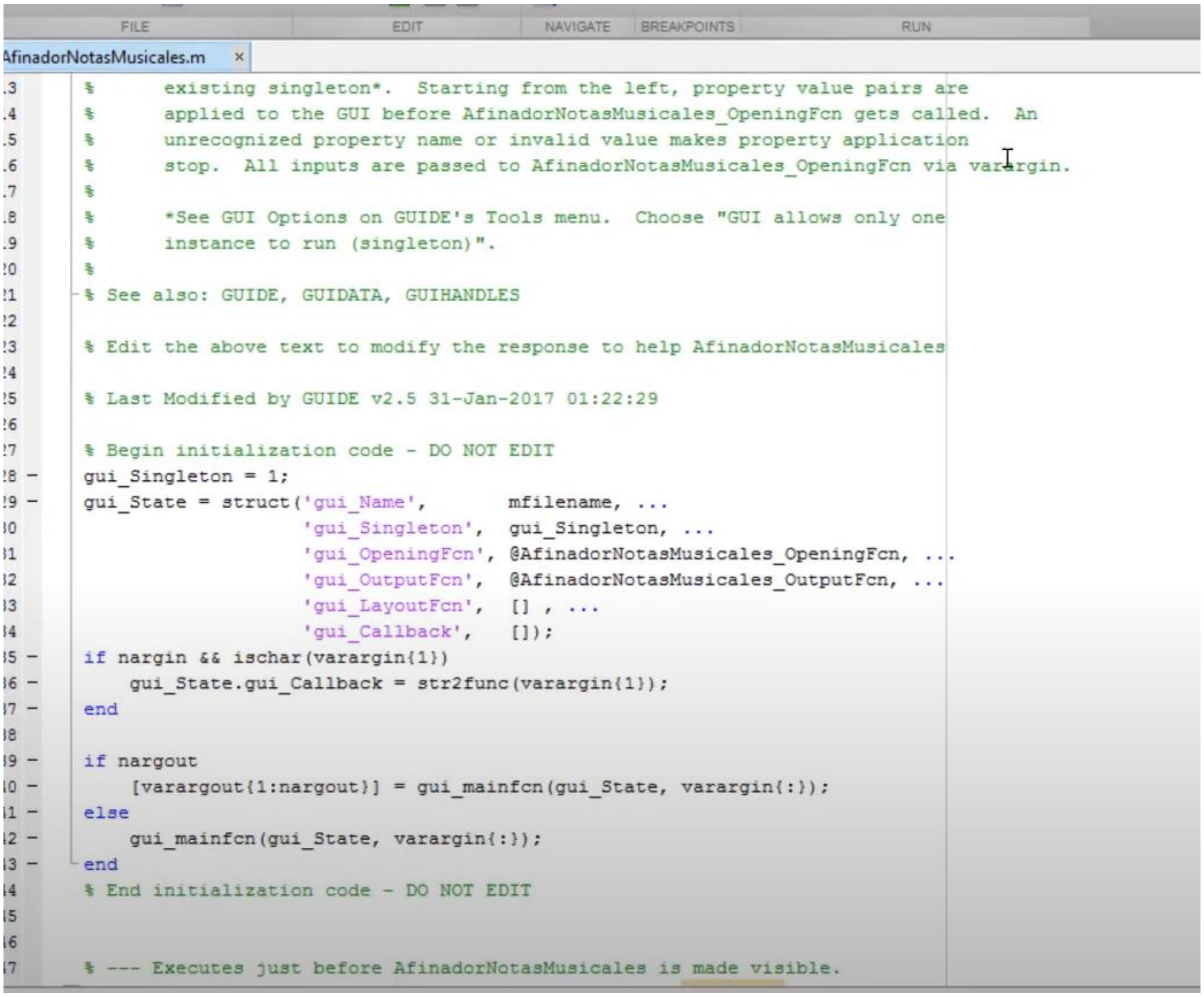
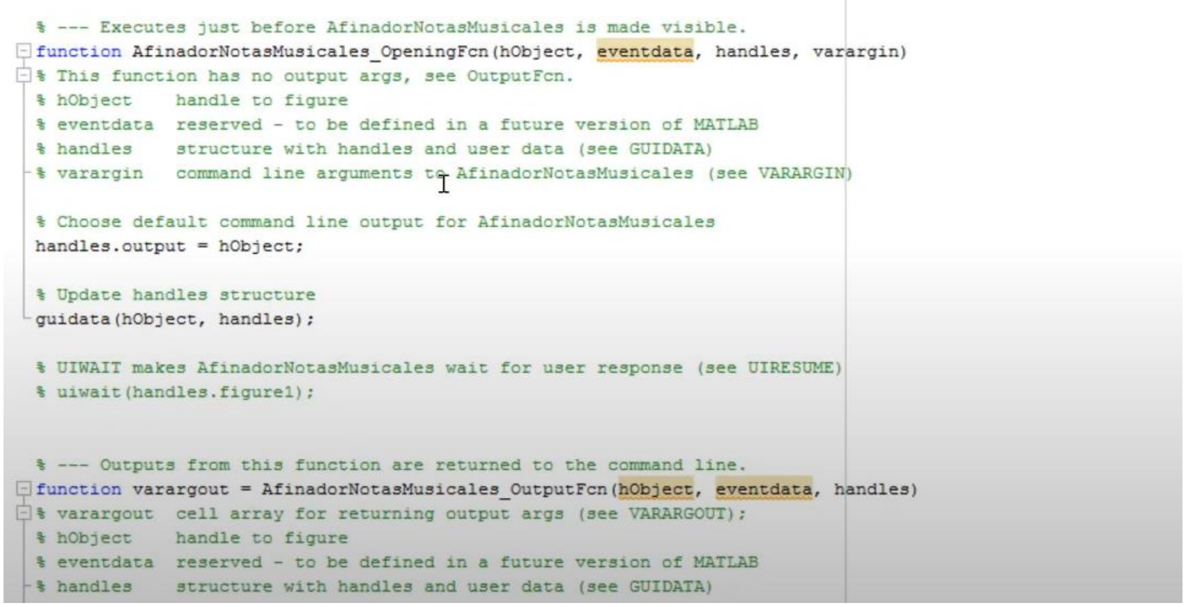
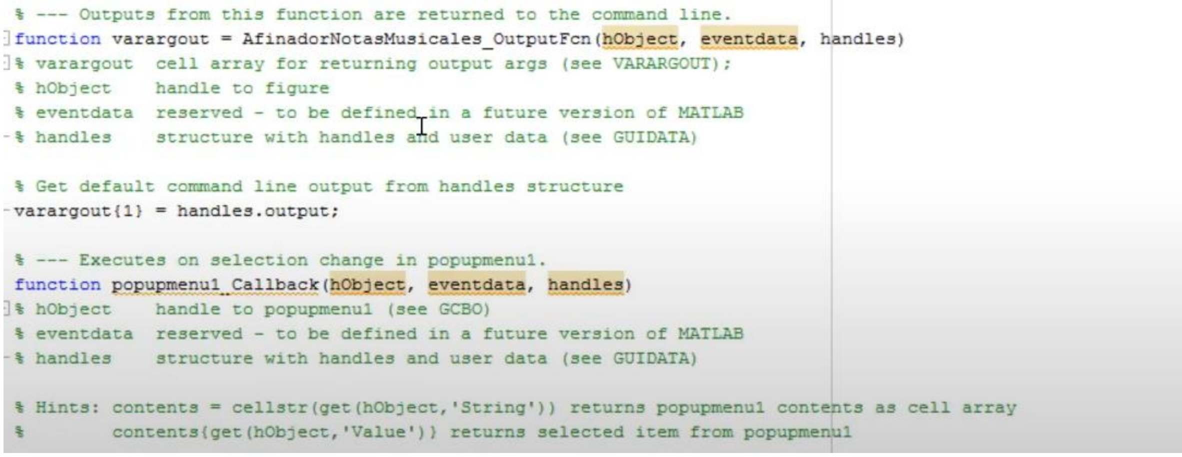


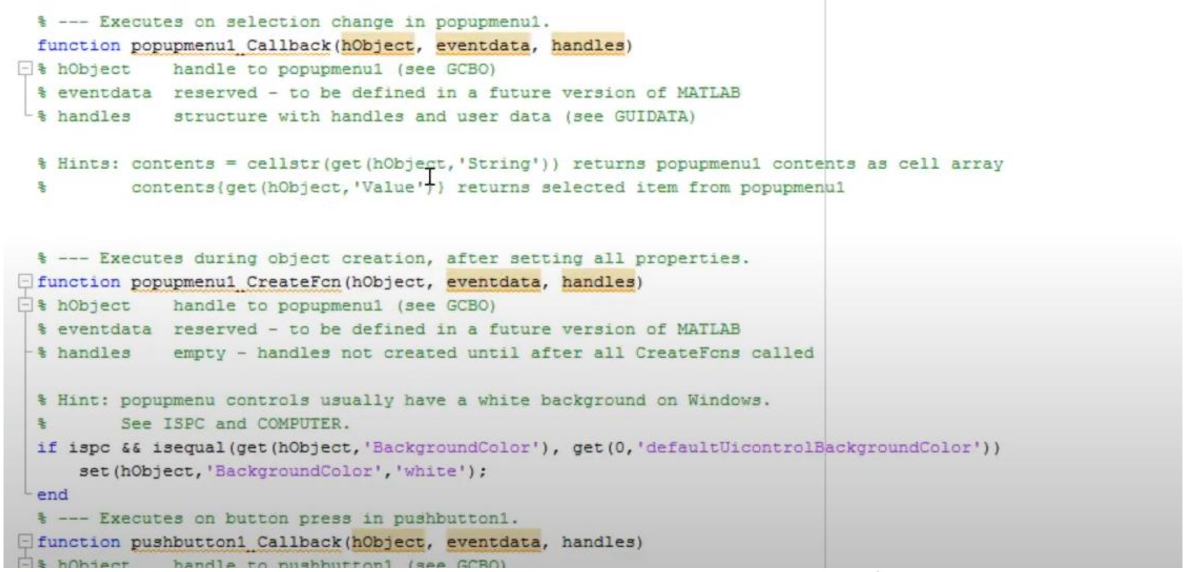
```
%       existing singleton*.  Starting from the left, property value pairs are
%       applied to the GUI before AfinadorNotasMusicales_OpeningFcn gets called.  An
%       unrecognized property name or invalid value makes property application
%       stop.  All inputs are passed to AfinadorNotasMusicales_OpeningFcn 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 AfinadorNotasMusicales

% Last Modified by GUIDE v2.5 31-Jan-2017 01:22:29

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @AfinadorNotasMusicales_OpeningFcn, ...
                   'gui_OutputFcn',  @AfinadorNotasMusicales_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 AfinadorNotasMusicales is made visible.
```



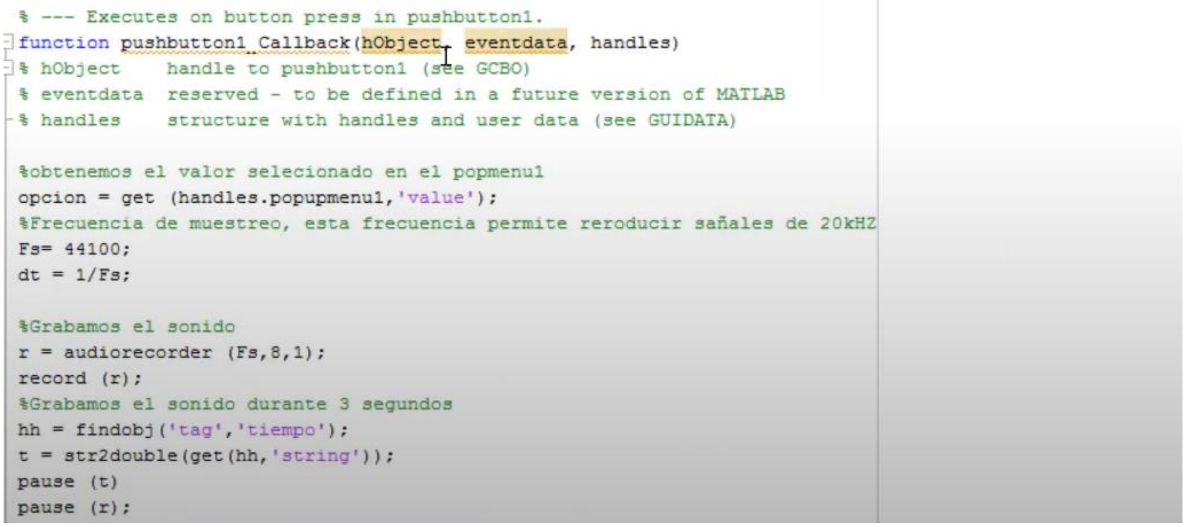
```
% --- Executes just before AfinadorNotasMusicales is made visible.
function AfinadorNotasMusicales_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB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

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

% Update handles structure
guidata(hObject, handles);

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


% --- Outputs from this function are returned to the command line.
function varargout = AfinadorNotasMusicales_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)
```



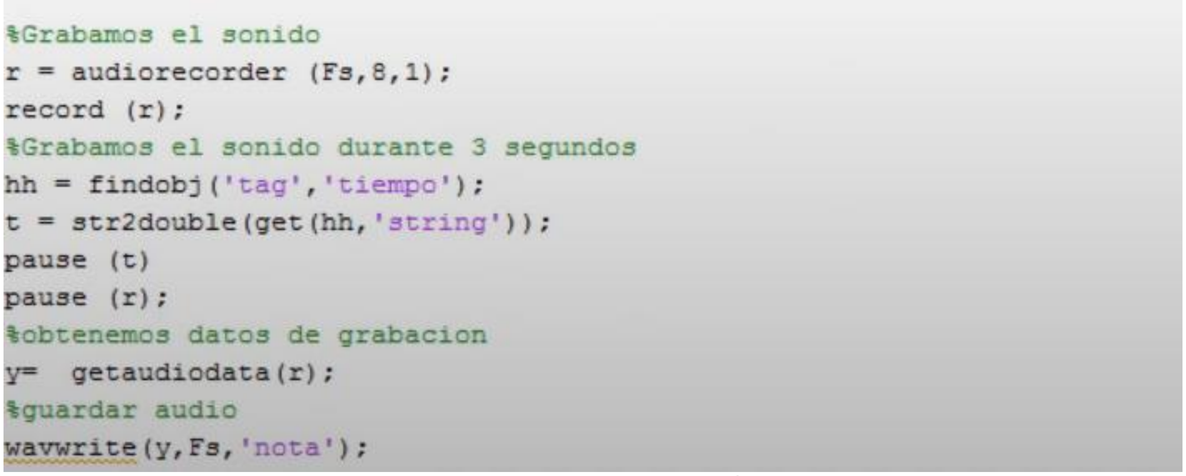
```
% --- Outputs from this function are returned to the command line.
function varargout = AfinadorNotasMusicales_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future 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 selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (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 popupmenu1 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenu1
```



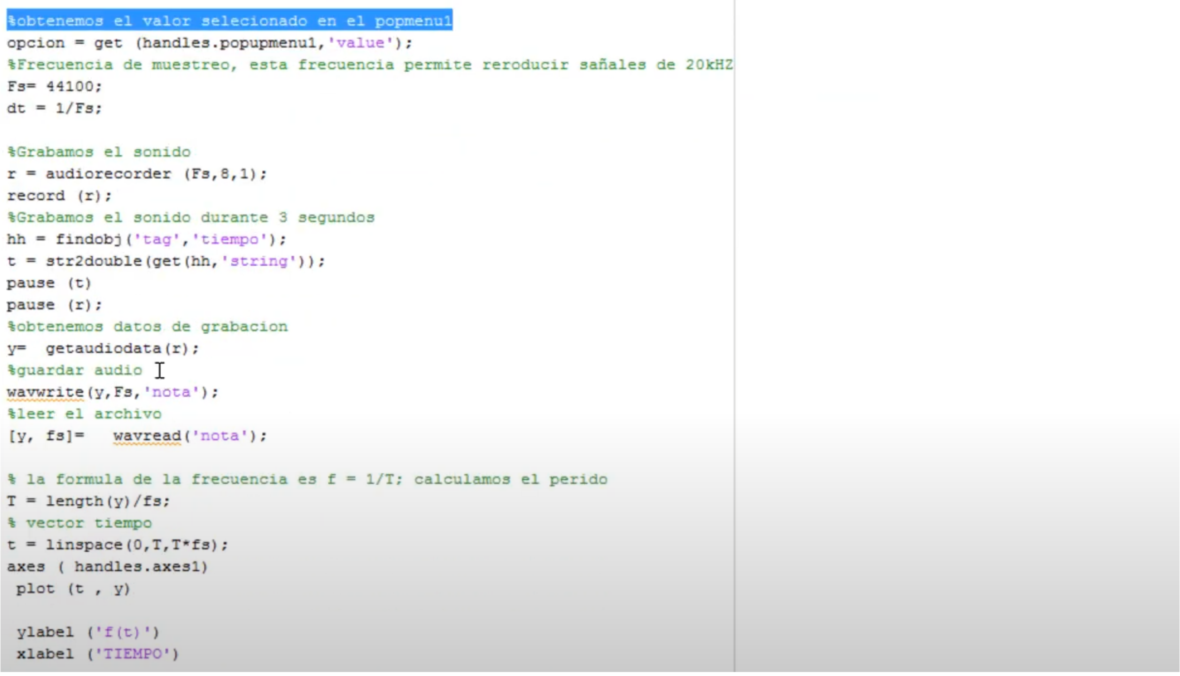
```
% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenu1


% --- 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.
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)
% hObject    handle to pushbutton1 (see GCBO)
```



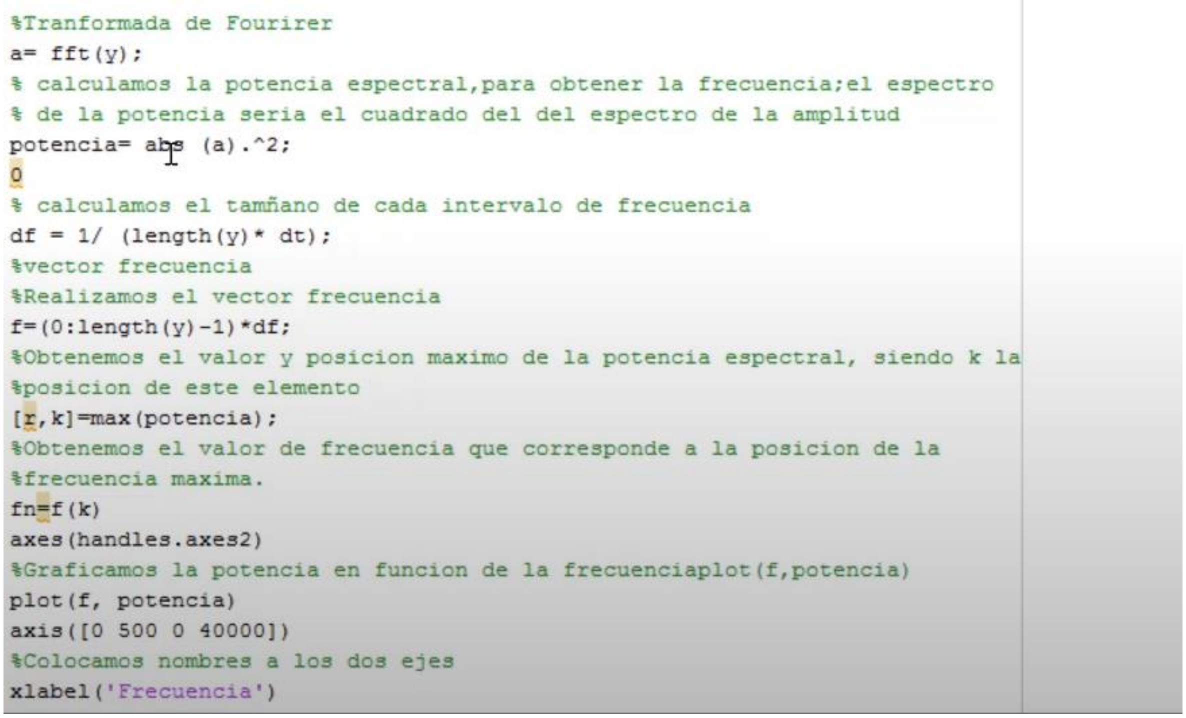
```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%obtenemos el valor selecionado en el popmenu1
opcion = get (handles.popupmenu1,'value');
%Frecuencia de muestreo, esta frecuencia permite reroducir sañales de 20kHZ
Fs= 44100;
dt = 1/Fs;

%Grabamos el sonido
r = audiorecorder (Fs,8,1);
record (r);
%Grabamos el sonido durante 3 segundos
hh = findobj('tag','tiempo');
t = str2double(get(hh,'string'));
pause (t)
pause (r);
```



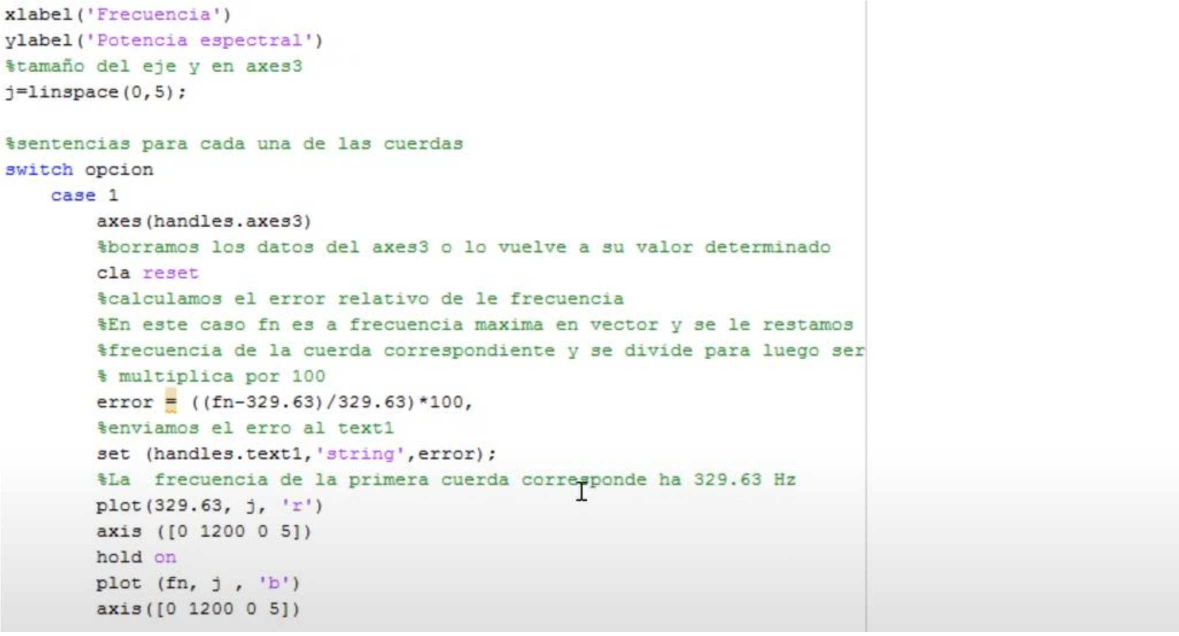
```
%Grabamos el sonido
r = audiorecorder (Fs,8,1);
record (r);
%Grabamos el sonido durante 3 segundos
hh = findobj('tag','tiempo');
t = str2double(get(hh,'string'));
pause (t)
pause (r);
%obtenemos datos de grabacion
y=  getaudiodata(r);
%guardar audio
wavwrite(y,Fs,'nota');
```



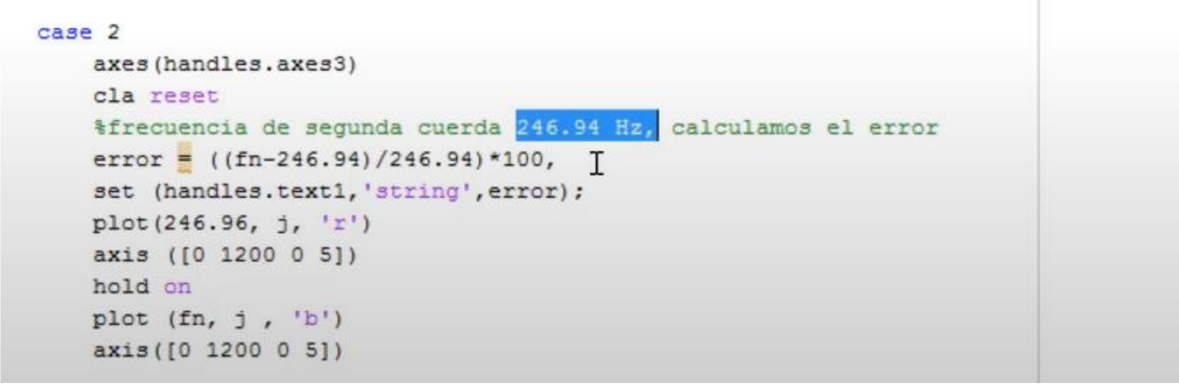
```
%obtenemos el valor selecionado en el popmenu1
opcion = get (handles.popupmenu1,'value');
%Frecuencia de muestreo, esta frecuencia permite reroducir sañales de 20kHZ
Fs= 44100;
dt = 1/Fs;

%Grabamos el sonido
r = audiorecorder (Fs,8,1);
record (r);
%Grabamos el sonido durante 3 segundos
hh = findobj('tag','tiempo');
t = str2double(get(hh,'string'));
pause (t)
pause (r);
%obtenemos datos de grabacion
y=  getaudiodata(r);
%guardar audio
wavwrite(y,Fs,'nota');
%leer el archivo
[y, fs]=   wavread('nota');

% la formula de la frecuencia es f = 1/T; calculamos el perido
T = length(y)/fs;
% vector tiempo
t = linspace(0,T,T*fs);
axes ( handles.axes1)
 plot (t , y)

 ylabel ('f(t)')
 xlabel ('TIEMPO')
```



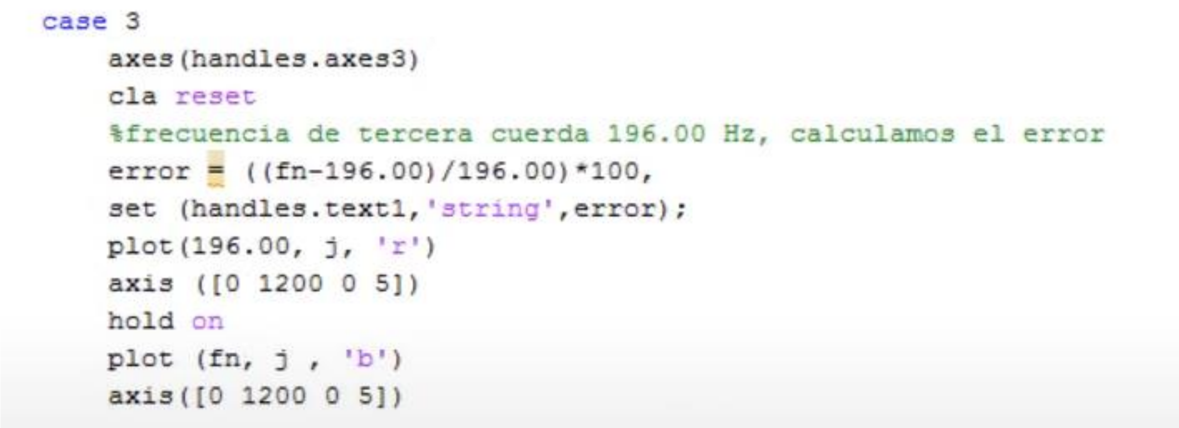
```
%Tranformada de Fourirer
a= fft(y);
% calculamos la potencia espectral,para obtener la frecuencia;el espectro
% de la potencia seria el cuadrado del del espectro de la amplitud
potencia= abs (a).^2;
0
% calculamos el tamñano de cada intervalo de frecuencia
df = 1/ (length(y)* dt);
%vector frecuencia
%Realizamos el vector frecuencia
f=(0:length(y)-1)*df;
%Obtenemos el valor y posicion maximo de la potencia espectral, siendo k la
%posicion de este elemento
[r,k]=max(potencia);
%Obtenemos el valor de frecuencia que corresponde a la posicion de la
%frecuencia maxima.
fn=f(k)
axes(handles.axes2)
%Graficamos la potencia en funcion de la frecuenciaplot(f,potencia)
plot(f, potencia)
axis([0 500 0 40000])
%Colocamos nombres a los dos ejes
xlabel('Frecuencia')
```



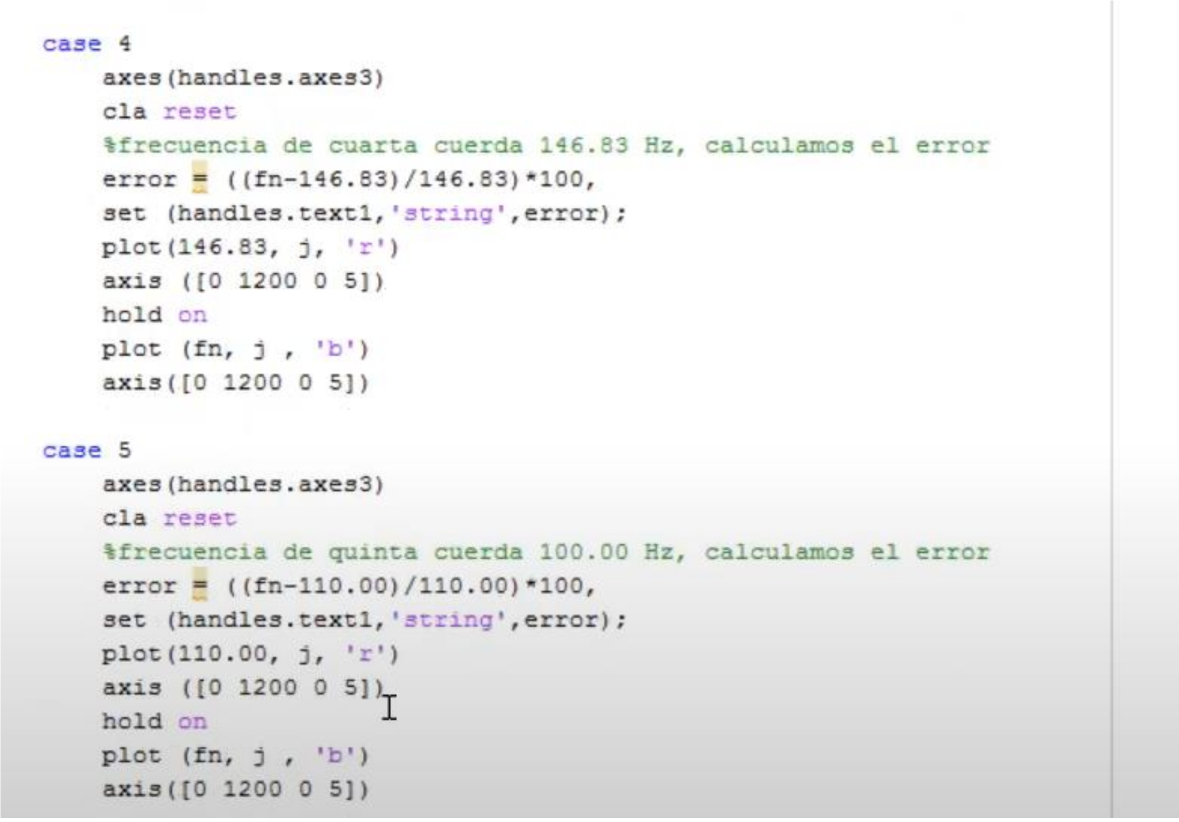
```
xlabel('Frecuencia')
ylabel('Potencia espectral')
%tamaño del eje y en axes3
j=linspace(0,5);

%sentencias para cada una de las cuerdas
switch opcion
    case 1
        axes(handles.axes3)
        %borramos los datos del axes3 o lo vuelve a su valor determinado
        cla reset
        %calculamos el error relativo de le frecuencia
        %En este caso fn es a frecuencia maxima en vector y se le restamos
        %frecuencia de la cuerda correspondiente y se divide para luego ser
        % multiplica por 100
        error = ((fn-329.63)/329.63)*100,
        %enviamos el erro al text1
        set (handles.text1,'string',error);
        %La  frecuencia de la primera cuerda corresponde ha 329.63 Hz
        plot(329.63, j, 'r')
        axis ([0 1200 0 5])
        hold on
        plot (fn, j , 'b')
        axis([0 1200 0 5])
```



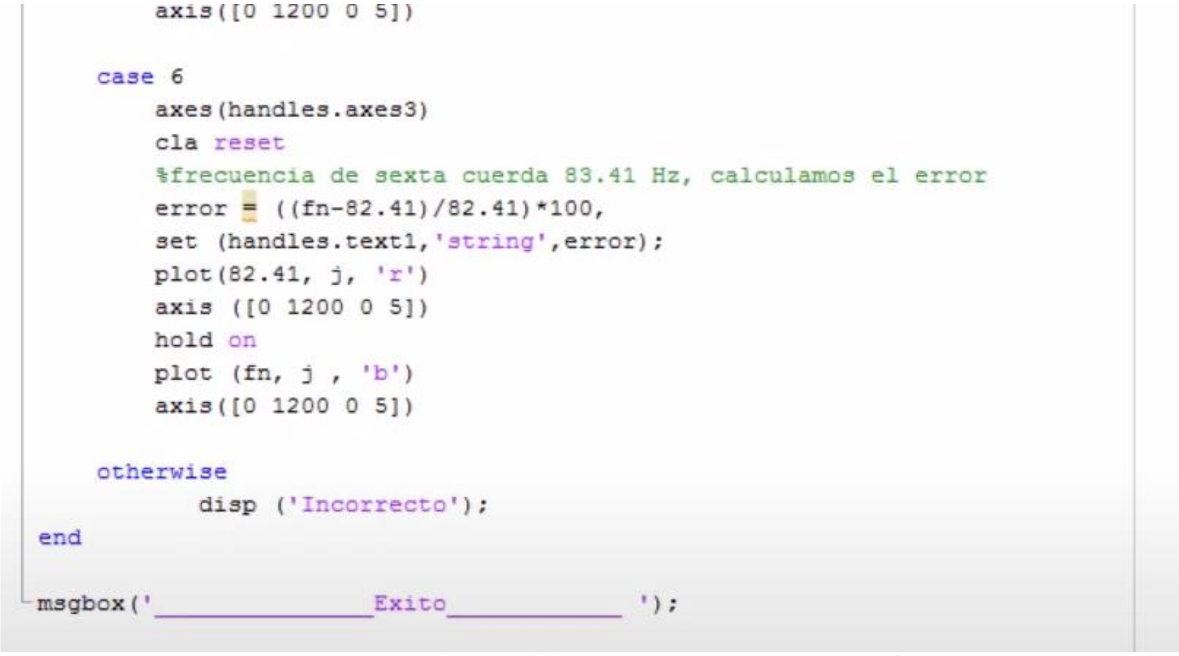
```
case 2
    axes(handles.axes3)
    cla reset
    %frecuencia de segunda cuerda 246.94 Hz, calculamos el error
    error = ((fn-246.94)/246.94)*100,
    set (handles.text1,'string',error);
    plot(246.96, j, 'r')
    axis ([0 1200 0 5])
    hold on
    plot (fn, j , 'b')
    axis([0 1200 0 5])
```



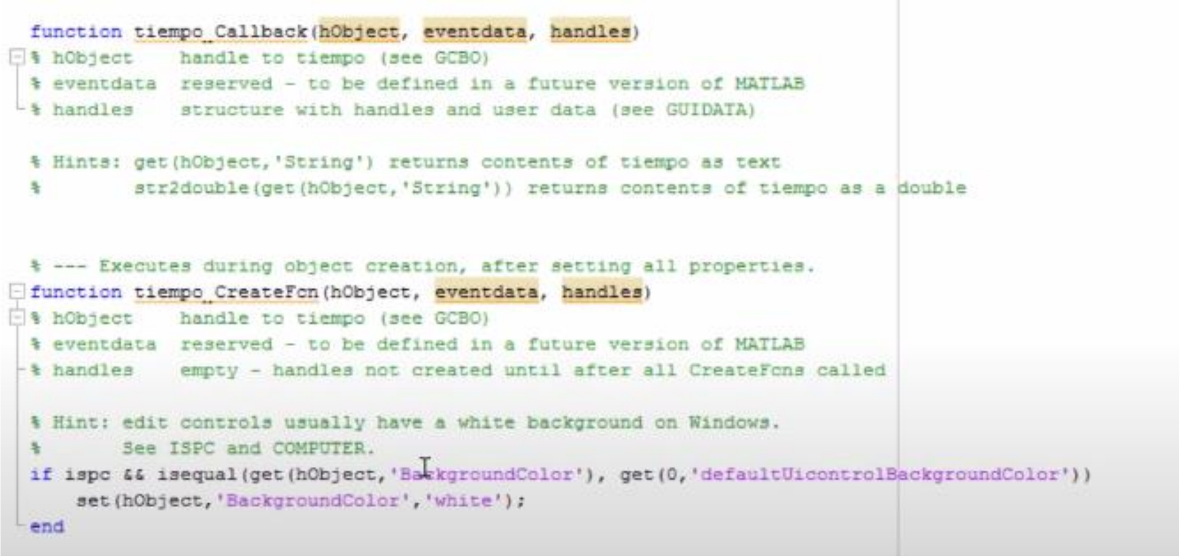
```
case 3
    axes(handles.axes3)
    cla reset
    %frecuencia de tercera cuerda 196.00 Hz, calculamos el error
    error = ((fn-196.00)/196.00)*100,
    set (handles.text1,'string',error);
    plot(196.00, j, 'r')
    axis ([0 1200 0 5])
    hold on
    plot (fn, j , 'b')
    axis([0 1200 0 5])
```



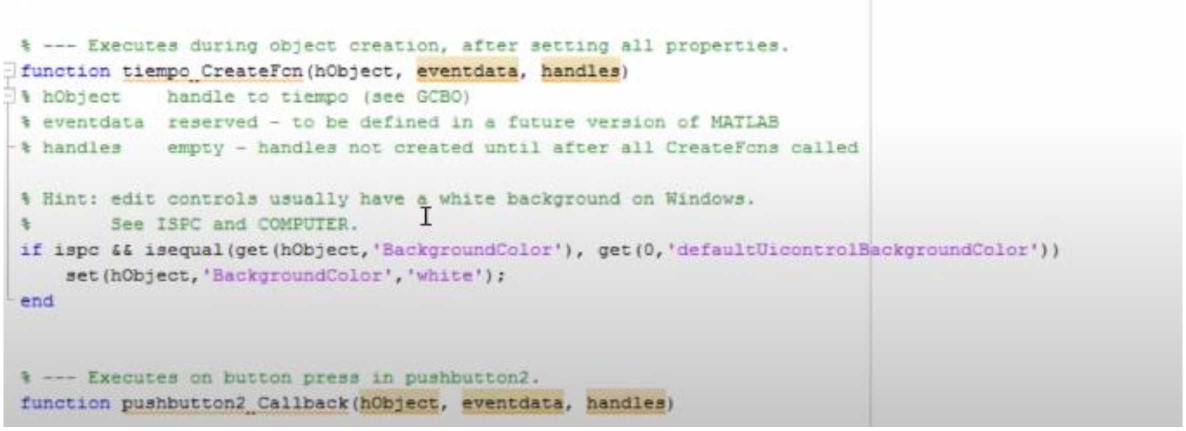
```
case 4
    axes(handles.axes3)
    cla reset
    %frecuencia de cuarta cuerda 146.83 Hz, calculamos el error
    error = ((fn-146.83)/146.83)*100,
    set (handles.text1,'string',error);
    plot(146.83, j, 'r')
    axis ([0 1200 0 5])
    hold on
    plot (fn, j , 'b')
    axis([0 1200 0 5])

case 5
    axes(handles.axes3)
    cla reset
    %frecuencia de quinta cuerda 100.00 Hz, calculamos el error
    error = ((fn-110.00)/110.00)*100,
    set (handles.text1,'string',error);
    plot(110.00, j, 'r')
    axis ([0 1200 0 5])
    hold on
    plot (fn, j , 'b')
    axis([0 1200 0 5])
```



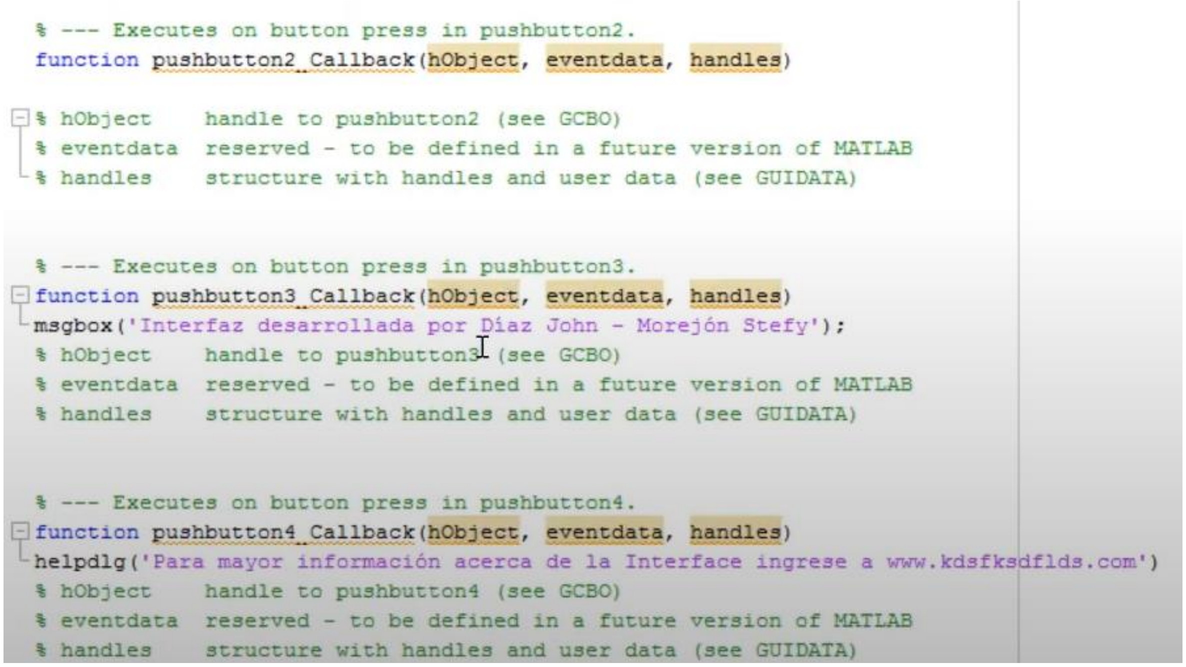
```
        axis([0 1200 0 5])

    case 6
        axes(handles.axes3)
        cla reset
        %frecuencia de sexta cuerda 83.41 Hz, calculamos el error
        error = ((fn-82.41)/82.41)*100,
        set (handles.text1,'string',error);
        plot(82.41, j, 'r')
        axis ([0 1200 0 5])
        hold on
        plot (fn, j , 'b')
        axis([0 1200 0 5])

    otherwise
           disp ('Incorrecto');
end

msgbox('_______________Exito____________ ');
```



```
function tiempo_Callback(hObject, eventdata, handles)
% hObject    handle to tiempo (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 tiempo as text
%        str2double(get(hObject,'String')) returns contents of tiempo as a double


% --- Executes during object creation, after setting all properties.
function tiempo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tiempo (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 during object creation, after setting all properties.
function tiempo_CreateFcn(hObject, eventdata, handles)
% hObject    handle to tiempo (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 pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
```



```
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)

% hObject    handle to pushbutton2 (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 pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
msgbox('Interfaz desarrollada por Díaz John - Morejón Stefy');
% hObject    handle to pushbutton3 (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 pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
helpdlg('Para mayor información acerca de la Interface ingrese a www.kdsfksdflds.com')
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```