

# *‘Análisis de Señal en Canto de Ballenas’*



*Antonio Javier Samaniego Jurado  
José Manuel Quiles Pérez*

Sistemas Lineales  
Grupo A Ing. en Tec. de Telecomunicación

## ***Objetivo y estructura.***

Identificación de mamíferos marinos mediante acústica pasiva. Detección e identificación de distintos tipos de ballenas dentadas ( odontocetes ) a través del uso de sonidos grabados bajo el agua.

Este trabajo consta de dos ejercicios, los cuales se han empleando el software Matlab resultando en dos scripts .m\*, comentados a continuación y adjuntos en formato original junto a esta memoria en la actividad de la plataforma Swad Ugr.

\**En los scripts se especifica, además de en este guión, la explicación de cada procedimiento.*

## **Ejercicio 1**

Nuestro objetivo de identificación lo llevamos a cabo a través del análisis de 3 parámetros distintos para cada tipo de ballena ( Blainville, Calderón Tropical y Calderón Gris ) en base a un elemento diferenciador: el click.

Estos son:

- Longitud de click.
- Longitud interclicks.
- Rango de frecuencias del click.

Cada uno de ellos se calculará a partir de la manipulación de un fichero .wav que consiste en una grabación de la emisión de clicks de cada ballena ( tres ficheros por tipo ).

Antes de nada, nuestro primer paso es el de quitar el ruido de la muestra, pues interfiere de manera severa en nuestro análisis.

Para ello, puede diseñarse un filtro que atenúa las frecuencias del ruido.

No obstante, hemos decidido reducirlo a cero en el dominio del tiempo, pues todo el algoritmo que hemos implementado está programado para trabajar según dicha condición. Anulamos con ello toda muestra cuya amplitud sea menor que un cierto valor por debajo se aprecia con más que decente exactitud está concentrado el ruido de la señal.

Somos perfectamente conscientes de que ello priva de cierta información al click y a los parámetros estudiados, sin embargo nuestro algoritmo funciona,

como veremos, con gran aproximación según los intervalos de referencia reales del guión propuesto, de modo que aunque se pierda información, el error cometido es, aunque error, despreciable.

Dicho esto, procedemos con nuestro análisis de parámetros.

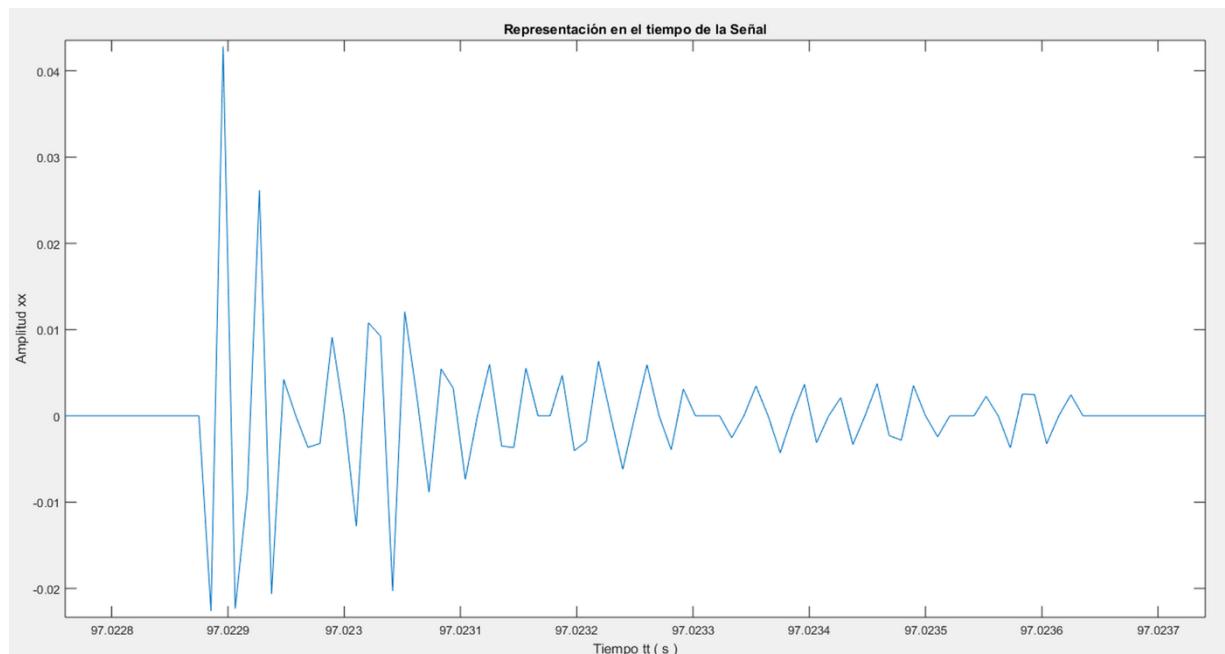
### Longitud de click

Para calcular este parámetro, partimos de que el valor muestral de mayor amplitud es, por clara evidencia de observación, parte de un click, que además en todos los casos se sigue de un click consecutivo y será nuestra referencia principal.

Hallamos por tanto la coordenada  $t$  de dicho valor, a la que llamaremos  $p_1$ .

A continuación establecemos un intervalo de tiempo de 0.001 s, de manera que trabajemos dentro del intervalo  $A \cup B = [p_1 - 0.001, p_1] \cup (p_1, p_1 + 0.001]$ . Esto es, abarcar con un tiempo más que suficiente y menor que el que tarda en llegar el reflejo, el click completo a partir de la referencia  $p_1$ .

Considerando que el ruido se ha hecho cero, gráficamente el click quedaría, para el ejemplo del tipo Calderón Gris, tal que:



De esta forma, evaluamos muestras a la izquierda y derecha de  $p_1$  de manera que, según esté en el intervalo  $A$  o  $B$ , si se llega a una distinta o igual a cero

respectivamente, se tomen las coordenadas de, en el caso de A, la muestra anterior y, en caso de B, de la misma muestra ( $(tt1-1)$  y  $tt2$ ) y se resten en valor absoluto.

Obtenemos con ello por pantalla la variable buscada ‘DuracionDelClick’.

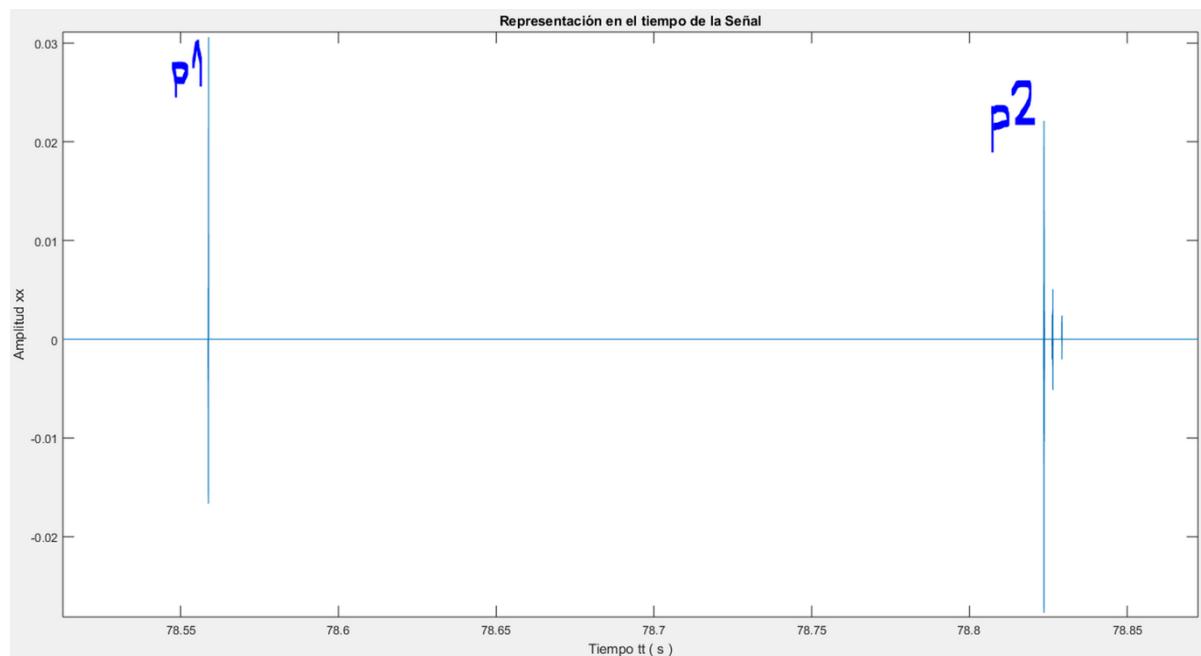
### Longitud interclicks

Para calcular este parámetro tomamos de nuevo como referencia principal la coordenada  $p1$  antes hallada.

Comenzamos eliminando el reflejo de  $p1$ . Para ello hacemos cero desde  $p1$  hasta 0.2 s a su derecha, un tiempo que abarca con gran margen dicho reflejo y, aunque haga cero la parte derecha del click, ésta ahora no es de interés, pues su presencia afecta de manera despreciable a la variable buscada.

A continuación, evaluamos muestras de  $p1$  en adelante en el eje  $tt$ , de modo que, dado que el ruido está a cero, en el momento que en amplitud alguna sea distinta de cero, guardemos su coordenada en  $tt$  como  $p2$  y la restemos en valor absoluto con  $p1$ .

Gráficamente, para el ejemplo del tipo Calderón Gris, nuestra situación es:



Obtenemos con ello por pantalla la variable buscada ‘IntervaloEntreClicks’.

## Rango de frecuencias del click

Para calcular este parámetro, pasamos la señal del dominio temporal al de la frecuencia a través de la Transformada de Fourier.

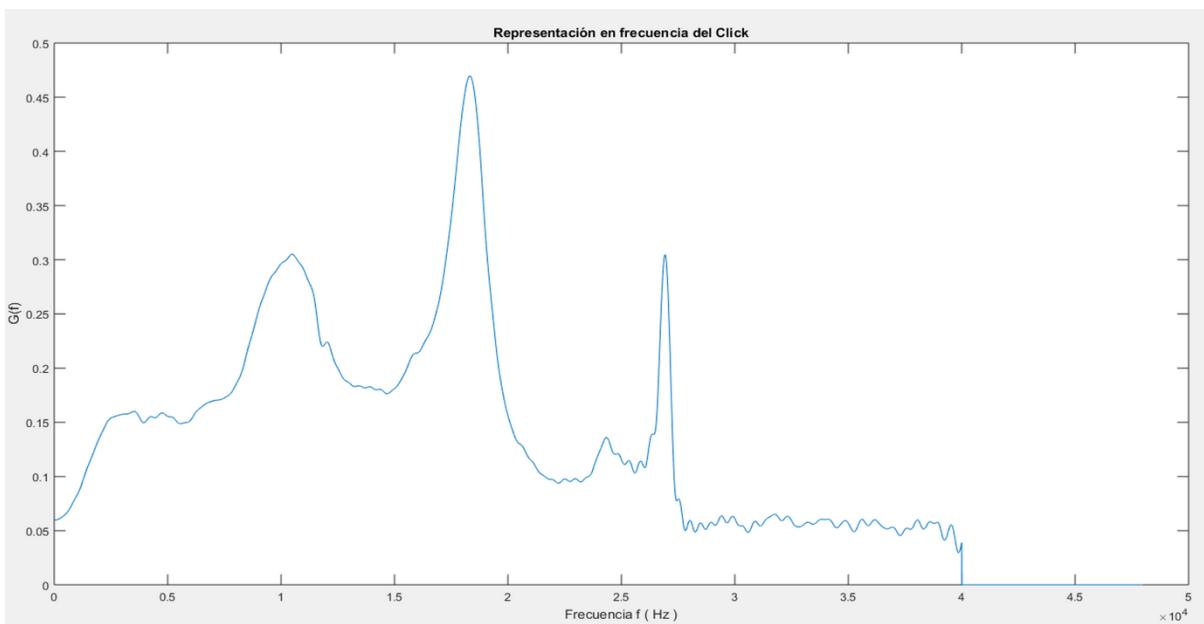
Empleamos para ello el comando de Matlab fft ( Fourier Fast Transformation ), que efectúa un algoritmo equivalente a la Transformada discreta de Fourier buscada, pues estamos trabajando con vectores de valores o muestras discretos, y no con funciones continuas.

Definimos antes adecuadamente un vector de frecuencias acorde a las longitudes de los vectores a transformar y representar, así como la función transformada G mencionada.

Dado que buscamos la frecuencia del click, transformaremos solamente éste a la frecuencia, de modo que antes de ejecutar el comando fft, hacemos cero en el dominio temporal todo valor excepto aquellos que abarque el click ( que de nuevo será el click de referencia usado en los dos apartados anteriores ).

Una vez efectuada la transformada, hallamos el valor máximo de dicha función G y obtenemos su coordenada de f ( llamándola fcentral ), de modo que a su alrededor se concentre el rango de frecuencias que abarca el click para cada ballena. Sumamos un promedio de 6000 Hz a ambos lados de dicha coordenada, un valor de frecuencia aproximado que nos dé como resultado el rango de frecuencias alrededor de dicha fcentral.

Gráficamente, para el ejemplo de Calderón Gris, nuestra situación es:



Obtenemos con ello por pantalla las variables buscadas ‘fminima’ y ‘fmaxima’ (frecuencias mínima y máxima del click para cada ballena, respectivamente ).

Debemos destacar la precisión obtenida en los resultados de los dos primeros parámetros estudiados, una precisión que, por motivos meramente técnicos, no hemos sido capaces de obtener para el rango de frecuencias.

Mostramos a continuación algunos de los parámetros obtenidos para cada uno de los 3 ficheros de cada tipo de ballena, junto al valor de la media aritmética de los mismos:

### ***LONGITUD DE CLICK***

Calderón Gris:

- 1) 1.04166666878882e-04
- 2) 1.97916666714605e-04
- 3) 1.14583333318080e-04

Media: 1.38888888970522e-04 (140 us aprox.)

Calderón Tropical:

- 1) 2.750000000006253e-04
- 2) 1.45833333328454e-04
- 3) 6.562500000022453e-04

Media: 3.59027777786667e-04 (360 us aprox.)

Blainville:

- 1) 2.500000000082991e-04
- 2) 2.91666666621381e-04
- 3) 3.43749999998650e-04

Media: 2.95138888901007e-04 (290 us aprox.)

### ***INTERVALO ENTRE CLICKS***

Calderón Gris:

- 1) 0.349072916666671
- 2) 0.641687499999996
- 3) 0.465333333333334

Media: 0.485364583333334 (0.5 s aprox.)

Calderón Tropical:

- 1) 0.116333333333333
- 2) 0.202166666666667
- 3) 0.60008333333334

Media: 0.30619444444445 (0.3 s aprox.)

Blainville:

- 1) 0.264635416666664
- 2) 0.325458333333330
- 3) 0.270156249999999

Media: 0.286749999999998 (0.3 s aprox.)

## Ejercicio 2

En este ejercicio se pide averiguar el tipo de ballena que corresponde a 4 ficheros .wav distintos a partir del script del ejercicio anterior.

Usando por tanto dicho script, añadimos una sentencia if donde imponemos que, en caso de cumplirse que los parámetros estudiados estén dentro de ciertos intervalos teóricos ( los propuestos del guión con un margen de error por razones experimentales y de irregularidad en la naturaleza ), se muestre por pantalla el tipo de ballena al que corresponde cada fichero.

No obstante, partiendo de lo dicho al final del Ejercicio 1 de este guión, hemos excluido del ‘if’ el rango de frecuencias, pues reiteramos que la precisión obtenida no ha sido óptima y, lejos de ayudar a identificar las ballenas, contribuye a dar error en el análisis.

Personalmente creemos que si este proyecto fuese para alguna empresa real, velar ante todo por la exactitud del producto/resultado final puede llevar consigo tomar decisiones como ésta.

Tomaremos por tanto como ‘filtro identificador’ los parámetros de la longitud del click y longitud interclicks.

Desafortunadamente, obtenemos un resultado fiable para los dos primeros ficheros de cuatro, mientras que para los dos restantes el filtro no es capaz, de nuevo por motivos técnicos, de aportar precisión suficiente como para realizar la identificación.

Conseguimos con ello y con creces un 50% por ciento de este Ejercicio 2, lo cual, al margen de el ‘medio fracaso’, también supone un ‘medio éxito’.

Obtenemos que el 1er fichero corresponde al tipo Calderón Gris y el 2do al tipo Blainville.

En la exposición de este trabajo, no obstante, mostraremos como sería la sentencia if si añadiésemos el parámetro de rango de frecuencias dentro de ese ‘filtro identificador’.

---

Damos con ello por concluido este trabajo, que ha sido de gran utilidad, pues nos ha permitido mejorar nuestra habilidad mental, de coordinación, resolución de problemas, síntesis y análisis de información.

Ha sido un gran reto para nosotros, como futuros ingenieros, del que hemos aprendido en base tanto a los errores como los aciertos, hacer cosas mal y cosas bien.

*Nota importante:* En la exposición de este trabajo se mostrarán en detalle los razonamientos y cálculos matemáticos que hemos llevado a cabo para hacer de lo expuesto en este guión algo viable para la herramienta Matlab.

## MATLAB Code

### Ejercicio 1:

```
clear all;
close all;
clc;
format long

% cargamos datos
[xx,fs] = wavread('./Datos de entrenamiento/Blainville_1.wav');
y=xx;
% fs es la frecuencia de muestreo (muestras/segundo)
% xx son los datos de sonido

% construimos el vector de tiempo en segundos
tt = (1:length(xx))/fs;
```

```

%Quitamos el ruido
for i=1:length(tt)
    if abs(xx(i))<0.002
        xx(i)=0;
    end
end

%Partimos de que el valor muestral de mayor
%amplitud es, por clara observaciÛn, un click.

%Buscamos las coordenadas de tt y xx del click de mayor amplitud.
z=max(xx);
for i=1:length(tt)
    if abs(xx(i))==z
        break
    end
end
PicoMaximo=xx(i);
p1=tt(i);

%Definimos el perÌodo de muestreo (segundos transcurridos entre
cada
%muestra).
T=1/fs;

%DURACI"N DEL CLICK.
%Hallamos la duraciÛn del click (96*T equivale a 0.001s, tiempo que
no
%interfiere, tras observar el gr·fico en el tiempo, en la variable
%estudiada).
for i=1:length(tt)
    if (tt(i)>p1-96*T & tt(i)< p1)
        if xx(i)~=0;
            break
        end
    end
end

tt1=tt(i-1);

for i=1:length(tt)
    if (tt(i)>p1 & tt(i)< p1+96*T)
        if xx(i)==0;
            break
        end
    end
end

```

```

tt2=tt(i);

DuracionDelClick=abs(tt2-tt1)

%INTERVALO ENTRE CLICKS
%Eliminamos el reflejo del click mencionado (9600*T equivale a
0.2s,
%tiempo que no interfiere, tras observar las muestras en el tiempo,
en la
%variable estudiada)(es decir, hacer cero xx desde el click hasta
0.2s a m·s a la derecha
%(valor promedio que tarda en reflejarse (y no interfiere con el
intervalo
%entre clicks buscado)).
for i=1:length(tt)
    if (tt(i)>p1 & tt(i)< p1+9600*T)
        xx(i)=0;
    end
end

%Evaluamos en amplitud las muestras desde el click mencionado hacia
la
%derecha, de modo que cuando ´sta sea distinta de cero, sabiendo que
hemos
%eliminado el ruido, nos encontramos en el click siguiente.
Obtenemos así la
%coordenadas de tt del mismo y, restándola con la del primer click,
%obtenemos el intervalo entre clicks buscado.
for i=1:length(tt)
    if tt(i)>p1
        if xx(i)~=0
            break
        end
    end
end

p2=tt(i);
IntervaloEntreClicks=p2-p1

%Representamos los datos en el dominio del tiempo
figure(2)
plot(tt,xx)
title('RepresentaciÛn en el tiempo de la SeÛal');
xlabel('Tiempo tt ( s )');
ylabel('Amplitud xx');

%Frecuencia.
%Eliminamos en el tiempo todo menos un click (el evaluado antes de
m·xima
%amplitud). Para ello dejamos un intervalo de valor 100*T (100

```

```

muestras) a
%un lado y otro de la coordenada tt de dicho click.

for i=1:length(tt)
    if tt(i)<p1-200*T
        y(i)=0;
    elseif tt(i)>p1+200*T
        y(i)=0;
    end
end

%Creamos el vector de frecuencias y hacemos la Transformada de
Fourier
%discreta del vector xx.
h=fs./length(y);
f=-fs/2:h:(fs/2)-h;
G=abs(fft(y));

%Quitamos frecuencias superfluas en base a los intervalos teÓricos
(a
%partir de 40000 aprox.).
for i=1:length(f)
    if f(i)>40000
        G(i)=0;
    elseif f(i)<-40000
        G(i)=0;
    end
end

%Representamos los datos en el dominio de la frecuencia (Hz).
figure(1)
plot(abs(f),G);
title('RepresentaciÛn en frecuencia del Click');
xlabel('Frecuencia f ( Hz )');
ylabel('G(f)');

%Buscamos la frecuencia con mayor valor en amplitud
k=max(G);
for i=1:length(f)
    if abs(G(i))==k
        break
    end
end

fcentral=abs(f(i));
fminima=fcentral-2000
fmaxima=fcentral+2000

```

## Ejercicio 2:

```
clear all;
close all;
clc;
format long

% cargamos datos
[xx,fs] = wavread('./Datos de test/test_1.wav');
y=xx;
% fs es la frecuencia de muestreo (muestras/segundo)
% xx son los datos de sonido

% construimos el vector de tiempo en segundos
tt = (1:length(xx))'/fs;

%Quitamos el ruido
for i=1:length(tt)
    if abs(xx(i))<0.002
        xx(i)=0;
    end
end

%INTERVALO ENTRE CLICKS
%Partimos de que el valor muestral de mayor
%amplitud es, por clara observaciÛn, un click.

%Buscamos las coordenadas de tt y xx del click de mayor amplitud.
z=max(xx);
for i=1:length(tt)
    if abs(xx(i))==z
        break
    end
end
PicoMaximo=xx(i);
p1=tt(i);

%Definimos el perÌodo de muestreo (segundos transcurridos entre
cada
%muestra).
T=1/fs;

%DURACI"ON DEL CLICK.
```

```

%Hallamos la duraci n del click (96*T equivale a 0.001s, tiempo que
no
%interfiere, tras observar el gr fico en el tiempo, en la variable
%estudiada).
for i=1:length(tt)
    if (tt(i)>p1-96*T & tt(i)<p1)
        if xx(i)~=0;
            break
        end
    end
end

tt1=tt(i-1);

for i=1:length(tt)
    if (tt(i)>p1 & tt(i)<p1+96*T)
        if xx(i)==0;
            break
        end
    end
end

tt2=tt(i);

DuracionDelClick=abs(tt2-tt1);

%Eliminamos el reflejo del click mencionado (9600*T equivale a
0.2s,
%tiempo que no interfiere, tras observar las muestras en el tiempo,
en la
%variable estudiada)(es decir, hacer cero xx desde el click hasta
0.2s a m s a la derecha
%(valor promedio que tarda en reflejarse (y no interfiere con el
intervalo
%entre clicks buscado)).
for i=1:length(tt)
    if (tt(i)>p1 & tt(i)<p1+9600*T)
        xx(i)=0;
    end
end

%Evaluamos en amplitud las muestras desde el click mencionado hacia
la
%derecha, de modo que cuando  sta sea distinta de cero, sabiendo que
hemos
%eliminado el ruido, nos encontraremos en el click siguiente.
Obtenemos as  la
%coordenadas de tt del mismo y, rest ndola con la del primer click,
%obtenemos el intervalo entre clicks buscado.
for i=1:length(tt)
    if tt(i)>p1
        if xx(i)~=0
            break

```

```

        end
    end
end

p2=tt(i);
IntervaloEntreClicks=p2-p1;

%Frecuencia.

for i=1:length(tt)
    if tt(i)<p1-200*T
        y(i)=0;
    elseif tt(i)>p1+200*T
        y(i)=0;
    end
end

%Creamos el vector de frecuencias y hacemos la Transformada de
Fourier
%discreta del vector xx.
h=fs./length(y);
f=-fs/2:h:(fs/2)-h;
G=abs(fft(y));

%Quitamos frecuencias superfluas en base a los intervalos teóricos
(a
%partir de 40000 aprox.).
for i=1:length(f)
    if f(i)>40000
        G(i)=0;
    elseif f(i)<-40000
        G(i)=0;
    end
end

%Buscamos la frecuencia con mayor valor en amplitud
k=max(G);
for i=1:length(f)
    if abs(G(i))==k
        break
    end
end

fcentral=abs(f(i));
fminima=fcentral-6000;
fmaxima=fcentral+6000;

```

%Si cumplen las caracterÌsticas teÛricas de alguna especie (con  
%m·rgenes de error por cuestiÛn de aproximaciones  
experimentalas), se muestra por  
%pantalla de quÈ especie se trata el conjunto de clicks analizados  
(Ejercicio 2).

```
if (DuracionDelClick>0.000100 & DuracionDelClick<0.000300)
    if (IntervaloEntreClicks>0.2 & IntervaloEntreClicks<0.4)
        disp('La especie analizada es Blainville');
    end
end

if (DuracionDelClick>0.000200 & DuracionDelClick<0.000300)
    if (IntervaloEntreClicks>0.4 & IntervaloEntreClicks<0.59)
        disp('La especie analizada es CalderÛn Tropical');
    end
end

if (DuracionDelClick>0 & DuracionDelClick<0.000150)
    if (IntervaloEntreClicks>0.4 & IntervaloEntreClicks<0.7)
        disp('La especie analizada es CalderÛn Gris');
    end
end
```