stc);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CÁLCULO DAS CORRENTES DE SATURAÇÃO PARA OS DIODOS 1 E 2
I_0_n = ((Isc_stc) ./ (exp((Voc_stc) ./ (Vt)) - 1));
I_0 = (I_0_n) .* ((T_stc ./ T) ^ 3) .* (exp (((q .* Eg) ...
    ./ (k)) .* ((1 ./ T_stc) - (1./T))));
I01 = I_0;
I02 = I_0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%INICIO DO CICLO
x = 1;
y = 2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CALCULO DE I APLICANDO O METODO
Va = 0;
Ia = 0;
i = 1;
while Ia>=0;
ID1 = (I01 .* (exp((Va + Ia .* RS )./ (a1 .* Vt_1)) - 1));
ID2 = (I02 .* (exp((Va + Ia .* RS )./ (a2 .* Vt_2)) - 1));
fIa = (Ipv - Ia - ID1 - ID2 - ((Va + Ia .* RS) ./ RSH));
fdIa =(-1 - I01 .* (RS ./ (a1 .* Vt)) .* exp ((Va + Ia .* RS)...
    ./ (a1 .* Vt)) - I02 .* (RS ./ (a2 .* Vt)) .* exp ((Va + Ia .*
RS)...
    ./ (a2 .* Vt)) - (RS ./ RSH));
Ia = Ia - ( fIa ./ fdIa);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
V(i) = Va;
I(i) = Ia;
i = i + 1;
Va = Va + 0.05;
end

Q = 1;
while P(Q) < P(Q+1);
Q = Q + 1;
end
P_MAX_M = P(Q);
V_MP = V(Q);
I_MP = I(Q);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%Curva I-V QUE CARACTERIZA O MÓDULO / %PARA
GRAVAR%%%%%%%%%%%%%%%%%%%%%%%%%
figure (7);
grid on
hold on
title('CURVAS I-V [Irradiância: 0.2 -> 1 kW/m^2 / T: STC');
xlabel('V [V]');
ylabel('I [A]');
xlim([0 Voc_stc*1.1]);
ylim([0 Ipv*1.1]);
plot(V,I,'LineWidth',2,'Color','k')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
```

```
%Curva I-V QUE CARACTERIZA O MÓDULO / PARA APARECER NA
FERRAMENTA%%%%%%%%%%
axes(handles.CURVAS_I_V_1)
grid on
hold on
xlabel('V [V]');
ylabel('I [A]');
xlim([0 Voc_stc*1.1]);
ylim([0 Ipv*1.1]);
plot(V,I,'LineWidth',2,'Color','k')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CAlCULO DA POTENCIA
P = V .* I;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%Curva P-V QUE CARACTERIZA O MÓDULO / PARA
GRAVAR%%%%%%%%%%%%%%%%%%%%%%%%%%
figure (8);
grid on
hold on
title('CURVAS P-V [Irradiância: 0.2 -> 1 kW/m^2 / T: STC');
xlabel('V [V]');
ylabel('P [W]');
xlim([0 Voc_stc*1.1])
ylim([0 Vmp_stc*Imp_stc*1.2]);
plot(V,P,'LineWidth',2,'Color','k')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%Curva I-V QUE CARACTERIZA O MÓDULO / PARA APARECER NA
FERRAMENTA%%%%%%%%%%
axes(handles.CURVAS_P_V_1)
grid on
hold on
xlabel('V [V]');
ylabel('P [W]');
xlim([0 Voc_stc*1.1])
ylim([0 Vmp_stc*Imp_stc*1.2]);
plot(V,P,'LineWidth',2,'Color','k')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
end
saveas(figure (7),'CURVAS I-V 2.fig');
close figure 7
saveas(figure (8),'CURVAS P-V 2.fig');
close figure 8
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
contents = cellstr(get(handles.mono_poli,'String'));
temp1 = contents{get(handles.mono_poli,'Value')};

if strcmp(temp1, 'Monocristalino')
%temp1 = 'csc';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CÉLULAS MOCRISTALINAS (SEM DEPOSIÇÃO DE POEIRAS)
for t = 0:0.05:30
P_max = PMAX_E .* ((-0.2826 .* t + 44.176) ./ (44.176));
```

```
%PARA
GRAVAR%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
figure (9);
grid on
hold on
title('Células Monocristalinas Sem Deposição de Poeiras [Condições
STC]');
xlabel('t [Anos]');
ylabel('P [W] (Monocristalino)');
xlim([0 30 .* 1.1])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(t,P_max,'LineWidth',2,'Color','k')
%PARA APARECER NA FERRAMENTA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axes(handles.axes22)
grid on
hold on
xlabel('t [Anos]');
ylabel('P [W] (Monocristalino)');
xlim([0 30 .* 1.1])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(t,P_max,'LineWidth',2,'Color','k')
end
saveas(figure (9),'figure 9.fig') ;
close figure 9
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%CÉLULAS MOCRISTALINAS (COM DEPOSIÇÃO DE POEIRAS)
for t = 0:0.05:30
P_max = PMAX_E .* ((-0.4999 .* t + 43.808) ./ (43.808));
%PARA GRAVAR
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure (10);
grid on
hold on
title('Células Monocristalinas Com Deposição de Poeiras [Condições
STC]');
xlabel('t [Anos]');
ylabel('P [W] (Monocristalino)');
xlim([0 30 .* 1.1])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(t,P_max,'LineWidth',2,'Color','k')
%PARA APARECER NA FERRAMENTA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axes(handles.axes23)
grid on
hold on
%title('MONO. COM PÓ [STC]');
xlabel('t [Anos]');
ylabel('P [W] (Monocristalino)');
xlim([0 30 .* 1.1])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(t,P_max,'LineWidth',2,'Color','k')
end
saveas(figure (10),'figure 10.fig') ;
close figure 10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
elseif strcmp(temp1, 'Policristalino')
    if PMAX_E < 80
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CÉLULAS POLICRISTALINAS (SEM PÓ - PARA POTÊNCIAS INFERIORES A 80W)
for t = 0:0.05:30
P_max = PMAX_E .* ((-0.1096 .* t + 31.964) ./ (31.964));
%PARA GRAVAR
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure (9);
grid on
hold on
title('Células Policristalinas / SEM PÓ / PMAX < 80W [Condições
STC]');
xlabel('t [Anos]');
ylabel('P[W] (Poli. Pmáx < 80)');
xlim([0 Voc_stc*1.5])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(t,P_max,'LineWidth',2,'Color','k')
%PARA APARECER NA FERRAMENTA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axes(handles.axes22)
grid on
hold on
%title('POLI. SEM PÓ PMAX < 80W [STC]');
xlabel('t [Anos]');
ylabel('P[W] (Poli. Pmáx < 80)');
xlim([0 Voc_stc*1.5])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(t,P_max,'LineWidth',2,'Color','k')
end
saveas(figure (9),'figure 9.fig');
close figure 9
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CÉLULAS POLICRISTALINAS (COM PÓ - PARA POTÊNCIAS INFERIORES A 80W)
for t = 0:0.05:30
P_max = PMAX_E .* ((-0.2838 .* t + 31.774) ./ (31.774));
%PARA GRAVAR
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure (10);
grid on
hold on
title('Células Policristalinas / COM PÓ / PMAX < 80W [Condições
STC]');
xlabel('t [Anos]');
ylabel('P [W] Com Pó (Poli. Pmáx < 80) [STC]');
xlim([0 30 .* 1.1])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(t,P_max,'LineWidth',2,'Color','k')
%PARA APARECER NA FERRAMENTA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axes(handles.axes23)
grid on
hold on
%title('POLI. COM PÓ PMÁx < 80W [STC]');
xlabel('t [Anos]');
ylabel('P [W] (Poli. Pmáx < 80)');
xlim([0 30 .* 1.1])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(t,P_max,'LineWidth',2,'Color','k')
end
saveas(figure (10),'figure 10.fig');
```

```
close figure 10
    else
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CÉLULAS POLICRISTALINAS (SEM PÓ - PARA POTÊNCIAS SUPERIORES A 80W)
for t = 0:0.05:30
P_max = PMAX_E .* ((-2.6173 .* t + 111.03) ./ (111.03));
%PARA GRAVAR
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure (9);
grid on
hold on
title('Células Policristalinas / SEM PÓ / PMAX > 80W [Condições
STC]');
xlabel('t [Anos]');
ylabel('P [W] (Poli. Pmáx > 80)');
xlim([0 30 .* 1.1])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(t,P_max,'LineWidth',2,'Color','k')
%PARA APARECER NA FERRAMENTA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axes(handles.axes22)
grid on
hold on
%title('POLI. PMÁX > 80W');
xlabel('t [Anos]');
ylabel('P [W] (Poli. Pmáx > 80)');
xlim([0 30 .* 1.1])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(t,P_max,'LineWidth',2,'Color','k')
end
saveas(figure (9),'figure 9.fig');
close figure 9
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CÉLULAS POLICRISTALINAS (COM PÓ - PARA POTÊNCIAS SUPERIORES A 80W)
for t = 0:0.05:30
P_max = PMAX_E .* ((-2.839 .* t + 108.69) ./ (108.69));
%PARA GRAVAR
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure (10)
grid on
hold on
title('Células Policristalinas / COM PÓ / PMAX > 80W [Condições
STC]');
xlabel('t [Anos]');
ylabel('P [W] (Poli. Pmáx > 80)');
xlim([0 30 .* 1.1])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(t,P_max,'LineWidth',2,'Color','k')
%PARA APARECER NA FERRAMENTA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axes(handles.axes23)
grid on
hold on
%title('POLI. PMÁX > 80W');
xlabel('t [Anos]');
ylabel('P [W] (Poli. Pmáx > 80)');
xlim([0 30 .* 1.1])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(t,P_max,'LineWidth',2,'Color','k')
```

```
end
saveas(figure (10),'figure 10.fig');
close figure 10
    end
end
guidata(hObject, handles);

function edit50_Callback(hObject, eventdata, handles)
% hObject    handle to edit50 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit50 as text
%        str2double(get(hObject,'String')) returns contents of
edit50 as a double

% --- Executes during object creation, after setting all properties.
function edit50_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit50 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit 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 edit51_Callback(hObject, eventdata, handles)
% hObject    handle to edit51 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit51 as text
%        str2double(get(hObject,'String')) returns contents of
edit51 as a double

% --- Executes during object creation, after setting all properties.
function edit51_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit51 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit 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 edit52_Callback(hObject, eventdata, handles)
% hObject    handle to edit52 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit52 as text
%        str2double(get(hObject,'String')) returns contents of
edit52 as a double
```

```
% --- Executes during object creation, after setting all properties.
function edit52_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit52 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit 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 edit53_Callback(hObject, eventdata, handles)
% hObject    handle to edit53 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit53 as text
%        str2double(get(hObject,'String')) returns contents of
edit53 as a double

% --- Executes during object creation, after setting all properties.
function edit53_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit53 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit 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 edit54_Callback(hObject, eventdata, handles)
% hObject    handle to edit54 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit54 as text
%        str2double(get(hObject,'String')) returns contents of
edit54 as a double

% --- Executes during object creation, after setting all properties.
function edit54_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit54 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit 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 edit55_Callback(hObject, eventdata, handles)
% hObject    handle to edit55 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit55 as text
%        str2double(get(hObject,'String')) returns contents of
edit55 as a double

% --- Executes during object creation, after setting all properties.
function edit55_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit55 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit 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 edit56_Callback(hObject, eventdata, handles)
% hObject    handle to edit56 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit56 as text
%        str2double(get(hObject,'String')) returns contents of
edit56 as a double

% --- Executes during object creation, after setting all properties.
function edit56_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit56 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit 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 edit57_Callback(hObject, eventdata, handles)
% hObject    handle to edit57 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit57 as text
%        str2double(get(hObject,'String')) returns contents of
edit57 as a double

% --- Executes during object creation, after setting all properties.
function edit57_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit57 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit 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 edit58_Callback(hObject, eventdata, handles)
% hObject    handle to edit58 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit58 as text
%        str2double(get(hObject,'String')) returns contents of
edit58 as a double

% --- Executes during object creation, after setting all properties.
function edit58_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit58 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit 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 edit59_Callback(hObject, eventdata, handles)
% hObject    handle to edit59 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit59 as text
%        str2double(get(hObject,'String')) returns contents of
edit59 as a double

% --- Executes during object creation, after setting all properties.
function edit59_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit59 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit 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 Voc_in_Callback(hObject, eventdata, handles)
% hObject    handle to Voc_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Voc_in as text
```

```
%        str2double(get(hObject,'String')) returns contents of
Voc_in as a double
input = str2num(get(hObject,'String'));
if (isempty(input))
     set(hObject,'String','0')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function Voc_in_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Voc_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit 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 Isc_in_Callback(hObject, eventdata, handles)
% hObject    handle to Isc_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Isc_in as text
%        str2double(get(hObject,'String')) returns contents of
Isc_in as a double
input = str2num(get(hObject,'String'));
if (isempty(input))
     set(hObject,'String','0')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function Isc_in_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Isc_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit 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 Vmp_in_Callback(hObject, eventdata, handles)
% hObject    handle to Vmp_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Vmp_in as text
%        str2double(get(hObject,'String')) returns contents of
Vmp_in as a double
input = str2num(get(hObject,'String'));
if (isempty(input))
     set(hObject,'String','0')
```

```
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function Vmp_in_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Vmp_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit 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 Imp_in_Callback(hObject, eventdata, handles)
% hObject    handle to Imp_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Imp_in as text
%        str2double(get(hObject,'String')) returns contents of
Imp_in as a double
input = str2num(get(hObject,'String'));
if (isempty(input))
     set(hObject,'String','0')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function Imp_in_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Imp_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit 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 ki_in_Callback(hObject, eventdata, handles)
% hObject    handle to ki_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ki_in as text
%        str2double(get(hObject,'String')) returns contents of ki_in
as a double
input = str2num(get(hObject,'String'));
if (isempty(input))
     set(hObject,'String','0')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function ki_in_CreateFcn(hObject, eventdata, handles)
```

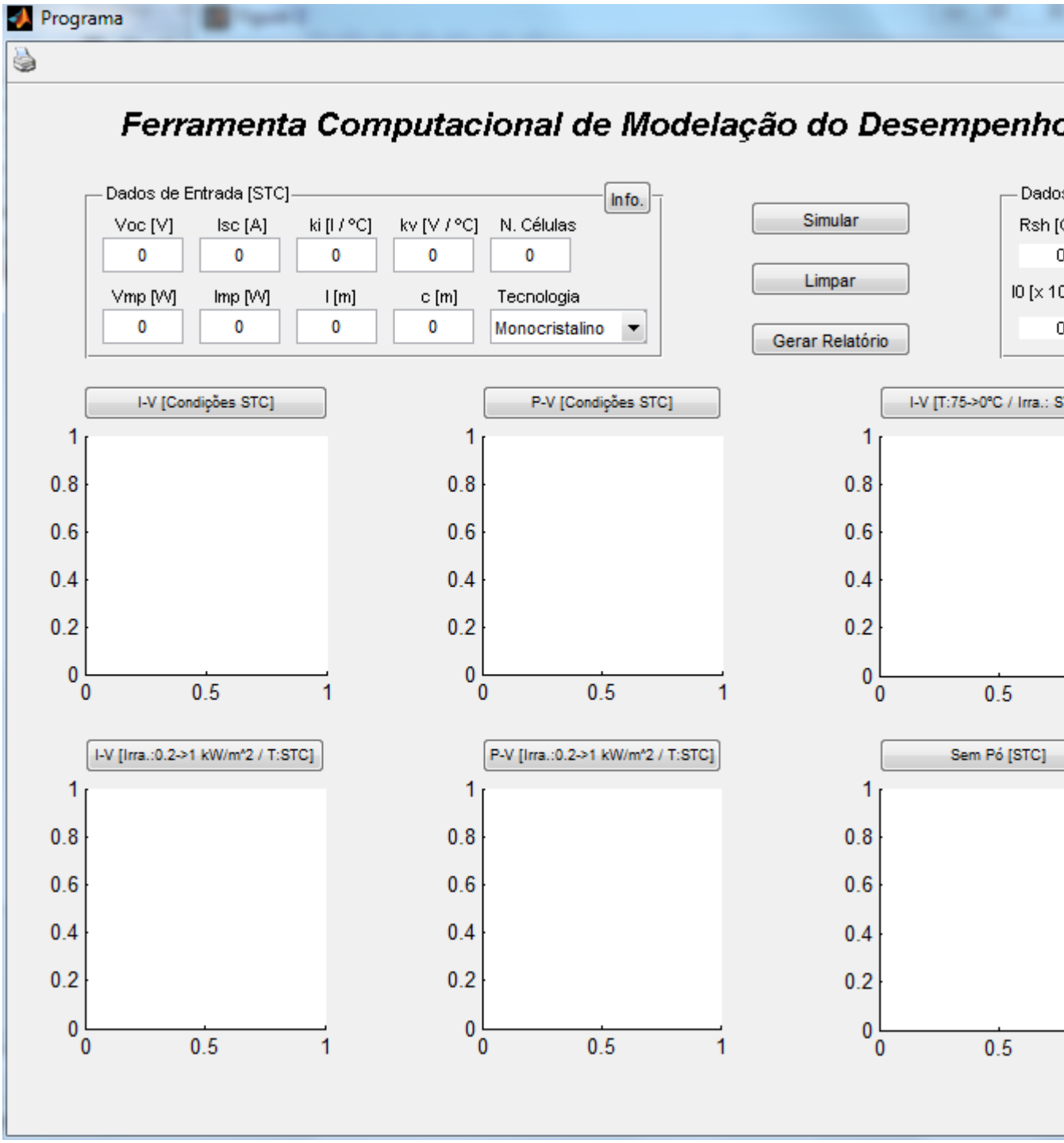
```
% hObject    handle to ki_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit 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 l_in_Callback(hObject, eventdata, handles)
% hObject    handle to l_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of l_in as text
%        str2double(get(hObject,'String')) returns contents of l_in
as a double

% --- Executes during object creation, after setting all properties.
function l_in_CreateFcn(hObject, eventdata, handles)
% hObject    handle to l_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit 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 kv_in_Callback(hObject, eventdata, handles)
% hObject    handle to kv_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of kv_in as text
%        str2double(get(hObject,'String')) returns contents of kv_in
as a double
input = str2num(get(hObject,'String'));
if (isempty(input))
     set(hObject,'String','0')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function kv_in_CreateFcn(hObject, eventdata, handles)
% hObject    handle to kv_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit 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 c_in_Callback(hObject, eventdata, handles)
% hObject    handle to c_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of c_in as text
%        str2double(get(hObject,'String')) returns contents of c_in
as a double

% --- Executes during object creation, after setting all properties.
function c_in_CreateFcn(hObject, eventdata, handles)
% hObject    handle to c_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit 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 Nc_in_Callback(hObject, eventdata, handles)
% hObject    handle to Nc_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Nc_in as text
%        str2double(get(hObject,'String')) returns contents of Nc_in
as a double
input = str2num(get(hObject,'String'));
if (isempty(input))
     set(hObject,'String','0')
end
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function Nc_in_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Nc_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit 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

% --- Executes on selection change in mono_poli.
function mono_poli_Callback(hObject, eventdata, handles)
% hObject    handle to mono_poli (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns mono_poli
contents as cell array
```

```
%        contents{get(hObject,'Value')} returns selected item from
mono_poli

% --- Executes during object creation, after setting all properties.
function mono_poli_CreateFcn(hObject, eventdata, handles)
% hObject    handle to mono_poli (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

% --- If Enable == 'on', executes on mouse press in 5 pixel border.
% --- Otherwise, executes on mouse press in 5 pixel border or over
Rsh_res.
function Rsh_res_ButtonDownFcn(hObject, eventdata, handles)
% hObject    handle to Rsh_res (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% ------------------------------------------------------------------
--
function Untitled_1_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in Limpar_Pushbutton.
function Limpar_Pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to Limpar_Pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.Voc_in,'String','0')
set(handles.Isc_in,'String','0')
set(handles.Vmp_in,'String','0')
set(handles.Imp_in,'String','0')
set(handles.ki_in,'String','0')
set(handles.kv_in,'String','0')
set(handles.Nc_in,'String','0')
set(handles.l_in,'String','0')
set(handles.c_in,'String','0')
set(handles.Rsh_res,'String','0');
set(handles.Rs_res,'String','0');
set(handles.Eg_res,'String','0');
set(handles.a2_res,'String','0');
set(handles.Error_res,'String','0');
set(handles.I0_res,'String','0');
set(handles.Ipv_res,'String','0');
set(handles.FF_res,'String','0');
set(handles.PMAX_res,'String','0');
set(handles.Rend_res,'String','0');
arrayfun(@cla,findall(0,'type','axes'))

% --- Executes on button press in Gerar_Pushbutton.
function Gerar_Pushbutton_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to Gerar_Pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in Curva_1.
function Curva_1_Callback(hObject, eventdata, handles)
% hObject    handle to Curva_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%load test.fig
CURVA_1 = 'CURVA I-V (STC).fig';
open(CURVA_1);

% --- Executes on button press in Curva_2.
function Curva_2_Callback(hObject, eventdata, handles)
% hObject    handle to Curva_2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
CURVA_2 = 'CURVA P-V (STC).fig';
open(CURVA_2);

% --- Executes on button press in Curva_3.
function Curva_3_Callback(hObject, eventdata, handles)
% hObject    handle to Curva_3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
CURVA_3 = 'CURVAS I-V.fig';
open(CURVA_3);

% --- Executes on button press in Curva_4.
function Curva_4_Callback(hObject, eventdata, handles)
% hObject    handle to Curva_4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
CURVA_4 = 'CURVAS P-V.fig';
open(CURVA_4);

% --- Executes on button press in Curva_5.
function Curva_5_Callback(hObject, eventdata, handles)
% hObject    handle to Curva_5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
CURVA_5 = 'CURVAS I-V 2.fig';
open(CURVA_5);

% --- Executes on button press in Curva_6.
function Curva_6_Callback(hObject, eventdata, handles)
% hObject    handle to Curva_6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
CURVA_6 = 'CURVAS P-V 2.fig';
open(CURVA_6);

% --- Executes on button press in Curva_7.
function Curva_7_Callback(hObject, eventdata, handles)
% hObject    handle to Curva_7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
CURVA_7 = 'figure 9.fig';
open(CURVA_7);
```

```
% --- Executes on button press in Curva_8.
function Curva_8_Callback(hObject, eventdata, handles)
% hObject    handle to Curva_8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
CURVA_8 = 'figure 10.fig';
open(CURVA_8);

% --- Executes on button press in Menu.
function Menu_Callback(hObject, eventdata, handles)
% hObject    handle to Menu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close
FCMDPF

% --- Executes on button press in Info2.
function Info2_Callback(hObject, eventdata, handles)
% hObject    handle to Info2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
helpdlg('[STC]: Condições Normais de Teste - Temp.= 25 [graus],
Irrad.= 1 [kW/m^2];
Rsh [Ohm]: Resistência Shunt (Paralela) [STC];
Rs [Ohm]: Resistência Série [STC];
Eg [eV]: Banda de Valência [STC];
a2: Factor de Idealidade do Díodo 2 [STC];
Erro [%]: Erro Relativo no Cálculo da Potência Máxima [STC];
I0 [A]: Corrente de Saturação [STC];
Ipv [A]: Corrente Fotovoltaica [STC];
FF: Factor de Forma [STC];
PMax [W]: Potência Máxima Real [STC];
Rend.[%]: Rendimento do Módulo [STC].','Descrição dos Dados de
Saída');

% --- Executes on button press in Info1.
function Info1_Callback(hObject, eventdata, handles)
% hObject    handle to Info1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
helpdlg('[STC]: Condições Normais de Teste - Temp.= 25 [graus],
Irrad.= 1 [kW/m^2];
Voc [V]: Tensão em Circuito Aberto [STC];
Isc [A]: Corrente em Curto-Circuito [STC];
ki [I/Grau Celsius]: Coeficiente de Temperatura de Isc [STC];
kv [V/Grau Celsius]: Coeficiente de Temperatura de Voc [STC];
N.Células: Número de Células do Módulo;
Vmp [W]: Tensão no Ponto de Potência Máxima [STC];
Imp [W]: Corrente no Ponto de Potência Máxima [STC];
l [m]: Largura da Célula Fotovoltaica;
c [m]: Comprimento da Célula Fotovoltaica;
Tecnologia: Seleccionar Monocristalino ou
Policristalino.','Descrição dos Dados de Entrada');
```



Published with MATLAB® 7.9