```
%% Definimos la funcion general de la aplicación
function varargout = MonitorCEC(varargin)
% MONITORCEC MATLAB code for MonitorCEC.fig
% MONITORCEC, by itself, creates a new MONITORCEC or raises the
existing
% singleton*.
%
% = MONITORCEC returns the handle to a new MONITORCEC or the handle to
% the existing singleton*.
%
% MONITORCEC('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in MONITORCEC.M with the given input
arguments.
%
% MONITORCEC('Property','Value',...) creates a new MONITORCEC or
raises the existing singleton*. Starting from the left, property value
pairs are
% applied to the GUI before MonitorCEC_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property
application
% stop. All inputs are passed to MonitorCEC_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 MonitorCEC
% Last Modified by GUIDE v2.5 14-Feb-2014 15:53:38
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
'gui_Singleton', gui_Singleton, ...
'gui_OpeningFcn', @MonitorCEC_OpeningFcn, ...
'gui_OutputFcn', @MonitorCEC_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 %% Inicialización del código
% --- Executes just before MonitorCEC is made visible.
function MonitorCEC_OpeningFcn(hObject, ~, 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 MonitorCEC (see VARARGIN)
% Choose default command line output for MonitorCEC
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
% UIWAIT makes MonitorCEC wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command line.
function varargout = MonitorCEC_OutputFcn(hObject, ~, handles)
%% Cargamos la GUI para configurar entrada de audio
salida= daqhwinfo('winsound'); %% Reconocemos entrada y salida de
audio
tarjetas=salida.BoardNames;
in = listdlg('PromptString','Selecione la entrada de sonido',...
'SelectionMode','single',...
'ListString',tarjetas,'ListSize',[300 300]);
% Finalizamos la configuración de entrada de audio
% Guardamos la variable canal que define el tipo de entrada de audio
if ( isempty(in))
set(handles.togglebutton1,'Visible','off');
waitfor( errordlg('Debe selecionar un canal para poder iniciar la
aplicación',...
'Alerta!','OK'));
else
set(handles.togglebutton1,'Visible','on');
end
handles.in=in;
% Finalización de la configuración de la entra de audio
%% Activamos el botón para iniciar el proceso de captura
%% Revisamos si es la primera vez que se ejecuta el programa
global datec;
datec=textscan('bin.txt',' %s');
```

```
varargout{1} = handles.output;
guidata(hObject,handles);
% --------------------------------------------------------------------
function archivo_menu_Callback(hObject, eventdata, handles) %%
Definimos las funciones los botones de la GUI
% hObject handle to archivo_menu (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% --------------------------------------------------------------------
function graficar_menu_Callback(hObject, eventdata, handles)
% hObject handle to graficar_menu (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% --------------------------------------------------------------------
function ayuda_menu_Callback(hObject, eventdata, handles)
% hObject handle to ayuda_menu (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% --------------------------------------------------------------------
function menu_ayuda_verayuda_Callback(hObject, eventdata, handles)
%% Muestra la ayuda de software
winopen('ManualUsuario.pdf')
guidata(hObject,handles);
% --------------------------------------------------------------------
function menu_ayuda_acercaSID_Callback(hObject, eventdata, handles)
msgbox(sprintf(' MonitorCEC V. 1.00\n\nEstá diseñado para detectar las
perturbaciones bruscas de la ionosfera,\n que son causadas por la
explosión de la radiación de rayos X\n intensos, cuando existe una
gran radiación solar.\n\n Oscar Eduardo Pulgarín Duque\n Luisa Maria
Puerta Gonzales\n\n Universidad Tecnológica de Pereira\n Programa de
Ingeniería Electrónica\n Grupo de Investigación en Astroingeniería
Alfa Orión\n 2013'),'Acerca de CEC');
th = findall(0, 'Tag','MessageBox' );
% --------------------------------------------------------------------
function menu_graficar_cargar_Callback(hObject, eventdata, handles)
%% Debe cargar la gráfica que se haya realizado con anterioridad
[FileName Path]=uigetfile({'*.xls'},'Abrir Documento'); %% Genera los
datos para la grafica principal
NameSave = ([Path FileName]);
```

```
SG=xlsread(NameSave,1,'C2:C17281');
h=17280
SG(1)= (SG(2)+SG(3))/2;
SG(17280)= (SG(17279)+SG(17278))/2;
for i=2:h-1 %% Se filtran valores de cero para que la graficas sean de
valor absoluto
if (i~=1 || i~=h)
if (SG(i)==0 )
SG(i)= (SG(i-1)+SG(i+1))/2;
end
end
end
windowSize = 10;
SG=filter(ones(1,windowSize)/windowSize,1,SG)
ejex =linspace(0,24,17280); %% Se define el tamaño y la cantidad de
los datos
SG=SG';
j=size(ejex) %% Grafica de la señal de entrada en DSP
v=size(SG)
figure(1)
hold off
plot(ejex(1:4320),SG(1:4320),'b')
hold on
plot(ejex(4321:12961),SG(4321:12961),'y')
hold on
plot(ejex(12962:17280),SG(12962:17280),'b')
min(SG)
ylim([min(SG)-2 max(SG)+1])
xlabel ('Time GMT-5')
ylabel ('Power Spectrum(dB)')
grid on
hold on
guidata(hObject,handles);
% --------------------------------------------------------------------
function menu_graficar_ver_Callback(hObject, eventdata, handles) %%
Definimos las demas funciones del programa
% hObject handle to menu_graficar_ver (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% --------------------------------------------------------------------
function menu_archivo_nuevo_Callback(hObject, eventdata, handles)
X=getframe(gca);
X=frame2im(X);
[FileName, PathName] = uiputfile('*.jpg', 'Save As');
```

```
imwrite(X,FileName,'jpg')
guidata(hObject,handles);
% --------------------------------------------------------------------
function menu_archivo_abrir_Callback(hObject, eventdata, handles)
%%% ojo aca se carga la señal
% --------------------------------------------------------------------
function menu_archivo_imprimir_Callback(hObject, eventdata, handles)
%% Se debe imprimir la imagen
X=getframe(gca);
printpreview()
guidata(hObject,handles);
% --------------------------------------------------------------------
function menu_archivo_salir_Callback(hObject, eventdata, handles)
% Función para salir de la plataforma
close all;
% --------------------------------------------------------------------
function menu_graficar_versenal_Callback(hObject, eventdata, handles)
%% Debe mostrar señal Capturada
global l1;
l1=1;
hold off;
guidata(hObject,handles);
% --------------------------------------------------------------------
function menu_graficar_verDPS_Callback(hObject, eventdata, handles)
%% Debe mostrar pestaña de señal DSP
global l1;
l1=0;
hold off;
guidata(hObject,handles);
% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject handle to popupmenu1 (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.
```

```matlab
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)
% --- Executes on button press in togglebutton1.
function togglebutton1_Callback(hObject, eventdata, handles)
global NWC NPM JJI NAA NLK NML cont fecha3 datec k;
% %Verificamos el estado del toogle button
k=get(handles.togglebutton1, 'Value');
if (k==1)
set(handles.togglebutton1, 'String', 'Detener Proceso');
set(handles.pushbutton11,'Visible','on');
else
set(handles.togglebutton1, 'String', 'Iniciar Proceso');
set(handles.pushbutton11,'Visible','off');
end
set(handles.axes1,'Visible','on');
grid on;
% Creamos la entrada de audio
AI= analoginput('winsound');
% Adicionamos el canal
addchannel(AI,handles.in);
% Configuramos el tiempo de muestreo y la duración
cont=0;
Periodo=4.3; % Periodo en segundos
SampleRate = 96000; % Máximo tmuestreo soportado
% Enviamos los datos de muestreo y duración al canal
set(AI,'SampleRate',SampleRate);
set(AI,'SamplesPerTrigger',Periodo*SampleRate);
% creación de bandera para archivos nuevos
% cuando inicia un nuevo día
global l1;
l1=0;
%Inicializamos variables para el calculo de la DPS
%para ahorrar memoria en Matlab
pxx=zeros(2501,1);
f=zeros(2501,1);
Pxx=zeros(2501,1);
```

```
while(k==1)
% Cálculamos el valor de tiempo donde deben almacenarse los datos
c=round(clock);
while( mod(c(6),5)|| c(6)==0 );
c=round(clock);
end
% Calculamos la variable i para saber en que punto de la tabla excel
% metemos los nuevos valores calculados donde:
% c(4)= horas, c(5)= minutos, c(6)= segundos
i=((c(4)*60*60)+(c(5)*60)+c(6))/5;
%% CÁLCULO DE LA HORA Y FECHA EN TIEMPO REAL
%Ajustamos la fecha y hora en tiempo real
ano=num2str(c(1));
mes=num2str(c(2));
dia=num2str(c(3));
%Si el mes o el día es de solo un dato entonces le pones un cero
%para completarlo
if (length(mes)==1)
mes=strcat('0',mes);
end
if (length(dia)==1)
dia=strcat('0',dia);
end
% Guardamos la fecha real
fecha3=[ano,'-',mes,'-',dia];
datec=textread('bin.txt',' %s');
datec=char(datec);
%% Comparamos fechas si es fecha diferente a la actual
if(strcmp(datec ,fecha3)~=1)
%Guardamos datos almacenados en el día
GuardarDatos();
NWC(1:17280)=zeros;
NPM(1:17280)=zeros;
JJI(1:17280)=zeros;
NAA(1:17280)=zeros;
NLK(1:17280)=zeros;
NML(1:17280)=zeros;
[~,struc] = fileattrib;
PathCurrent = struc.Name;
PathFolder= ([PathCurrent '/Datos/']);
```

```
buffer_name=strcat('Buffer_Actual','.xls');
NameBF = ([PathFolder (buffer_name)]);
senales = {'NWC' 'NPM' 'JJI' 'NAA' 'NLK' 'NML'};
xlswrite(NameBF, {fecha3}, 1, 'A1');
xlswrite(NameBF, senales, 1, 'A2');
xlswrite(NameBF, NWC', 1, 'A3');
xlswrite(NameBF, NPM', 1, 'B3');
xlswrite(NameBF, JJI', 1, 'C3');
xlswrite(NameBF, NAA', 1, 'D3');
xlswrite(NameBF, NLK', 1, 'E3');
xlswrite(NameBF, NML', 1, 'F3');
end
% Configuramos la tarjeta para iniciar la adquisición
start(AI);
wait(AI,Periodo*1.003);
%-------- Guarda los datos de la memoria en una variable en el espacio
de trabajo----
data=getdata(AI);
handles.data= data;
if (l1==1)
% Vector eje X para graficar la señal original
N=length(data);
t(N)=zeros;
parfor j=1:N
t(j)=j;
end
set(handles.axes1,'Visible','off');
set(handles.axes2,'Visible','on');
axes(handles.axes2);
plot(t,data)
hold off
grid on
end
%%____________GRAFICA DE LA SEÑAL ACTUAL_________________
%%
%% CÁLCULO Y GRÁFICA DE LA DENSIDAD ESPECTRAL DE POTENCIA
% mediante la aproximación pwelch con oversampling=1024
[pxx,f] = pwelch(data,5000,1024,5000,SampleRate);
handles.f=f;
```

```
Pxx= 10.4*log10(pxx)+143; %% Valor determinado para que la señal se
muestre en dB
handles.Pxx=Pxx;
guidata(hObject,handles);
if (l1==0)
set(handles.axes2,'Visible','off');
set(handles.axes1,'Visible','on');
% Graficamos la DSP
axes(handles.axes1);
plot(f,Pxx,'m')
hold off
set(gca,'XTickLabel',{'0','5000','10000','15000','20000','25000','3000
0','35000','40000','45000','50000'})
grid on
xlabel('Frequency (Hz)'); ylabel('Power Spectrum (dB)');
end
%% FIN DEL CÁLCULO DE LA DENSIDAD ESPECTRAL DE POTENCIA
%% ALMACENAMIENTO DE LAS SEÑALES DE INTERÉS (NWC,NPW,JJI,NAA,NLK y
NML)
%% DEFINIDO PARA LAS ESTACIONES MAS CERCANAS EL MONITOR CEC SOLO
TOMARA COMO BASE LA ESTACIÓN NAA
NWC(i)=Pxx(1033);
if (NWC(i)<=0)
NWC(i)=0;
end
NPM(i)=Pxx(1116);
if (NPM(i)<=0)
NPM(i)=0;
end
JJI(i)=Pxx(1158);
if (JJI(i)<=0)
JJI(i)=0;
end
NAA(i)=Pxx(1251);
if (NAA(i)<=0)
NAA(i)=0;
end
NLK(i)=Pxx(1293);
if (NLK(i)<=0)
NLK(i)=0;
end
NML(i)=Pxx(1314);
if (NML(i)<=0)
NML(i)=0;
end
cont=cont+1;
%% Cada 36 muestras se graban temporalmente los datos (5 minutos)
```

```
if(cont==36)
%Creamos los nombres de cada archivo
[stat,struc] = fileattrib;
PathCurrent = struc.Name;
PathFolder= ([PathCurrent '/Datos/']);
buffer_name=strcat('Buffer_Actual','.xls');
NameBF = ([PathFolder (buffer_name)]);
% Si el archivo de excel está abierto lo cierra
file = fopen(NameBF, 'a');
if file ~= -1
fclose(file);
else
while (file == -1)
wbkname = 'Buffer_Actual.xls';
h = actxGetRunningServer('Excel.Application');
h.WorkBooks.Item(wbkname).Save; h.WorkBooks.Item(wbkname).Close;
file = fopen(NameBF, 'a');
end
fclose(file);
end
senales = {'NWC' 'NPM' 'JJI' 'NAA' 'NLK' 'NML'};
xlswrite(NameBF, {fecha3}, 1, 'A1');
xlswrite(NameBF, senales, 1, 'A2');
xlswrite(NameBF, NWC', 1, 'A3');
xlswrite(NameBF, NPM', 1, 'B3');
xlswrite(NameBF, JJI', 1, 'C3');
xlswrite(NameBF, NAA', 1, 'D3');
xlswrite(NameBF, NLK', 1, 'E3');
xlswrite(NameBF, NML', 1, 'F3');
%Reiniciamos el contador para grabar datos temporales
cont=0;
end
texto=[' ',fecha3,'
',int2str(c(4)),':',int2str(c(5)),':',int2str(c(6)),' [',int2str(i),']
NWC='...
int2str(floor( NWC(i))) ,' NPM=', int2str(floor(NPM(i))),' JJI=',
int2str(floor(JJI(i)))...
' NAA=', int2str(floor(NAA(i))),' NLK=', int2str(floor(NLK(i))),'
NML=', int2str(floor(NML(i)))];
set(handles.text1, 'String',texto); % Imprimimos los datos en la barra
inferior del programa
k=get(handles.togglebutton1, 'Value'); % Verifica el estado del toogle
button
end
%% Eliminamos el canal de recepción de datos.
delete(AI);
set(handles.text1, 'String',' Esperando Datos'); % Imprimimos los
datos en la barra inferior del programa
```

```
% --------------------------------------------------------------------
----------
guidata(hObject,handles);
% --- Executes on button press in pb_with_bg.
function pb_with_bg_Callback(hObject, eventdata, handles)
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% --- Executes during object creation, after setting all properties.
function togglebutton1_CreateFcn(hObject, eventdata, handles)
% hObject handle to togglebutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns
called
% --- Executes on button press in togglebutton2.
function togglebutton2_Callback(hObject, eventdata, handles)
% hObject handle to togglebutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of togglebutton2
% --- Executes on button press in togglebutton3.
function togglebutton3_Callback(hObject, eventdata, handles)
% hObject handle to togglebutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of togglebutton3
% --- Executes on button press in radiobutton1.
function radiobutton1_Callback(hObject, eventdata, handles)
% --- Executes on button press in radiobutton2.
function radiobutton2_Callback(hObject, eventdata, handles)
% hObject handle to radiobutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of radiobutton2
% --- Executes on button press in pushbutton4.
```

```
function pushbutton4_Callback(hObject, eventdata, handles)
% --- Executes on button press in radiobutton3.
function radiobutton3_Callback(hObject, eventdata, handles)
% hObject handle to radiobutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of radiobutton3
function edit1_Callback(hObject, eventdata, handles)
% hObject handle to edit1 (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 edit1 as text
% str2double(get(hObject,'String')) returns contents of edit1 as a
double
% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit1 (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 button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% --------------------------------------------------------------------
function uitoggletool1_ClickedCallback(hObject, eventdata, handles)
% hObject handle to uitoggletool1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
function edit2_Callback(hObject, eventdata, handles)
% hObject handle to edit2 (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 edit2 as text
% str2double(get(hObject,'String')) returns contents of edit2 as a
double
% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit2 (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 button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton7 (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 pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton8 (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 checkbox1.
function checkbox1_Callback(hObject, eventdata, handles)
% hObject handle to checkbox1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hint: get(hObject,'Value') returns toggle state of checkbox1
function edit3_Callback(hObject, eventdata, handles)
% hObject handle to edit3 (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 edit3 as text
% str2double(get(hObject,'String')) returns contents of edit3 as a
double
```

```
% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit3 (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 config_menu_Callback(hObject, eventdata, handles)
% hObject handle to config_menu (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% --------------------------------------------------------------------
%%FUNCION PARA BOTON DE CONFIGURAR ENTRADA DE AUDIO MANUALMENTE
function menu_configurar_audioinput_Callback(hObject, eventdata,
handles)
salida= daqhwinfo('winsound');
tarjetas=salida.BoardNames;
in = listdlg('PromptString','Selecione la entrada de sonido',...
'SelectionMode','single',...
'ListString',tarjetas,'ListSize',[300 300]);
if ( isempty(in))
set(handles.togglebutton1,'Visible','off');
waitfor( errordlg('Debe selecionar un canal para poder iniciar la
aplicación',...
'Alerta!','OK'));
else
set(handles.togglebutton1,'Visible','on');
end
handles.in=in;
guidata(hObject,handles);
% --- Executes when figure1 is resized.
function figure1_ResizeFcn(hObject, eventdata, handles)
% hObject handle to figure1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% --------------------------------------------------------------------
```

```
function tools_menu_Callback(hObject, eventdata, handles)
% hObject handle to tools_menu (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% --------------------------------------------------------------------
function menu_tools_mover_Callback(hObject, eventdata, handles)
axes(handles.axes1)
zoom off
pan on
guidata(hObject,handles);
% --------------------------------------------------------------------
function menu_tools_zoom_Callback(hObject, eventdata, handles)
axes(handles.axes1)
zoom on
pan off
guidata(hObject,handles);
% --------------------------------------------------------------------
function menu_tools_cursor_Callback(hObject, eventdata, handles)
axes(handles.axes1)
datacursormode on
guidata(hObject,handles);
% --- Executes on button press in pushbutton10.
function pushbutton10_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton10 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% --------------------------------------------------------------------
function uipushtool4_ClickedCallback(hObject, eventdata, handles)
% --------------------------------------------------------------------
function uitoggletool9_OnCallback(hObject, eventdata, handles)
global l1;
l1=1;
hold off;
guidata(hObject,handles);
% --------------------------------------------------------------------
function uitoggletool9_OffCallback(hObject, eventdata, handles)
global l1;
l1=0;
hold off;
guidata(hObject,handles);
```

```
% --- Executes when user attempts to close figure1.
%% SE DEFINE CIERRE DE LA APLICACION
function figure1_CloseRequestFcn(hObject, eventdata, handles)
global k;
opc=questdlg('¿Desea salir del programa y guardar los cambios?',...
'SALIR','Si','No','No');
if strcmp(opc,'No')
return;
end
waitfor( warndlg('Por favor recuerde cerrar todos los archivos de
Excel asociados a este software y oprimir OK ',...
'Advertencia !','OK'));
GuardarDatos();
waitfor( warndlg('Datos Guardados Exitosamente!','Datos
Guardados','OK'));
delete(hObject);
% --------------------------------------------------------------------
%% FUNCIONES PARA GRAFICAR DATOS GUARDADOS
function uipushtool3_ClickedCallback(hObject, eventdata, handles)
[FileName Path]=uigetfile({'*.xls'},'Abrir Documento');
NameSave = ([Path FileName]);
SG=xlsread(NameSave,1, 'C2:C17281');
ejex =linspace(0,24,17280);
SG=SG';
figure(1)
hold off
plot(ejex(1:4320),SG(1:4320),'b')
hold on
plot(ejex(4321:12961),SG(4321:12961),'y')
hold on
plot(ejex(12962:17280),SG(12962:17280),'b')
grid on
hold on
guidata(hObject,handles);
% --------------------------------------------------------------------
function uipushtool5_ClickedCallback(hObject, eventdata, handles)
X=getframe(gca);
X=frame2im(X);
```

```
[FileName, PathName] = uiputfile('*.jpg', 'Save As');
imwrite(X,FileName,'jpg')
guidata(hObject,handles);
% ----------------------------------------------------------------------
function Open_ClickedCallback(hObject, eventdata, handles)
[FileName Path]=uigetfile({'*.xls'},'Abrir Documento');
NameSave = ([Path FileName]);
SG=xlsread(NameSave,1, 'C2:C17281');
ejex =linspace(0,24,17280);
SG=SG';
figure(1)
hold off
plot(ejex(1:4320),SG(1:4320),'b')
hold on
plot(ejex(4321:12961),SG(4321:12961),'y')
hold on
plot(ejex(12962:17280),SG(12962:17280),'b')
grid on
hold on
guidata(hObject,handles);
% ----------------------------------------------------------------------
function uipushtool2_ClickedCallback(hObject, eventdata, handles)
%% Se debe imprimir la imagen
X=getframe(gca);
printpreview()
guidata(hObject,handles);
% --- Executes on button press in pushbutton11.
function pushbutton11_Callback(hObject, eventdata, handles)
figure(2)
plot(handles.f,handles.Pxx,'m')
hold off
set(gca,'XTickLabel',{'0','5000','10000','15000','20000','25000','3000
0','35000','40000','45000','50000'})
grid on
xlabel('Frequency (Hz)'); ylabel('Power Spectrum (dB)');
guidata(hObject,handles);
```