

TAREA 4: Procesamiento digital de señales-Grabadora de audio.*

Edgar David Barrios Franco, 201906465.¹

¹Facultad de Ingeniería, Escuela de Mecánica Eléctrica, Universidad de San Carlos de Guatemala.

I. RESUMEN.

Se utilizó Octave para realizar un programa capaz de realizar una serie de funciones las cuales pueden ser elegidas por el usuario a través de un menú, dichas opciones son: grabar, reproducir, graficar la densidad de audio y salir de la aplicación. Además el usuario puede establecer la duración de la grabación, esto marcará una diferencia en todos los gráficos que el programa pueda realizar.

II. CÓDIGO.

```
1 %Comprobación de ejecución de Octave
2
3 if (exist('OCTAVE_VERSION', 'builtin') ~=0)
4     pkg load signal;
5 end
6
7 %Menú principal
8
9 opcion = 0;
10 while opcion ~=5
11     %opcion = input('Seleccione una opcion:\n 1.Grabar audio\n 2.Reproducir audio\n 3.Graficar audio')
12     %Menu de opciones
13     disp('Seleccione una opcion: ')
14     disp('1. Grabar')
15     disp('2. Reproducir')
16     disp('3. Graficar')
17     disp('4. Graficar densidad')
18     disp('5. Salir')
19     opcion = input('Ingrese su elección: ');
20 switch opcion
21 case 1 %Grabación de audio
22     try
23         duracion = input('Ingrese la duración de la grabación en segundos: ');
24         disp('Comenzando la grabación....');
25         recObj = audiorecorder;
26         recordblocking(recObj, duracion);
27         disp('Grabación finalizada. ');
28         data = getaudiodata(recObj);
29         audiowrite('audio.wav', data, recObj.SampleRate);
30         disp('Archivo de audio grabado correctamente. ');
31     catch
```

Figura 1: Código en Octave.

```
32     disp('Error al grabar el audio. ');
33     end_try_catch
34
35 case 2 %Reproducción de audio
36     try
37         [data, fs] = audioread('audio.wav');
38         sound(data, fs);
39     catch
40         disp('Error al reproducir el audio. ');
41     end_try_catch
42
43 case 3 %Gráfico de audio
44     try
45         [data, fs] = audioread('audio.wav');
46         tiempo = linspace(0, length(data)/fs, length(data));
47         plot(tiempo, data);
48         xlabel('Tiempo (s)');
49         ylabel('Amplitud');
50         title('Audio');
51     catch
52         disp('Error al graficar el audio. ');
53     end_try_catch
54
55 case 4 %Graficando el espectro de frecuencia
56     try
57         disp('Graficando espectro de frecuencia....');
58         [audio, Fs] = audioread('audio.wav'); %Lee la señal desde el archivo .wav
59         N = length(audio); %Numero de muestras de la señal
60         f = linspace(0, Fs/2, N/2+1); %Vector de frecuencias
61         ventana = hann(N); %Ventana de Hann para reducir el efecto de las discontinuidades al calcul
62         Sxx = pwelch(audio, ventana, 0, N, Fs); %Densidad espectral de potencia
```

Figura 2: Código en Octave.

```
49         ylabel('Amplitud');
50         title('Audio');
51     catch
52         disp('Error al graficar el audio. ');
53     end_try_catch
54
55 case 4 %Graficando el espectro de frecuencia
56     try
57         disp('Graficando espectro de frecuencia....');
58         [audio, Fs] = audioread('audio.wav'); %Lee la señal desde el archivo .wav
59         N = length(audio); %Numero de muestras de la señal
60         f = linspace(0, Fs/2, N/2+1); %Vector de frecuencias
61         ventana = hann(N); %Ventana de Hann para reducir el efecto de las discontinuidades al calcul
62         Sxx = pwelch(audio, ventana, 0, N, Fs); %Densidad espectral de potencia
63         plot(f, 10*log10(Sxx(1:N/2+1))); %Gráfica del espectro de frecuencia en dB
64         xlabel('Frecuencia (Hz)');
65         ylabel('Densidad espectral de potencia (dB/Hz)');
66         title('Espectro de frecuencia de la señal grabada');
67     catch
68         disp('Error al graficar el audio. ');
69     end_try_catch
70
71 case 5 %Salir
72     disp('Saliendo del programa....');
73     break
74 otherwise
75     disp('Opción no válida....');
76 end
77 end
```

Figura 3: Código en Octave..

A. Resultados obtenidos.

```
>> GRABADORA
Seleccione una opcion:
1. Grabar
2. Reproducir
3. Graficar
4. Graficar densidad
5. Salir
Ingrese su elección: 1
Ingrese la duración de la grabación en segundos: 10
Comenzando la grabación....
Grabación finalizada.
Archivo de audio grabado correctamente.
Seleccione una opcion:
1. Grabar
2. Reproducir
3. Graficar
4. Graficar densidad
5. Salir
Ingrese su elección: 2
Seleccione una opcion:
1. Grabar
2. Reproducir
3. Graficar
4. Graficar densidad
5. Salir
Ingrese su elección: 3
Seleccione una opcion:
1. Grabar
2. Reproducir
3. Graficar
4. Graficar densidad
5. Salir
Ingrese su elección: 4
Graficando espectro de frecuencia....
Seleccione una opcion:
1. Grabar
2. Reproducir
3. Graficar
4. Graficar densidad
5. Salir
```

Figura 4: Resultados vistos desde la ventana de comandos de Octave.

* Proyectos de computación aplicados a Ingeniería Electrónica.

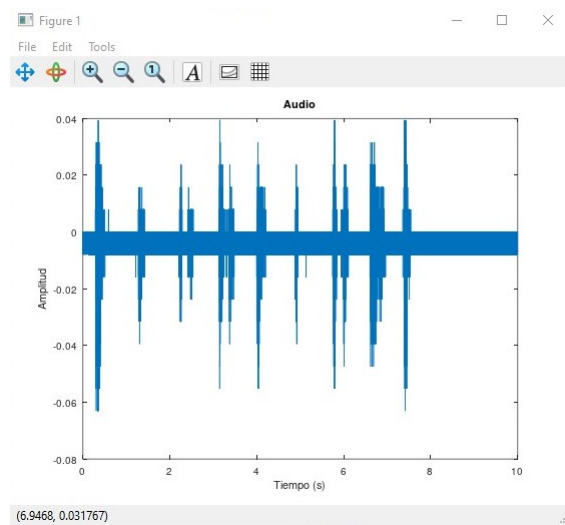


Figura 5: Gráfica de audio.

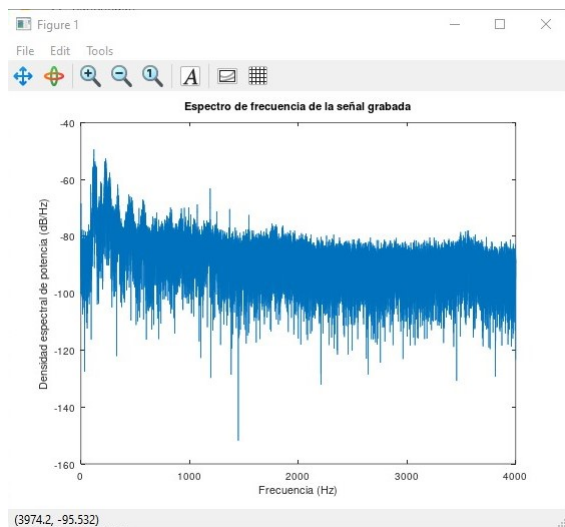


Figura 6: Espectro de frecuencia de audio.

Al desarrollar este programa, en la salida podemos observar un menú de opciones el cual nos permitirá: realizar una grabación, establecer el tiempo de duración en segundos que tendrá nuestra grabación, una vez establecido el tiempo el programa comenzará a grabar, al llegar al tiempo establecido, podremos obtener una gráfica del audio que hayamos captado y un espectro de frecuencia del mismo audio. Todas estas acciones las haremos mediante un CASE el cual es una estructura de control que se utiliza para comparar el valor de una expresión con una lista de valores posibles y ejecutar bloques de código específicos dependiendo de qué valor coincida con la expresión.