

# Tarea 4

Alejandro José, Luncey Contreras, 202112396  
*Escuela de Mecánica Eléctrica, Facultad de Ingeniería, Universidad de San Carlos de Guatemala*

En esta tarea se realizó un programa capaz de almacenar, reproducir y graficar audio de manera simultánea, ya sea en su amplitud o en el espectro de frecuencia. Para ello se realizó una interfaz de usuario con las diversas opciones y una entrada de elección.

## I. CÓDIGO

```
if(exist('OCTAVE_VERSION', 'builtin')~=0)
    pkg load signal;
end

opcion=0;
while opcion~=5
    disp('Seleccione una opción:')
    disp('1.Grabar')
    disp('2.Reproducir')
    disp('3.Graficar')
    disp('4.Graficar densidad')
    disp('5.Salir')
    opcion=input('Ingrese su elección: ');
    switch opcion
        case 1
            try
                duracion=input('Ingrese la duración de la grabación en segundos: ');
                disp('Comenzando la grabación...');
                recObj=audiorecorder;
                recordblocking(recObj,duracion);
                disp('Grabación finalizada. ');
                data=getaudiodata(recObj);
                audiowrite('audio.wav',data,recObj.SampleRate);
                disp('Archivo de audio grabado correctamente. ');
            catch
                disp('Error al grabar el audio. ');
            end
        case 2
            try
                [data,fs]=audioread('audio.wav');
                sound(data,fs);
            catch
                disp('Error al reproducir el audio. ');
            end
        case 3
            try
                [data,fs]=audioread('audio.wav');
                tiempo=linspace(0,length(data)/fs,length(data));
                plot(tiempo,data);
                xlabel('Tiempo (s)');
                ylabel('Amplitud');
                title('Audio');
            catch
                disp('Error al graficar el audio. ');
            end
        case 4
            try
                disp('Graficando espectro de frecuencia...');
                [audio,fs]=audioread('audio.wav');
                N=length(audio);
                f=linspace(0,Fs/2,N/2+1);
                ventana=hann(N);
                Sxx=pwelch(audio,ventana,0,N,Fs);
                plot(f,10*log10(Sxx(1:N/2+1)));
                xlabel('Frecuencia (Hz)');
                ylabel('Densidad espectralde potencia (dB/Hz)');
                title('Espectro de frecuencia de la señal grabada');
            catch
                disp('Error al graficar el audio. ');
            end
        case 5
            disp('Saliedo del programa...');
            otherwise
                disp('Opción no válida. ');
            end
    end
end
```

Figura 1: Código en gedit, elaboración propia

```
catch
    disp('Error al grabar el audio. ');
end
case 2
try
    [data,fs]=audioread('audio.wav');
    sound(data,fs);
catch
    disp('Error al reproducir el audio. ');
end
case 3
try
    [data,fs]=audioread('audio.wav');
    tiempo=linspace(0,length(data)/fs,length(data));
    plot(tiempo,data);
    xlabel('Tiempo (s)');
    ylabel('Amplitud');
    title('Audio');
catch
    disp('Error al graficar el audio. ');
end
case 4
try
    disp('Graficando espectro de frecuencia...');
    [audio,fs]=audioread('audio.wav');
    N=length(audio);
    f=linspace(0,Fs/2,N/2+1);
    ventana=hann(N);
    Sxx=pwelch(audio,ventana,0,N,Fs);
    plot(f,10*log10(Sxx(1:N/2+1)));
    xlabel('Frecuencia (Hz)');
    ylabel('Densidad espectralde potencia (dB/Hz)');
    title('Espectro de frecuencia de la señal grabada');
catch
    disp('Error al graficar el audio. ');
end
case 5
    disp('Saliedo del programa...');
otherwise
    disp('Opción no válida. ');
end
end
```

Figura 2: Código en gedit (continuación), elaboración propia

```
if(exist('OCTAVE_VERSION', 'builtin')~=0)
    pkg load signal;
```

```
end
opcion=0;
while opcion~=5
    disp('Seleccione una opción:')
    disp('1.Grabar')
    disp('2.Reproducir')
    disp('3.Graficar')
    disp('4.Graficar densidad')
    disp('5.Salir')
    opcion=input('Ingrese su elección: ');
    switch opcion
        case 1
            try
                duracion=input('Ingrese la duración de la grabación en segundos: ');
                disp('Comenzando la grabación...');
                recObj=audiorecorder;
                recordblocking(recObj,duracion);
                disp('Grabación finalizada. ');
                data=getaudiodata(recObj);
                audiowrite('audio.wav',data,recObj.SampleRate);
                disp('Archivo de audio grabado correctamente. ');
            catch
                disp('Error al grabar el audio. ');
            end
        case 2
            try
                [data,fs]=audioread('audio.wav');
                sound(data,fs);
            catch
                disp('Error al reproducir el audio. ');
            end
        case 3
            try
                [data,fs]=audioread('audio.wav');
                tiempo=linspace(0,length(data)/fs,length(data));
                plot(tiempo,data);
                xlabel('Tiempo (s)');
                ylabel('Amplitud');
                title('Audio');
            catch
                disp('Error al graficar el audio. ');
            end
        case 4
            try
                disp('Graficando espectro de frecuencia...');
                [audio,fs]=audioread('audio.wav');
                N=length(audio);
                f=linspace(0,Fs/2,N/2+1);
                ventana=hann(N);
                Sxx=pwelch(audio,ventana,0,N,Fs);
                plot(f,10*log10(Sxx(1:N/2+1)));
                xlabel('Frecuencia (Hz)');
                ylabel('Densidad espectralde potencia (dB/Hz)');
                title('Espectro de frecuencia de la señal grabada');
            catch
                disp('Error al graficar el audio. ');
            end
        case 5
            disp('Saliedo del programa...');
            otherwise
                disp('Opción no válida. ');
            end
    end
end
```

```

end
case 5
disp('Saliendo del programa...');
otherwise
disp('Opción no válida.');
```

```

end
end
```

El código fue ejecutado directamente en la terminal de Ubuntu en su versión 22.04 para el subsistema de linux que dispone Windows (WSL).

En el código se crearon las distintas opciones dentro de un ciclo while que permitiera delimitar las opciones de entrada, para cada uno de los casos se crearon las funciones necesarias para el manejo y procesamiento de la señal auditiva. Para ello fue necesario que los paquetes de *signal* y *control* fueran instalados en Octave.

## II. RESULTADOS

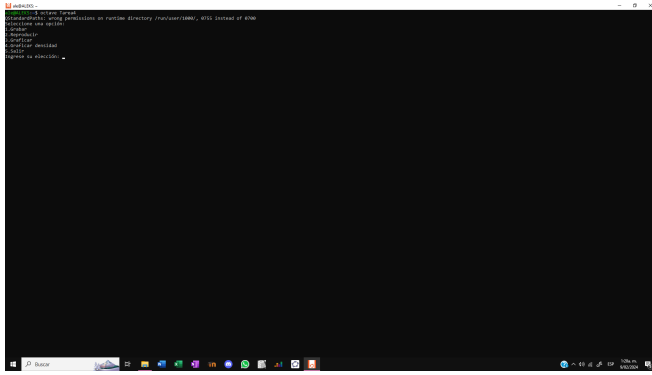


Figura 3: Interfaz de usuario, elaboración propia

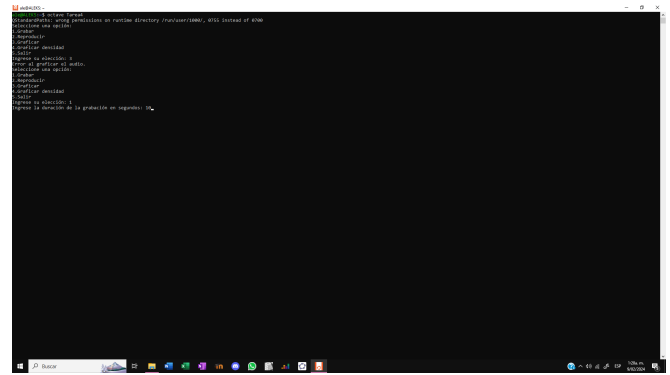


Figura 4: selección de opciones, elaboración propia

En la imagen podemos observar la interfaz que se muestra al usuario y con la cual es capaz de trabajar con el programa.