

# ANEXO I – CODIGO DEL PROGRAMA

```
function varargout = GUIson(varargin)
% GUISON M-file for GUIson.fig
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @GUIson_OpeningFcn, ...
                  'gui_OutputFcn',  @GUIson_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 GUIson is made visible.
function GUIson_OpeningFcn(hObject, ~, handles, varargin)

%% Dibuja la imagen en axes1
axes(handles.axes4);
axis off;
axes(handles.axes1)
a=imread('Dibujoson.jpg');
b=imread('logo.jpg');
axes(handles.axes1)
image(a);
axis off;
axes(handles.axes3)
image(b);
axis off;
axes(handles.axes2);
xlabel('ln(t) [s]')
ylabel('T_f(t) - T(t=0) [°C]')

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

% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
function varargout = GUIson_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function edit1_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
```

```

end

function edit4_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
%% Variables globales
global L;
global Q;
global p;
global q_c;
global lambda;
%% Calcula lambda
q_c=Q/L;
lambda = q_c/(4*pi*p(1));
set(handles.text6,'string',lambda);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
%% Variables globales
global r_b;
global L;
global T_t0;
global Q;
global p;
global n;

[nombre path]=uigetfile('*.');
%% Esto es por si se usa el boton cancelar
if nombre == 0
    return
end
%% Carga las entradas de los cuadros en blanco
r_b1=get(handles.edit1,'string');
L1=get(handles.edit2,'string');
T_t01=get(handles.edit3,'string');
Q1=get(handles.edit4,'string');
%% Transforma de string a nummer
r_b=str2num(r_b1);
L=str2num(L1);
T_t0=str2num(T_t01);
Q=str2num(Q1);
%% Elaborar ruta del archivo excel
todojunto = strcat(path,nombre);
%% Carga de datos del archivo excel
datos=XLSREAD(todojunto);
NumDat=length(datos);
NumDat=NumDat+1;
NumDat1=num2str(NumDat);
CeldasA = strcat('A2:A',NumDat1);
CeldasB = strcat('B2:B',NumDat1);
CeldasC = strcat('C2:C',NumDat1);
[t]=XLSREAD(todojunto,'Hojal',CeldasA);
[Tin]=XLSREAD(todojunto,'Hojal',CeldasB);
[Tout]=XLSREAD(todojunto,'Hojal',CeldasC);
%% Calculo de valores para elaborar la grafica
lnt=log(t);
Tf_t=(Tin+Tout)/2;
y=Tf_t-T_t0;
x=lnt;
%% Calcula la recta de regresión
p=polyfit(x,y,1);
m=num2str(p(1));
n=num2str(p(2));
ecuacion = strcat('y = ',m,' · x + ',n);
set(handles.text5,'string',ecuacion);
%% Dibuja la grafica
axes(handles.axes2);
plot(x,y);
xlabel('ln(t)')
ylabel('T_f(t) - T(t=0) [°C]')
text(x(1),y(1)+2,ecuacion)
%% Dibuja la recta de regresión

```

```

hold on
xx=1nt;
yy=p(1)*xx+p(2);
plot(xx,yy,'-.k');

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
%% Variables globales
global r_b;
global L;
global T_t0;
global Q;
global p;
global e;
global t;
global u1;
global g;

T_t01=num2str(T_t0);
Q1=num2str(Q);
%% Aquí va el toolbox PDE
[pde_fig,ax]=pdeinit;
pdetool('appl_cb',9);
set(ax,'DataAspectRatio',[12 37.5 1]);
set(ax,'PlotBoxAspectRatio',[1 1 1]);
set(ax,'XLim',[-r_b*55 r_b*55]);
set(ax,'YLim',[-L^2+10 5]);
set(ax,'XTickMode','auto');
set(ax,'YTickMode','auto');

% Geometry description:
pderect([-r_b r_b 0 -L],'R1');
pderect([-r_b*50 r_b*50 0 -L*2],'R2');
set(findobj(get(pde_fig,'Children'),'Tag','PDEEval'),'String','R2+R1')

% Boundary conditions:
pdetool('changemode',0)
pdesetbd(9,'dir',1,'1',T_t01)
pdesetbd(8,'dir',1,'1',T_t01)
pdesetbd(7,'dir',1,'1',T_t01)
pdesetbd(6,'dir',1,'1',T_t01)
pdesetbd(5,'dir',1,'1',T_t01)
pdesetbd(4,'dir',1,'1',T_t01)

% Mesh generation:
setappdata(pde_fig,'trisize',1);
setappdata(pde_fig,'Hgrad',1.8999999999999999);
setappdata(pde_fig,'refinemethod','longest');
setappdata(pde_fig,'jiggle',char('on','minimum',''));
pdetool('initmesh')

% PDE coefficients:
pdeseteq(2,'2!1.4','0!0','(500)+(0).*(13.882)!(0)+(0).*(13.882)','(1.0).*(1.87)!(1.0).*(1.87)','0:10','0.0','0.0',
',[0 100]')
setappdata(pde_fig,'currparam',[ '1.0!1.0      ' ; '1.87!1.87      ' ; '2!1.4      ' ; '500!0      ' ; '0!0      ' ;
'13.882!13.882'])

% Solve parameters:
setappdata(pde_fig,'solveparam',str2mat('0','5844','10','pdeadworst','0.5','longest','0','1E-4','','fixed','Inf'))

% Plotflags and user data strings:
setappdata(pde_fig,'plotflags',[1 1 1 1 1 1 1 0 0 0 11 1 1 0 0 0 2]);
setappdata(pde_fig,'colstring','');
setappdata(pde_fig,'arrowstring','');
setappdata(pde_fig,'deformstring','');
setappdata(pde_fig,'heightstring','');

% Solve PDE:

pdetool('solve')

getpetuc;

axes(handles.axes4)
pdegplot(g)
hold on
pdecont(p,t,u1)

```

```

hold on
pdeplot(p,e,t,'xydata',u1,'zdata',u1,'mesh','off')
zlabel('temp [°C]')

function [p,e,t,u1,l] = getpetuc

global p;
global e;
global t;
global u1;
global g;

%GETPETUC Get p,e,t,u, and c.

pde_fig=findobj(allchild(0),'flat','Tag','PDETool');

if isempty(pde_fig)

    error('PDE Toolbox GUI not active.')

end

u = get(findobj(pde_fig,'Tag','PDEPlotMenu'),'UserData');

l =get(findobj(pde_fig,'Tag','winmenu'),'UserData');

h=findobj(get(pde_fig,'Children'),'flat','Tag','PDEMeshMenu');

hp=findobj(get(h,'Children'),'flat','Tag','PDEInitMesh');

he=findobj(get(h,'Children'),'flat','Tag','PDERefine');

ht=findobj(get(h,'Children'),'flat','Tag','PDEMeshParam');

p=get(hp,'UserData');

e=get(he,'UserData');

t=get(ht,'UserData');

usize=size(u)
u1=u(:,usize(2));

hbound = findobj(get(pde_fig,'Children'),'flat','Tag','PDEBoundMenu');

g = get(hbound, 'userdata');

function edit5_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
global posicion;
posicion=get(hObject,'Value');
if posicion==1
    set(handles.text8,'string','alpha [m^2·s-1]');
    set(handles.text10,'string','Rb [mKW-1]');
end
if posicion==2
    set(handles.text8,'string','Rb [mKW-1]');
    set(handles.text10,'string','alpha [m^2·s-1]');
end

% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)

```

```

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

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% Definición de globales
global posicion;
global n;
global q_c;
global lambda;
global r_b;
% Elección de pestaña y calculo
if posicion==1
    alpha=get(handles.edit6,'string');
    alpha1=str2num(alpha);
    n1=str2num(n);
    Rb=(n1/q_c)-(1/(4*pi*lambda))*(log(4*alpha1/r_b^2)-0.5772);
    Rb1=num2str(Rb);
    set(handles.text11,'string',Rb1);
end
if posicion==2
    Rb=get(handles.edit6,'string');
    Rb1=str2num(Rb);
    n1=str2num(n);
    alpha=(r_b^2/4)*(exp(4*pi*lambda*(n1/q_c-Rb1)+0.5772));
    alpha1=num2str(alpha);
    set(handles.text11,'string',alpha1);
end

```