

<b>Nombre:</b> Urzua Arellano Diego Leonardo		<b>Matrícula:</b> 10100644
<b>Nombre del curso:</b> Análisis Digital de señales		<b>Nombre del profesor:</b> Rodrigo Calderón Rico
<b>Módulo:</b> Proyecto		<b>Actividad:</b> Detección de voz
<b>Fecha:</b> 09/12/2013		

**Objetivo:**

Realizar un reconocedor de voz el cual pueda reconocer diferentes palabras dadas por cualquier usuario y también que pueda reconocer a diferentes usuarios

**Procedimiento:**

El programa denominado “voice leo” se a programado en Matlab por su gran capacidad

Para entender el funcionamiento del programa se expone la figura P1

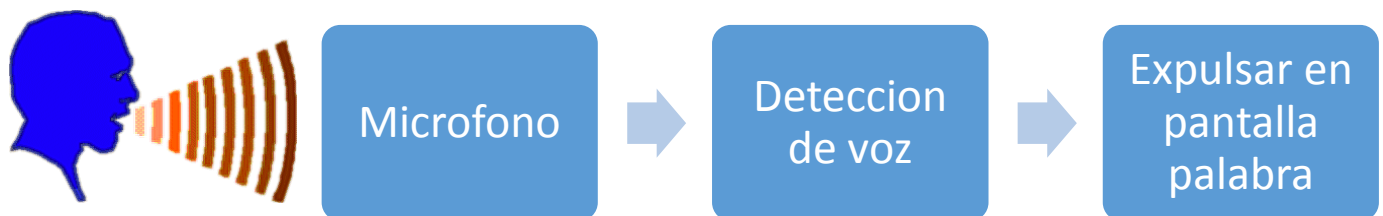


Figura P1 “Funcionamiento de voice leo”

Antes de continuar se va hacer hincapié en las palabras que podrá reconocer  
“voice leo”

- Hola
- Enciende
- Apaga
- Leonardo

Para comenzar se tendrá que grabar la voz y almacenar en una variable para poderla manipular en este proceso cabe mencionar que se grabara con una frecuencia de muestreo ya que se graba una señal discreta en tiempo y amplitud de esa manera se ahorra el convertidor analógico digital (ADC) ya que este proceso esta implícito en el proceso

Una vez que se a grabado la voz en el dominio del tiempo se va aplicar la transformada de Fourier para obtener la voz en el dominio de la frecuencia ya que una señal (variable) en el dominio del tiempo es muy difícil de controlar ya que el tiempo es incontrolable e infinito y en el dominio de la frecuencia las señales (variables) son manipulables para efectúales diversos procesos además de que en este dominio podemos apreciar todas las componentes de frecuencias que conforman a la señal

Ahora que se tiene la señal en el dominio de la frecuencia se desea obtener las potencias que tiene en el espectro de potencias para observar las frecuencias mas características que tiene la señal esto se realiza sacando el conjugado a la transformada de Fourier de la señal

Una vez que se tienen las características de la señal se tiene que observar o comparar con un banco de señales (palabras) que se les a aplicado el mismo proceso previamente y guardado en un documento (solo las potencias de la señal)

Para poder comparar el parecido entre ambas señales existe un método denominado “correlación cruzada” el cual tiene como objetivo medir el parecido que existe entre dos señales y así extraer información que dependerá de la aplicación concreta. En otras palabras la correlación expulsara como resultado que tanto se parecen 2 señales en este caso que tanto parecido en promedio hay entre las frecuencias características de las señales

Una vez que se ha comparado el parecido de la señal con el banco de señales por medio de la correlación cruzada se procede a determinar la palabra que ha dicho el usuario para expulsarla. Para determinar la palabra que ha dicho el usuario basta con comparar cual es la que tiene la correlación cruzada promedio mayor de cada promedio de la correlación del banco de señales y la señal

Se hicieron 2 grabaciones previas de las palabras la primera en un ambiente silencioso y la otra en un ambiente general

### **Código fuente**

```
%Programer Urzua Arellano Diego Leonardo LEOTRON
%October-december 2013
%Voice Detection

% Record your voice for 3 seconds.
clear all;
clc;
disp('Voice detection/n')
recObj = audiorecorder;
disp('Start speaking.')
recordblocking(recObj, 3);
disp('End of Recording.');
```

```
% Store data in double-precision array.

myRecording = getaudiodata(recObj);           %1 audio

%fast transform
resolution=15e3;

transform=fft(myRecording,resolution);

%%%% pot of frencuncy
pot=transform.*conj(transform); % pot of signal 1
f=(1:resolution); % espectrum of frequency

%%%load words

load('Hola.mat');
load('Holaa.mat');
load('Leonardo.mat');
load('Leonardoa.mat');
load('Enciende.mat');
load('Enciendea.mat');
load('Apagar.mat');
load('Apagara.mat');

%%% correlation(signal)

Holacorrelation1 = max(corr(pot,Hola));
Holacorrelation2= max(corr(pot,Hola1));
Holacorrelation3= max(corr(pot,Hola2));
Holacorrelation4= max(corr(pot,Hola3));
Holacorrelation5= max(corr(pot,Hola4));
Holacorrelation6= max(corr(pot,Hola5));
Holacorrelation7= max(corr(pot,Hola6));
Holacorrelation8= max(corr(pot,Hola7));

%%%% Hola leonardo ambiente

Holacorrelational1 = max(corr(pot,Holaa));
Holacorrelational2= max(corr(pot,Holaa1));
Holacorrelational3= max(corr(pot,Holaa2));
Holacorrelational4= max(corr(pot,Holaa3));
Holacorrelational5= max(corr(pot,Holaa4));
Holacorrelational6= max(corr(pot,Holaa5));
Holacorrelational7= max(corr(pot,Holaa6));
Holacorrelational8= max(corr(pot,Holaa7));
```

```
Holameann =
(Holacorrelation1+Holacorrelation2+Holacorrelation3+Holacorrelation4+HOLA
correlation5+Holacorrelation6+Holacorrelation7+Holacorrelation8)/8;
Holameanna =
(Holacorrelational1+Holacorrelational2+Holacorrelational3+Holacorrelational4+
Holacorrelational5+Holacorrelational6+Holacorrelational7+Holacorrelational8)/
8;
Holatotal = (Holameann+Holameanna)/2;
```

```
Leonardocorrelation1 = max(corr(pot,Leonardo));
Leonardocorrelation2= max(corr(pot,Leonardo1));
Leonardocorrelation3= max(corr(pot,Leonardo2));
Leonardocorrelation4= max(corr(pot,Leonardo3));
Leonardocorrelation5= max(corr(pot,Leonardo4));
Leonardocorrelation6= max(corr(pot,Leonardo5));
Leonardocorrelation7= max(corr(pot,Leonardo6));
Leonardocorrelation8= max(corr(pot,Leonardo7));
```

%%% Leonardo Ambiente Leonardo

```
Leonardocorrelational1 = max(corr(pot,Leonardoa));
Leonardocorrelational2= max(corr(pot,Leonardoa1));
Leonardocorrelational3= max(corr(pot,Leonardoa2));
Leonardocorrelational4= max(corr(pot,Leonardoa3));
Leonardocorrelational5= max(corr(pot,Leonardoa4));
Leonardocorrelational6= max(corr(pot,Leonardoa5));
Leonardocorrelational7= max(corr(pot,Leonardoa6));
Leonardocorrelational8= max(corr(pot,Leonardoa7));
```

```
Leonardomeann =
(Leonardocorrelation1+Leonardocorrelation2+Leonardocorrelation3+Leonardoc
orrelation4+Leonardocorrelation5+Leonardocorrelation6+Leonardocorrelation
7+Leonardocorrelation8)/8;
Leonardomeanna =
(Leonardocorrelational1+Leonardocorrelational2+Leonardocorrelational3+Leonar
docorrelational4+Leonardocorrelational5+Leonardocorrelational6+Leonardocorre
lational7+Leonardocorrelational8)/8;
Leonardototal = ( Leonardomeann + Leonardomeanna ) / 2;
```

```
Enciendecorrelation1 = max(corr(pot,Enciende));
Enciendecorrelation2= max(corr(pot,Enciende1));
Enciendecorrelation3= max(corr(pot,Enciende2));
Enciendecorrelation4= max(corr(pot,Enciende3));
Enciendecorrelation5= max(corr(pot,Enciende4));
Enciendecorrelation6= max(corr(pot,Enciende5));
Enciendecorrelation7= max(corr(pot,Enciende6));
Enciendecorrelation8= max(corr(pot,Enciende7));
```

%%%%%%%% Leonardo ambiente

```
Enciendecorrelational1 = max(corr(pot,Enciendea));
Enciendecorrelational2= max(corr(pot,Enciendea1));
Enciendecorrelational3= max(corr(pot,Enciendea2));
Enciendecorrelational4= max(corr(pot,Enciendea3));
```

```
Enciendecorrelational5= max(corr(pot,Enciendea4));
Enciendecorrelational6= max(corr(pot,Enciendea5));
Enciendecorrelational7= max(corr(pot,Enciendea6));
Enciendecorrelational8= max(corr(pot,Enciendea7));
```

```
Enciendemeann =
(Enciendecorrelation1+Enciendecorrelation2+Enciendecorrelation3+Enciendecorrelation4+Enciendecorrelation5+Enciendecorrelation6+Enciendecorrelation7+Enciendecorrelation8)/8;
Enciendemeanna =
(Enciendecorrelational1+Enciendecorrelational2+Enciendecorrelational3+Enciendecorrelational4+Enciendecorrelational5+Enciendecorrelational6+Enciendecorrelational7+Enciendecorrelational8)/8;
Enciendetotal = (Enciendemeann+Enciendemeanna)/2;
```

```
Apagarcorrelation1 = max(corr(pot,Apagar));
Apagarcorrelation2= max(corr(pot,Apagar1));
Apagarcorrelation3= max(corr(pot,Apagar2));
Apagarcorrelation4= max(corr(pot,Apagar3));
Apagarcorrelation5= max(corr(pot,Apagar4));
Apagarcorrelation6= max(corr(pot,Apagar5));
Apagarcorrelation7= max(corr(pot,Apagar6));
Apagarcorrelation8= max(corr(pot,Apagar7));
```

%%Apagar ambiente Leonardo

```
Apagarcorrelational1 = max(corr(pot,Apagara));
Apagarcorrelational2= max(corr(pot,Apagara1));
Apagarcorrelational3= max(corr(pot,Apagara2));
Apagarcorrelational4= max(corr(pot,Apagara3));
Apagarcorrelational5= max(corr(pot,Apagara4));
Apagarcorrelational6= max(corr(pot,Apagara5));
Apagarcorrelational7= max(corr(pot,Apagara6));
Apagarcorrelational8= max(corr(pot,Apagara7));
```

```
Apagarmeann =
(Apagarcorrelation1+Apagarcorrelation2+Apagarcorrelation3+Apagarcorrelation4+Apagarcorrelation5+Apagarcorrelation6+Apagarcorrelation7+Apagarcorrelation8)/8;
Apagarmeanna =
(Apagarcorrelational1+Apagarcorrelational2+Apagarcorrelational3+Apagarcorrelational4+Apagarcorrelational5+Apagarcorrelational6+Apagarcorrelational7+Apagarcorrelational8)/8;
Apagartotal = ( Apagarmeann + Apagarmeanna )/2;
```

```
wordHola = Holatotal;
wordEnciende = Enciendetotal;      %%%
wordApagar= Apagartotal;
wordLeonardo = Leonardototal;
```

`%compare`

```
if wordHola >= wordEnciende && wordHola >= wordApagar && wordHola >=
wordLeonardo
    disp('Hola')
elseif wordEnciende >= wordHola && wordEnciende >= wordApagar &&
wordEnciende >= wordLeonardo
    disp('Enciende')
elseif wordApagar >= wordEnciende && wordApagar >= wordHola && wordApagar
>= wordLeonardo
    disp('Apagar')
else
    disp('Leonardo')

end
```

## ***Código Fuente extras***

### Hola

```
% Record your voice for 3 seconds.
clear all;
clc;
disp('Voice detection/n')
Holanalog = audiorecorder;
disp('Start speaking.')
recordblocking(Holanalog, 3);
disp('End of Recording.');
```

`% Store data in double-precision array.`

```
myRecording = getaudiodata(Holanalog);           %1 audio
```

`%fast tramsform`

```
resolution=15e3;

transform=fft(myRecording,resolution);
```

`%%%% pot of frencuncy`

```
Hola=transform.*conj(transform);           % pot of signal 1
```

```
disp('Voice detection/n')
Holanalog = audiorecorder;
disp('Start speaking.')
recordblocking(Holanalog, 3);
```

```
disp('End of Recording.');
```

```
% Store data in double-precision array.
```

```
myRecording = getaudiodata(Holalog);           %1 audio
```

```
%fast transform  
resolution=15e3;
```

```
transform=fft(myRecording,resolution);
```

```
%%%% pot of frencuncy  
Holal=transform.*conj(transform);    % pot of signal 1
```

```
disp('Voice detection/n')  
Holalog = audiorecorder;  
disp('Start speaking.')
```

```
recordblocking(Holalog, 3);  
disp('End of Recording.');
```

```
% Store data in double-precision array.
```

```
myRecording = getaudiodata(Holalog);           %1 audio
```

```
%fast transform  
resolution=15e3;
```

```
transform=fft(myRecording,resolution);
```

```
%%%% pot of frencuncy  
Hola2=transform.*conj(transform);    % pot of signal 1
```

```
disp('Voice detection/n')  
Holalog = audiorecorder;  
disp('Start speaking.')
```

```
recordblocking(Holalog, 3);  
disp('End of Recording.');
```

```
% Store data in double-precision array.
```

```
myRecording = getaudiodata(Holalog);           %1 audio
```



```
%fast transform
resolution=15e3;

transform=fft(myRecording,resolution);

%%%% pot of frencuncy
Hola3=transform.*conj(transform);    % pot of signal 1

disp('Voice detection/n')
Holanalog = audiorecorder;
disp('Start speaking.')
recordblocking(Holanalog, 3);
disp('End of Recording.');
```

% Store data in double-precision array.

```
myRecording = getaudiodata(Holanalog);    %1 audio
```

```
%fast transform
resolution=15e3;

transform=fft(myRecording,resolution);

%%%% pot of frencuncy
Hola4=transform.*conj(transform);    % pot of signal 1

disp('Voice detection/n')
Holanalog = audiorecorder;
disp('Start speaking.')
recordblocking(Holanalog, 3);
disp('End of Recording.');
```

% Store data in double-precision array.

```
myRecording = getaudiodata(Holanalog);    %1 audio
```

```
%fast transform
resolution=15e3;

transform=fft(myRecording,resolution);
```

```
%%%% pot of frencuncy
Hola5=transform.*conj(transform);    % pot of signal 1

disp('Voice detection/n')
Holanalog = audiorecorder;
disp('Start speaking.')
recordblocking(Holanalog, 3);
disp('End of Recording.');
```

% Store data in double-precision array.

```
myRecording = getaudiodata(Holanalog);    %1 audio
```

%fast transform  
resolution=15e3;

```
transform=fft(myRecording,resolution);
```

%%%% pot of frencuncy  
Hola6=transform.\*conj(transform); % pot of signal 1

```
disp('Voice detection/n')
Holanalog = audiorecorder;
disp('Start speaking.')
recordblocking(Holanalog, 3);
disp('End of Recording.');
```

% Store data in double-precision array.

```
myRecording = getaudiodata(Holanalog);    %1 audio
```

%fast transform  
resolution=15e3;

```
transform=fft(myRecording,resolution);
```

%%%% pot of frencuncy  
Hola7=transform.\*conj(transform); % pot of signal 1

```
save words.mat Hola Hola1 Hola2 Hola3 Hola4 Hola5 Hola6 Hola7;
```

## Enciende

```
% Record your voice for 3 seconds.
clear all;
clc;
disp('Voice detection/n')
Holanalog = audiorecorder;
disp('Start speaking.')
recordblocking(Holanalog, 3);
disp('End of Recording.');
```

```
% Store data in double-precision array.

myRecording = getaudiodata(Holanalog);           %1 audio
```

```
%fast transform
resolution=15e3;

transform=fft(myRecording,resolution);
```

```
%%%% pot of frencuncy
Enciende=transform.*conj(transform);           % pot of signal
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
disp('Voice detection/n')
Holanalog = audiorecorder;
disp('Start speaking.')
recordblocking(Holanalog, 3);
disp('End of Recording.');
```

```
% Store data in double-precision array.

myRecording = getaudiodata(Holanalog);           %1 audio
```

```
%fast transform
resolution=15e3;

transform=fft(myRecording,resolution);
```

```

%%%% pot of frencuncy
Enciende7=transform.*conj(transform);    % pot of signal

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
disp('Voice detection/n')
Holanalog = audiorecorder;
disp('Start speaking.')
recordblocking(Holanalog, 3);
disp('End of Recording.');
```

% Store data in double-precision array.

```

myRecording = getaudiodata(Holanalog);          %1 audio

%fast transform
resolution=15e3;

transform=fft(myRecording,resolution);

%%%% pot of frencuncy
Enciende1=transform.*conj(transform);    % pot of signal

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%
disp('Voice detection/n')
Holanalog = audiorecorder;
disp('Start speaking.')
recordblocking(Holanalog, 3);
disp('End of Recording.');
```

% Store data in double-precision array.

```

myRecording = getaudiodata(Holanalog);          %1 audio

%fast transform
resolution=15e3;

transform=fft(myRecording,resolution);

%%%% pot of frencuncy
Enciende2=transform.*conj(transform);    % pot of signal
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%
```

```
disp('Voice detection/n')
Holanalog = audiorecorder;
disp('Start speaking.')
recordblocking(Holanalog, 3);
disp('End of Recording.');
```

```
% Store data in double-precision array.
```

```
myRecording = getaudiodata(Holanalog);           %1 audio
```

```
%fast transform
resolution=15e3;
```

```
transform=fft(myRecording,resolution);
```

```
%%%% pot of frencuncy
Enciende3=transform.*conj(transform);           % pot of signal
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%
```

```
disp('Voice detection/n')
Holanalog = audiorecorder;
disp('Start speaking.')
recordblocking(Holanalog, 3);
disp('End of Recording.');
```

```
% Store data in double-precision array.
```

```
myRecording = getaudiodata(Holanalog);           %1 audio
```

```
%fast transform
resolution=15e3;
```

```
transform=fft(myRecording,resolution);
```

```
%%%% pot of frencuncy
Enciende4=transform.*conj(transform);           % pot of signal
```

[illegible]

```
save Enciende.mat Enciende Enciende1 Enciende2 Enciende3 Enciende4  
Enciende5 Enciende6 Enciende7;
```

## Leonardo

```
% Record your voice for 3 seconds.  
clear all;  
clc;  
disp('Voice detection/n')  
Holanalog = audiorecorder;  
disp('Start speaking.')  
recordblocking(Holanalog, 3);  
disp('End of Recording.');
```

```
% Store data in double-precision array.  
  
myRecording = getaudiodata(Holanalog);           %1 audio  
  
%fast transform  
resolution=15e3;  
  
transform=fft(myRecording,resolution);  
  
%%%% pot of frencuncy  
Leonardo=transform.*conj(transform);           % pot of signal  
  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
disp('Voice detection/n')  
Holanalog = audiorecorder;  
disp('Start speaking.')  
recordblocking(Holanalog, 3);  
disp('End of Recording.');
```

```
% Store data in double-precision array.  
  
myRecording = getaudiodata(Holanalog);           %1 audio  
  
%fast transform
```

```
resolution=15e3;

transform=fft(myRecording,resolution);

%%%% pot of frencuncy
Leonardo7=transform.*conj(transform);    % pot of signal

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
disp('Voice detection/n')
Holanalog = audiorecorder;
disp('Start speaking.')
recordblocking(Holanalog, 3);
disp('End of Recording.');
```

% Store data in double-precision array.

```
myRecording = getaudiodata(Holanalog);          %1 audio

%fast tramsform
resolution=15e3;

transform=fft(myRecording,resolution);

%%%% pot of frencuncy
Leonardo1=transform.*conj(transform);    % pot of signal

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%
disp('Voice detection/n')
Holanalog = audiorecorder;
disp('Start speaking.')
recordblocking(Holanalog, 3);
disp('End of Recording.');
```

% Store data in double-precision array.

```
myRecording = getaudiodata(Holanalog);          %1 audio

%fast tramsform
resolution=15e3;

transform=fft(myRecording,resolution);
```



```
##### pot of frencuncy
Leonardo2=transform.*conj(transform);    % pot of signal

#####

disp('Voice detection/n')
Holanalog = audiorecorder;
disp('Start speaking.')
recordblocking(Holanalog, 3);
disp('End of Recording.');
```

% Store data in double-precision array.

```
myRecording = getaudiodata(Holanalog);          %1 audio

%fast tramsform
resolution=15e3;

transform=fft(myRecording,resolution);

##### pot of frencuncy
Leonardo3=transform.*conj(transform);    % pot of signal

#####

disp('Voice detection/n')
Holanalog = audiorecorder;
disp('Start speaking.')
recordblocking(Holanalog, 3);
disp('End of Recording.');
```

% Store data in double-precision array.

```
myRecording = getaudiodata(Holanalog);          %1 audio

%fast tramsform
resolution=15e3;

transform=fft(myRecording,resolution);
```

```

%%%% pot of frencuncy
Leonardo4=transform.*conj(transform);    % pot of signal

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%

disp('Voice detection/n')
Holanalog = audiorecorder;
disp('Start speaking.')
recordblocking(Holanalog, 3);
disp('End of Recording.');
```

% Store data in double-precision array.

```

myRecording = getaudiodata(Holanalog);          %1 audio

%fast tramsform
resolution=15e3;

transform=fft(myRecording,resolution);

%%%% pot of frencuncy
Leonardo5=transform.*conj(transform);    % pot of signal

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%

disp('Voice detection/n')
Holanalog = audiorecorder;
disp('Start speaking.')
recordblocking(Holanalog, 3);
disp('End of Recording.');
```

% Store data in double-precision array.

```

myRecording = getaudiodata(Holanalog);          %1 audio
```

```
%fast transform
resolution=15e3;

transform=fft(myRecording,resolution);

%%%% pot of frencuncy
Leonardo6=transform.*conj(transform);    % pot of signal

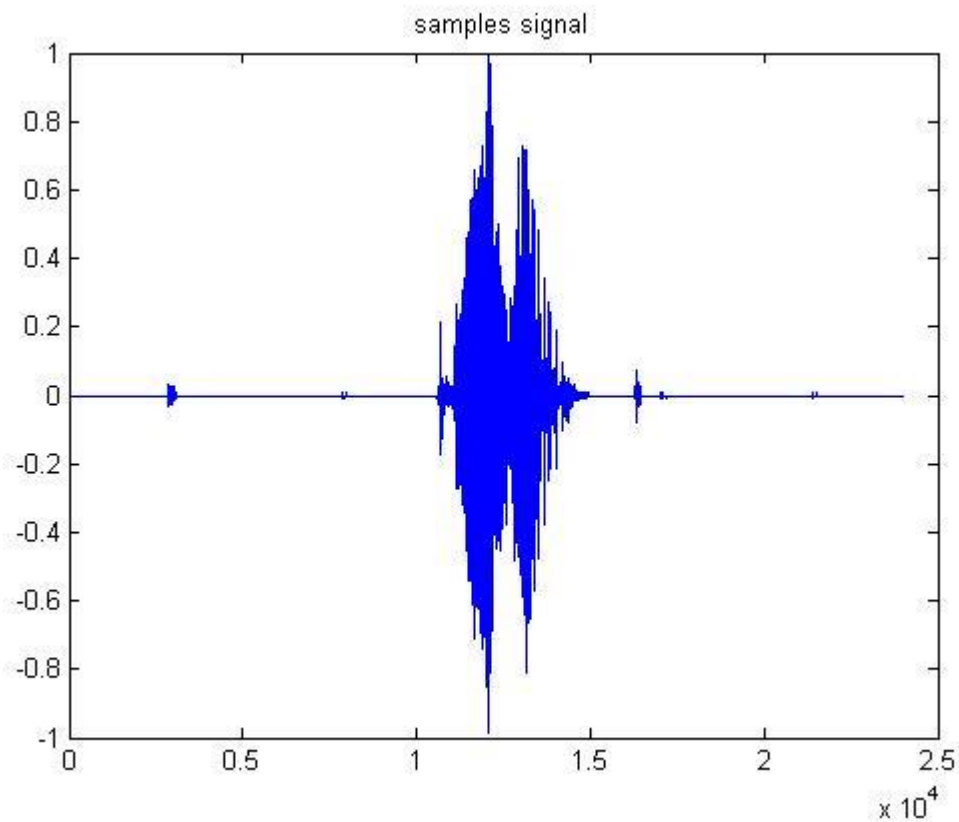
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%

save Leonardo.mat Leonardo Leonardo1 Leonardo2 Leonardo3 Leonardo4
Leonardo5 Leonardo6 Leonardo7;
```

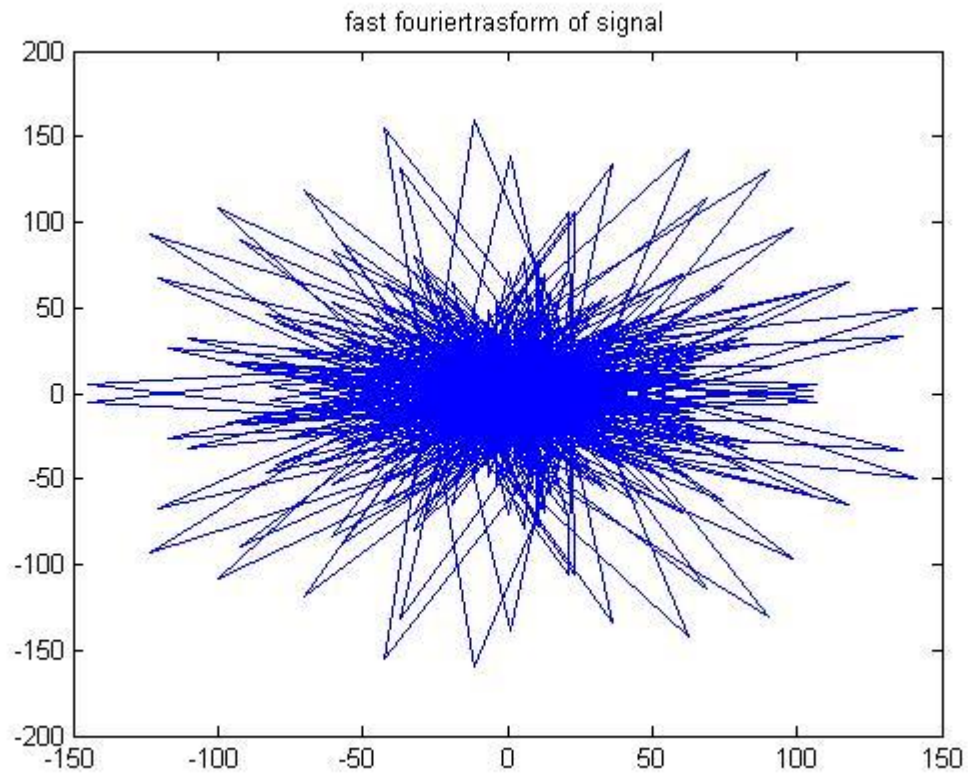
## Resultados:

Aquí se muestra el resultado con una palabra “Hola” la cual da como resultado las siguientes graficas

Señal “analógica”

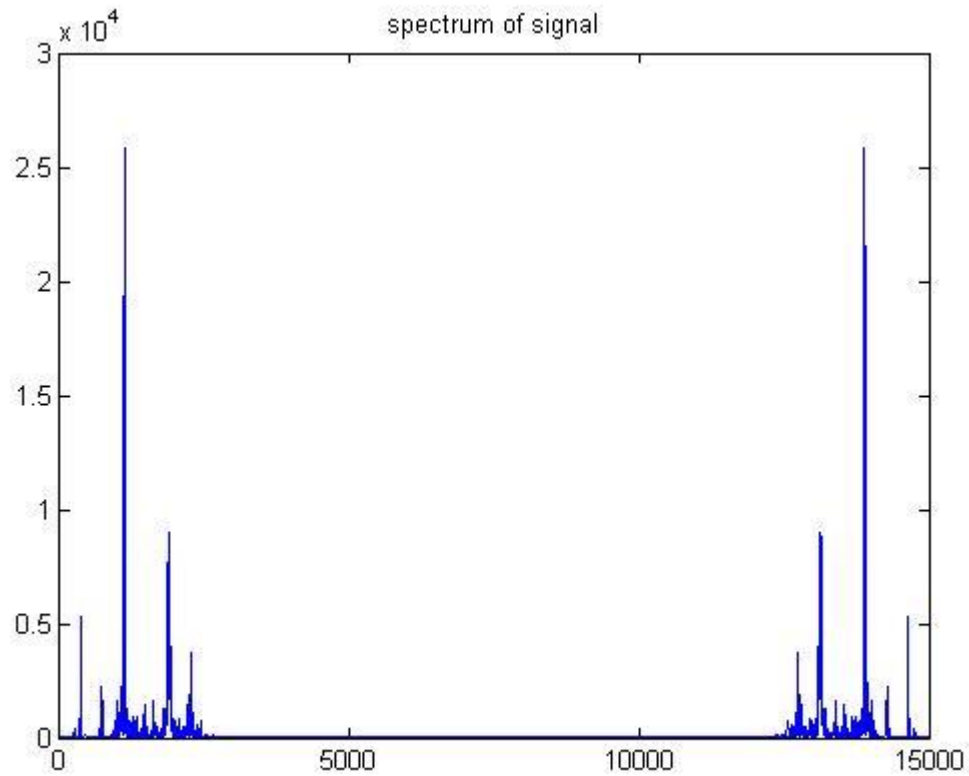


Señal en el dominio de la frecuencia (transformada de Fourier)



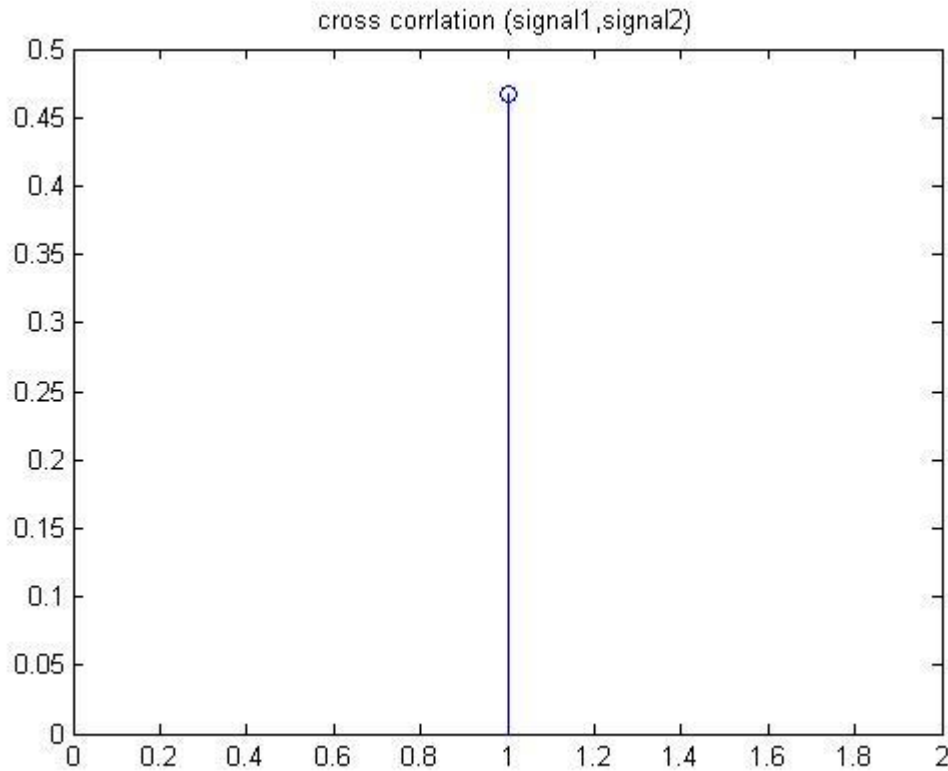
Estas son las componentes de frecuencia de la palabra Hola

### Espectro de frecuencias de la palabra Hola



Este es el espectro de frecuencias de la palabra hola en el cual se puede observar las frecuencias características

### Correlación cruzada de la palabra Hola



Este es el resultado del promedio de las correlaciones cruzadas que se le aplicaron a la señal que fue grabada en ese momento con las señales que ya estaban guardadas en el banco

### **Conclusión:**

Para tener un reconocimiento de voz eficiente se requiere tomar muchas muestras y muchos tonos de voz para aplicarles la correlación cruzada ya que se tendrá mayor referencia de lo que es una palabra y el parecido que tiene.

Cabe mencionar que dichas muestras deben ser tomadas en todo tipo de ambientes ya que no siempre se hablara en un ambiente totalmente silencioso ya que el usuario puede requerir utilizar la aplicación en un lugar donde haya sonido ambiente o quizás en donde haya mas personas hablando por ese motivo se deben tomar muestras de las señales en todos los lugares comunes en donde el usuario podría requerir el reconocimiento de voz.

Entre mas muestras haya por palabra mayor será el reconocimiento que tendrá este reconocedor de voz denominado “voice leo”