

VOCALES

CARGUE LAS SEÑALES

```
%Funcion para cargar archivos
a1= audioread('abajo.wav');
a2= audioread('arbol.wav');
a3= audioread('arriba.wav');
i1= audioread('iglesia.wav');
i2= audioread('inteligente.wav');
i3= audioread('ingresar.wav');
o1= audioread('oso.wav');
o2= audioread('orden.wav');
o3= audioread('organizar.wav');
```

Visualize las señales

```
figure(1)

subplot(3,3,1)
plot(a1)
xlabel('Tiempo')
ylabel('Amplitud')
title('abajo')

subplot(3,3,2)
plot(a2)
xlabel('Tiempo')
ylabel('Amplitud')
title('arbol')

subplot(3,3,3)
plot(a3)
xlabel('Tiempo')
ylabel('Amplitud')
title('arriba')

subplot(3,3,4)
plot(i1)
xlabel('Tiempo')
ylabel('Amplitud')
title('iglesia')

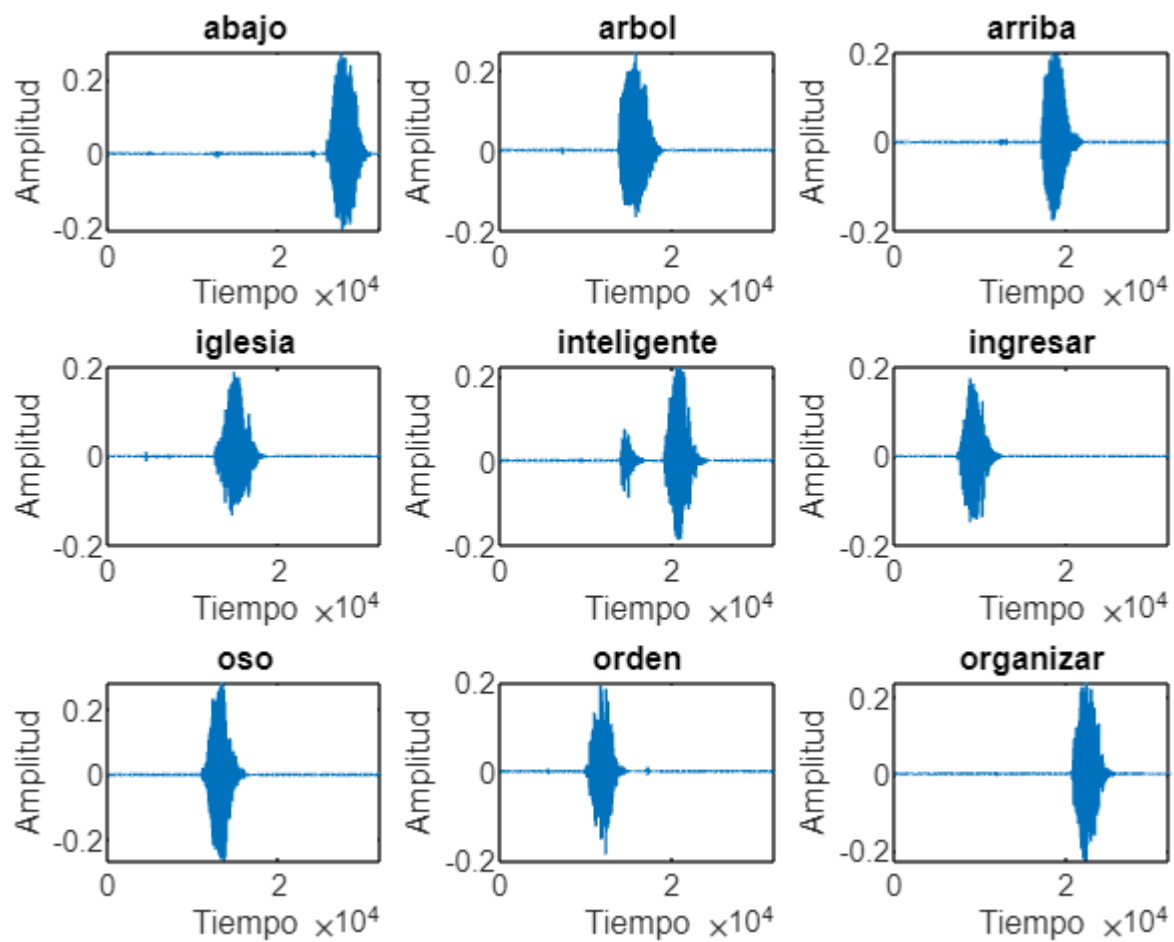
subplot(3,3,5)
plot(i2)
xlabel('Tiempo')
ylabel('Amplitud')
title('inteligente')
```

```
subplot(3,3,6)
plot(i3)
xlabel('Tiempo')
ylabel('Amplitud')
title('ingresar')
```

```
subplot(3,3,7)
plot(o1)
xlabel('Tiempo')
ylabel('Amplitud')
title('oso')
```

```
subplot(3,3,8)
plot(o2)
xlabel('Tiempo')
ylabel('Amplitud')
title('orden')
```

```
subplot(3,3,9)
plot(o3)
xlabel('Tiempo')
ylabel('Amplitud')
title('organizar')
```



CALCULAR

Energía

```
E_A1= sum(a1.^2)
```

```
E_A1 = 15.9885
```

```
E_I1= sum(i1.^2)
```

```
E_I1 = 11.0237
```

```
E_O3= sum(o3.^2)
```

```
E_O3 = 21.9810
```

Transformada de Fourier

```
n=32000
```

```
n = 32000
```

```
spec_a1 = fft(a1,n);
```

```
spec_a2 = fft(a2,n);
spec_a3 = fft(a3,n);
spec_i1 = fft(i1,n);
spec_i2 = fft(i2,n);
spec_i3 = fft(i3,n);
spec_o1 = fft(o1,n);
spec_o2 = fft(o2,n);
spec_o3 = fft(o3,n);
```

```
fs=16000;
t=2;
L=t*fs
```

```
L = 32000
```

```
f = fs*(0:(L/2))/(L);
```

Arreglamos el vector de FFT para graficarlo

```
P2_a1= abs(spec_a1/L);
P1_a1 = P2_a1(1:L/2+1);
P1_a1(2:end-1) = 2*P1_a1(2:end-1);

P1_a1_dB = mag2db(P1_a1);

P2_a2= abs(spec_a2/L);
P1_a2 = P2_a2(1:L/2+1);
P1_a2(2:end-1) = 2*P1_a2(2:end-1);

P1_a2_dB = mag2db(P1_a2);

P2_a3= abs(spec_a3/L);
P1_a3 = P2_a3(1:L/2+1);
P1_a3(2:end-1) = 2*P1_a3(2:end-1);

P1_a3_dB = mag2db(P1_a3);

P2_i1= abs(spec_i1/L);
P1_i1 = P2_i1(1:L/2+1);
P1_i1(2:end-1) = 2*P1_i1(2:end-1);

P1_i1_dB = mag2db(P1_i1);

P2_i2= abs(spec_i2/L);
P1_i2 = P2_i2(1:L/2+1);
P1_i2(2:end-1) = 2*P1_i2(2:end-1);

P1_i2_dB = mag2db(P1_i2);
```

```

P2_i3= abs(spec_i3/L);
P1_i3 = P2_i3(1:L/2+1);
P1_i3(2:end-1) = 2*P1_i3(2:end-1);

P1_i3_dB = mag2db(P1_i3);

P2_o1= abs(spec_o1/L);
P1_o1 = P2_o1(1:L/2+1);
P1_o1(2:end-1) = 2*P1_o1(2:end-1);

P1_o1_dB = mag2db(P1_o1);

P2_o2= abs(spec_o2/L);
P1_o2 = P2_o2(1:L/2+1);
P1_o2(2:end-1) = 2*P1_o2(2:end-1);

P1_o2_dB = mag2db(P1_o2);

P2_o3= abs(spec_o3/L);
P1_o3 = P2_o3(1:L/2+1);
P1_o3(2:end-1) = 2*P1_o3(2:end-1);

P1_o3_dB = mag2db(P1_o3);

```

Graficamos la FFT

```

figure(2)

subplot(3,3,1)

plot(f,P1_a1)
xlabel('Frecuencia [Hz]')
ylabel('Amplitud ')
title('abajo')

subplot(3,3,2)
plot(f,P1_a2)
xlabel('Frecuencia [Hz]')
ylabel('Amplitud ')
title('arbol')

subplot(3,3,3)
plot(f,P1_a3)
xlabel('Frecuencia [Hz]')
ylabel('Amplitud ')
title('arriba')

```

```
subplot(3,3,4)
plot(f,P1_i1)
xlabel('Frecuencia [Hz]')
ylabel('Amplitud ')
title('iglesia')

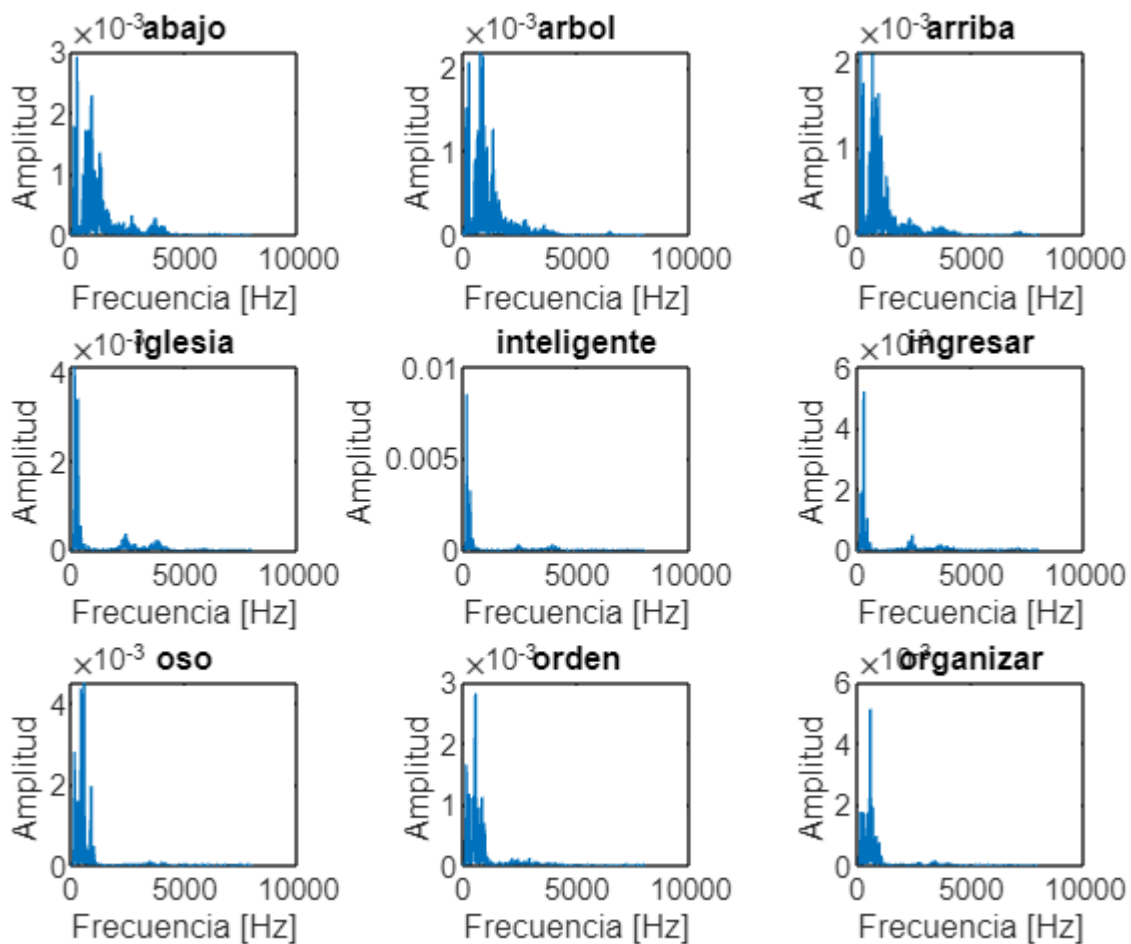
subplot(3,3,5)
plot(f,P1_i2)
xlabel('Frecuencia [Hz]')
ylabel('Amplitud ')
title('inteligente')

subplot(3,3,6)
plot(f,P1_i3)
xlabel('Frecuencia [Hz]')
ylabel('Amplitud ')
title('ingresar')

subplot(3,3,7)
plot(f,P1_o1)
xlabel('Frecuencia [Hz]')
ylabel('Amplitud ')
title('oso')

subplot(3,3,8)
plot(f,P1_o2)
xlabel('Frecuencia [Hz]')
ylabel('Amplitud ')
title('orden')

subplot(3,3,9)
plot(f,P1_o3)
xlabel('Frecuencia [Hz]')
ylabel('Amplitud ')
title('organizar')
```



Porcentaje de Energía entre 200-1200 Hz

```
filter=designfilt('bandpassiir', 'FilterOrder', 8, 'HalfPowerFrequency1', 200, 'HalfPowerFrequency2', 1200, 'DesignMethod', 'butter');
```

```
filter =  
    digitalFilter with properties:  
        Coefficients: [4x6 double]  
  
    Specifications:  
        FrequencyResponse: 'bandpass'  
        ImpulseResponse: 'iir'  
        SampleRate: 16000  
        HalfPowerFrequency2: 1200  
        HalfPowerFrequency1: 200  
        FilterOrder: 8  
        DesignMethod: 'butter'
```

```
Use fvtool to visualize filter  
Use designfilt to edit filter  
Use filter to filter data
```

```
Filt_a1 = filtfilt(filter, a1);  
Filt_a2 = filtfilt(filter, a2);  
Filt_a3 = filtfilt(filter, a3);
```

```
Filt_i1 = filtfilt(filter, i1);
Filt_i2 = filtfilt(filter, i2);
Filt_i3 = filtfilt(filter, i3);
Filt_o1 = filtfilt(filter, o1);
Filt_o2 = filtfilt(filter, o2);
Filt_o3 = filtfilt(filter, o3);
```

```
E_FILT_A1= sum(Filt_a1.^2)
```

```
E_FILT_A1 = 11.2818
```

```
E_FILT_A2= sum(Filt_a2.^2)
```

```
E_FILT_A2 = 8.7772
```

```
E_FILT_A3= sum(Filt_a3.^2)
```

```
E_FILT_A3 = 7.4871
```

```
E_FILT_I1= sum(Filt_i1.^2)
```

```
E_FILT_I1 = 5.7662
```

```
E_FILT_I2= sum(Filt_i2.^2)
```

```
E_FILT_I2 = 5.4356
```

```
E_FILT_I3= sum(Filt_i3.^2)
```

```
E_FILT_I3 = 7.0029
```

```
E_FILT_O1= sum(Filt_o1.^2)
```

```
E_FILT_O1 = 24.3143
```

```
E_FILT_O2= sum(Filt_o2.^2)
```

```
E_FILT_O2 = 8.2418
```

```
E_FILT_O3= sum(Filt_o3.^2)
```

```
E_FILT_O3 = 20.3913
```

4. Grafique las FFT para las tres vocales en 1 misma figura, la magnitud en dB.

```
figure(3)
f = fs*(0:(L/2))/(L);
plot(f,P1_a3, 'r')
hold on
plot(f,P1_i1_dB, 'g')
```



```

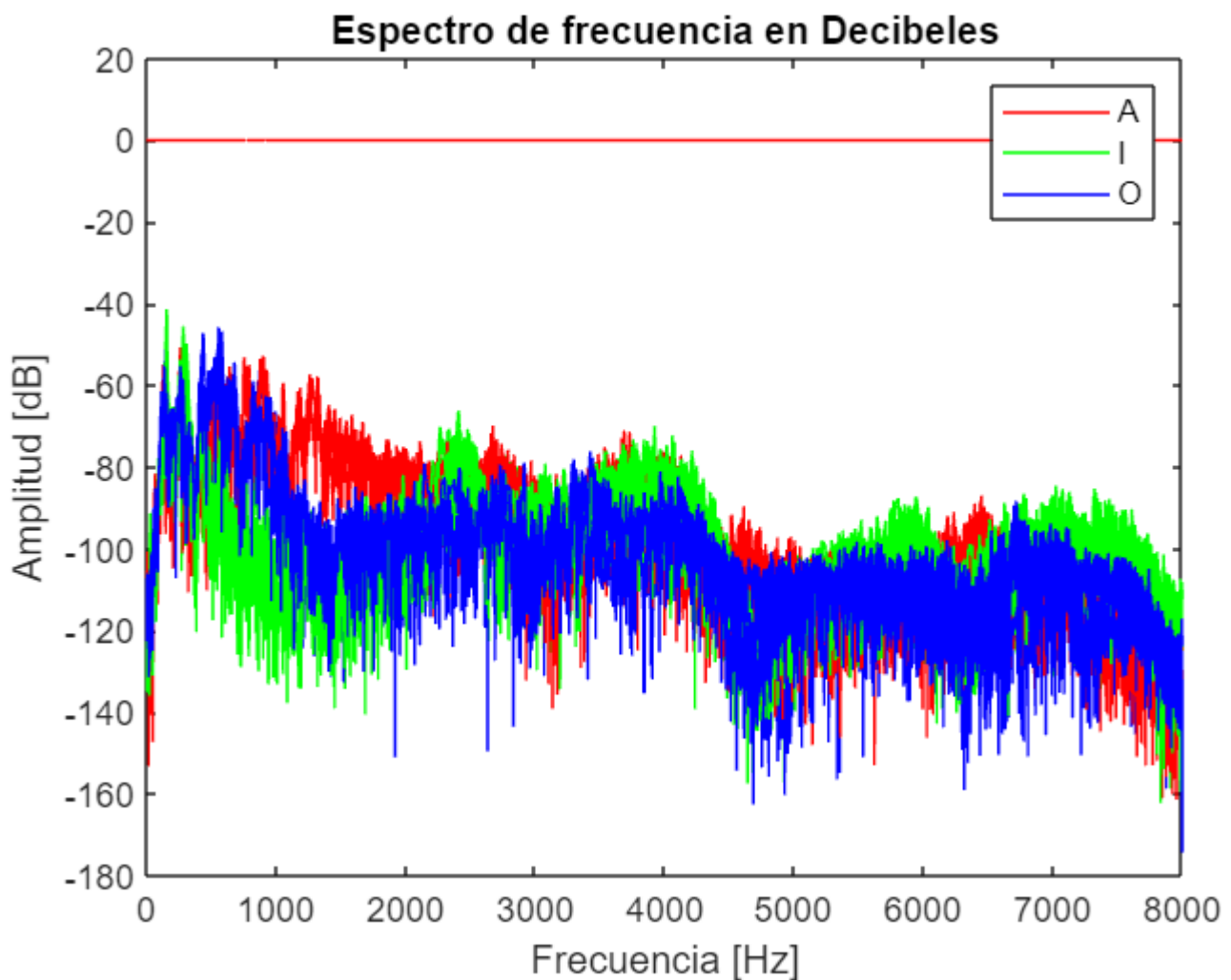
plot(f,P1_o1_dB, 'b')
plot(f,P1_a1_dB, 'r')

plot(f,P1_a2_dB, 'r')

plot(f,P1_i2_dB, 'g')
plot(f,P1_i3_dB, 'g')

plot(f,P1_o2_dB, 'b')
plot(f,P1_o3_dB, 'b')
xlabel('Frecuencia [Hz]')
ylabel('Amplitud [dB]')
title('Espectro de frecuencia en Decibeles')
legend('A','I','O')
hold off

```



Podemos ver la gráfica con amplitud en decibeles. El color azul representa las 3 vocales 'a', el color verde representa las 3 vocales 'i' y el color azul representa las 3 vocales 'o'.

Se ve que todas las 3 señales tienen una mayor amplitud para frecuencias bajas y a mayor frecuencia menor es el volumen de la señal.

Se puede ver que en algunas zonas sobresale un pico característico de las señales.

5. Proponga una medida a partir del porcentaje de energía de la señal en bandas de frecuencia específicas que permita diferenciar una vocal de otra

Si nos fijamos en las frecuencias antes de 3000 Hz podemos observar picos distintivos de las 3 señales. Si miramos los valores de amplitud en esta región, hay algunos picos que se diferencian para cada señal; vemos 1 pico positivo azul y un pico negativo verde en frecuencias menores a 1000, un pico rojo entre 1000-2000 y un pico verde en 2200 y 4000.

Esto corresponde con las frecuencias de los formantes de las vocales.

Haciendo uso del espectro de Fourier sin pasarlo a Decibelios nos sirve para identificar con facilidad las bandas de frecuencias representativas de cada señal.

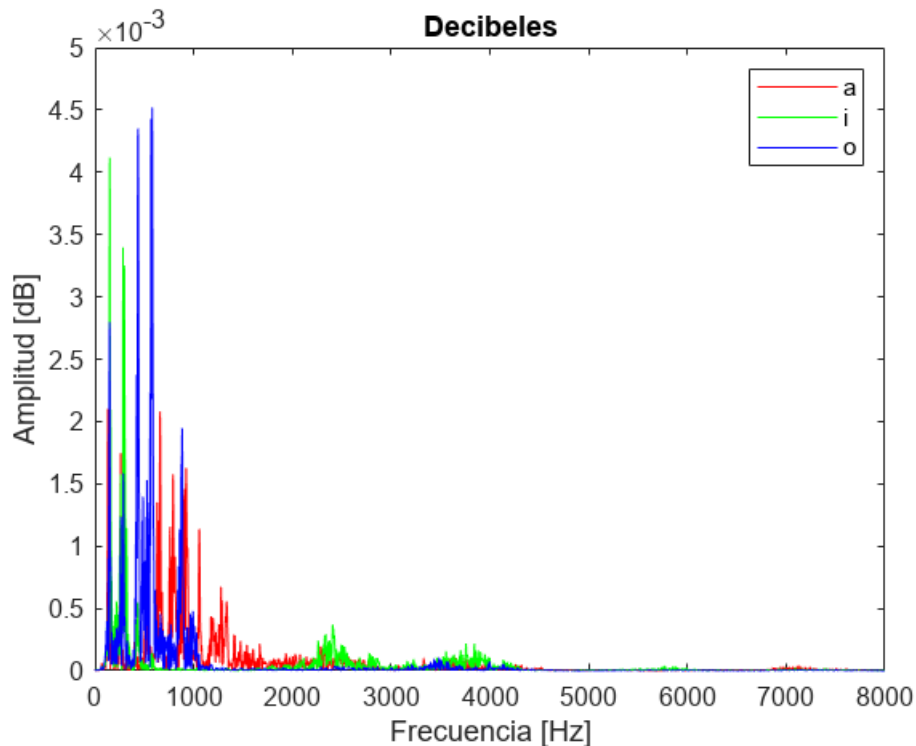
Observando la gráfica podemos ver que las frecuencias a las cuales encontraremos las vocales son:

A: 700, 1200

I: 300, 2400

O: 500

```
figure(4)
f = fs*(0:(L/2))/(L);
plot(f,abs(P1_a3), 'r')
hold on
%plot(f,P1_a1, 'r')
%plot(f,P1_a2, 'r')
plot(f,abs(P1_i1), 'g')
%plot(f,P1_i2, 'g')
%plot(f,P1_i3, 'g')
plot(f,abs(P1_o1), 'b')
%plot(f,P1_o2, 'b')
%plot(f,P1_o3, 'b')
xlabel('Frecuencia [Hz]')
ylabel('Amplitud [dB]')
title('Decibeles')
legend('a','i','o')
hold off
```



Propuesta:

Dada una señal de audio desconocida de una de las tres vocales : a, i , o , se aplicará la Transformada Rápida de Fourier para obtener la señal en terminos de la frecuencia.

Luego, se crearan 3 filtros especificos para resaltar las bandas de frecuencia de cada una de las vocales, según se discutió en el 4 punto. Posteriormente se calculará la energía de la señal en estas seales filtradas y se medirá la Energía.

La señal corresponderá a la señal filtrada con mayor energía.

Las bandas que se esperan conservar en los 3 filtros que se diseñaran se muestran a continuacion [Hz]

A: 1200

I: 300

O: 500

6. Hacer un clasificador de vocales y realizar minimo 10 grabacione para probarlo.

```
g2=probar10Audios()
```

Es una A
Es una A
Es una A
Es una A

```

Es una A
Es una A
Es una A
Es una A
Es una A
Es una A
Es una I
Es una I
Es una I
Es una I
Es una I
Es una I
Es una I
Es una I
Es una I
Es una A
Es una 0
Es una 0
Es una 0
Es una 0
Es una 0
Es una 0
Es una 0
Es una 0
Es una 0
Es una 0
Es una 0
Es una 0
Es una 0
Es una 0
Es una 0
g2 = 1×30
    1    1    1    1    1    1    1    1    1    1    2    2    2 ...

```

```

g1 = [1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3]; % Known groups
C = confusionmat(g1,g2)

```

```

C = 3×3
    10     0     0
     1     9     0
     0     0    10

```

```

confusionchart(C)

```

1	10		
2	1	9	
3			10
	1	2	3
		Predicted Class	

```

aciertos=29;
total=30;
Acierto=(aciertos/total)*100

```

Acierto = 96.6667

```

function [g2] = probar10Audios()
current_directory=pwd;

dir_A=strcat(current_directory, '\A');
files_A= dir(dir_A);
dir_I=strcat(current_directory, '\I');
files_I= dir(dir_I);
dir_O=strcat(current_directory, '\O');
files_O= dir(dir_O);

```

```

g2 = [];

for i=3:12
    name=files_A(i).name;
    name=strcat('A\ ', name);
    vocal=audioread(name);
    g2(end+1)=clasificador(vocal);

end

for i=3:12
    name=files_I(i).name;
    name=strcat('I\ ', name);
    vocal=audioread(name);
    g2(end+1)=clasificador(vocal);

end

for i=3:12
    name=files_O(i).name;
    name=strcat('O\ ', name);
    vocal=audioread(name);
    g2(end+1)=clasificador(vocal);

end

end

```

```

function [] = grabar()
disp('Vamos a grabar 10 vocales A de 2 segundos');
pause(1)
disp('Preparate... vocal A , duraci[on 2 segundos...');
pause(1)

for i=1:10
    txt=strcat('Audio A #', num2str(i));
    disp(txt)
    [x,t]=grabarA();
    audiowrite(t, x, fs);
    pause(1)

end

disp('Vamos a grabar 10 vocales I de 2 segundos');
pause(1)
disp('Preparate... vocal I , duraci[on 2 segundos...');

```

```

pause(1)

for i=1:10
    txt=strcat('Audio I #', num2str(i));
    disp(txt)
    [x,t]=grabarI(i);
    audiowrite(t, x, fs);
    pause(1)

end

disp('Vamos a grabar 10 vocales 0 de 2 segundos');
pause(1)
disp('Preparate... vocal 0 , duraci[on 2 segundos...');
pause(1)

for i=1:10
    txt=strcat('Audio O #', num2str(i));
    disp(txt)
    [x,t]=grabarO();
    audiowrite(t, x, fs);
    pause(1)

end
end

```

```

function vocal = clasificador(audio)

n=32000;
L=n;

filterA=designfilt('bandpassiir', 'FilterOrder', 8, 'HalfPowerFrequency1', 1200, 'HalfPowerFrequency2', 2000, 'FilterOrder', 8, 'HalfPowerFrequency1', 200, 'HalfPowerFrequency2', 500);
filterI=designfilt('bandpassiir', 'FilterOrder', 8, 'HalfPowerFrequency1', 200, 'HalfPowerFrequency2', 500);
filterO=designfilt('bandpassiir', 'FilterOrder', 8, 'HalfPowerFrequency1', 500, 'HalfPowerFrequency2', 2000);

Filt_a = filtfilt(filterA, audio);
Filt_i = filtfilt(filterI, audio);
Filt_o = filtfilt(filterO, audio);

spec_a = fft(Filt_a,n);
spec_i = fft(Filt_i,n);
spec_o = fft(Filt_o,n);

```

```

P2_a= abs(spec_a/L);
P2_i= abs(spec_i/L);
P2_o= abs(spec_o/L);
P1_a = P2_a(1:L/2+1);
P1_i = P2_i(1:L/2+1);
P1_o = P2_o(1:L/2+1);
P1_a(2:end-1) = 2*P1_a(2:end-1);
P1_i(2:end-1) = 2*P1_i(2:end-1);
P1_o(2:end-1) = 2*P1_o(2:end-1);

indices_a2= P1_i > 2e-3;
P1_i=P1_i.* indices_a2;

indices_a3= P1_o > 2e-3;
P1_o=P1_o.* indices_a3;

E_A= sum(abs(P1_a).^2);
E_I= sum(abs(P1_i).^2);
E_O= sum(abs(P1_o).^2);

if (E_A>= E_I) && (E_A >= E_O)
    disp('Es una A')
    vocal=1;
elseif (E_I>= E_A) && (E_I >= E_O)
    disp('Es una I')
    vocal=2;
else
    disp('Es una O')
    vocal=3;
end

end

```

```

function [x,texto] = grabarA()
%% Grabacion e identificador
fs= 16000;
t = 2;
nBits = 16;
nChannels = 1;
ID = -1;
recObj = audiorecorder(fs,nBits,nChannels,ID);

%% ARCHIVOS
current_directory=pwd;
dir_A=strcat(current_directory, '\A');
files_A= dir(dir_A);

```



```

last_file_A= files_A(end).name;
[j,name_A,u]=fileparts(last_file_A);
id_A=split(name_A, 'A');
id_A=str2double(id_A{end});

texto=strcat('\A',num2str(id_A+1) );
texto=strcat(dir_A,texto );
texto=strcat(texto, '.wav' );

disp('grabando audio. ');
recordblocking(recObj, t);
disp('Fin del audio. ');

x=recObj.getaudiodata();

disp('Señal capturada');

end

```

```

function [x,texto] = grabarI()
%% Grabacion e identificador
fs= 16000;
t = 2;
nBits = 16;
nChannels = 1;
ID = -1;
recObj = audiorecorder(fs,nBits,nChannels,ID);

%% ARCHIVOS
current_directory=pwd;
dir_I=strcat(current_directory,'\I');
files_I= dir(dir_I);

last_file_I= files_I(end).name;
[j,name_I,u]=fileparts(last_file_I);
id_I=split(name_I, 'I');
id_I=str2double(id_I{end});

texto=strcat('\I',num2str(id_I+1) );
texto=strcat(dir_I,texto );
texto=strcat(texto, '.wav' );

disp('grabando audio. ');
recordblocking(recObj, t);
disp('Fin del audio. ');

x=recObj.getaudiodata();

```

```
disp('Señal capturada');
```

```
end
```

```
function [x,texto] = grabar0()  
%% Grabacion e identificador  
fs= 16000;  
t = 2;  
nBits = 16;  
nChannels = 1;  
ID = -1;  
recObj = audiorecorder(fs,nBits,nChannels,ID);
```

```
%% ARCHIVOS  
current_directory=pwd;  
dir_0=strcat(current_directory, '\0');  
files_0= dir(dir_0);
```

```
last_file_0= files_0(end).name;  
[j,name_0,u]=fileparts(last_file_0);  
id_0=split(name_0, '0');  
id_0=str2double(id_0{end});
```

```
texto=strcat('\0',num2str(id_0+1) );  
texto=strcat(dir_0,texto );  
texto=strcat(texto, '.wav' );
```

```
disp('grabando audio.');
```

```
recordblocking(recObj, t);
```

```
disp('Fin del audio.');
```

```
x=recObj.getaudiodata();
```

```
disp('Señal capturada');
```

```
end
```