

# TRABAJO FINAL INTELIGENCIA ARTIFICIAL I

AÑO 2017

ALUMNA: TORRES LÓPEZ, MÍA  
LEGAJO: 10991  
PROFESORA: DRA. ING. SELVA S. RIVERA

## RESUMEN

El objetivo del trabajo es diseñar un agente de aprendizaje para un robot que pinta piezas mecánicas de color ROJO, AMARILLO o VERDE recibiendo comandos de voz. El agente debe aprender a diferenciar entre las palabras ROJO, AMARILLO y VERDE de modo tal que el robot pueda aplicar un color en función de la orden dada por el usuario.

## INTRODUCCIÓN

El objetivo del trabajo es diseñar un agente capaz de diferenciar las palabras ROJO, AMARILLO y VERDE dadas al mismo por comandos de voz, para que pueda pintar piezas mecánicas de acuerdo al color indicado. Para esto, el agente utilizará dos algoritmos de Aprendizaje: K-means y K-nn. Se busca evaluar el rendimiento de ambos algoritmos para decidir cuál de ellos es la mejor solución. La implementación de los algoritmos fue realizada en el IDE (entorno de desarrollo integrado) de MathWorks **MatLab** versión R2014b, en plataforma Windows.

Se utilizó un micrófono externo para capturar el audio y como vector de características de cada captura realizada se emplearon “cepstrums”, el resultado de calcular transformada inversa de Fourier del espectro de la señal estudiada en escala logarítmica.

Se realizó un prototipo del problema, utilizando una placa **Arduino UNO** y un circuito de LEDs con los colores rojo, amarillo y verde que deben encenderse al recibir el comando de voz.

**Tipo de agente:** es un “agente que aprende”, es decir que utiliza las percepciones no sólo para actuar sino también para mejorar la habilidad del agente para actuar en el futuro.

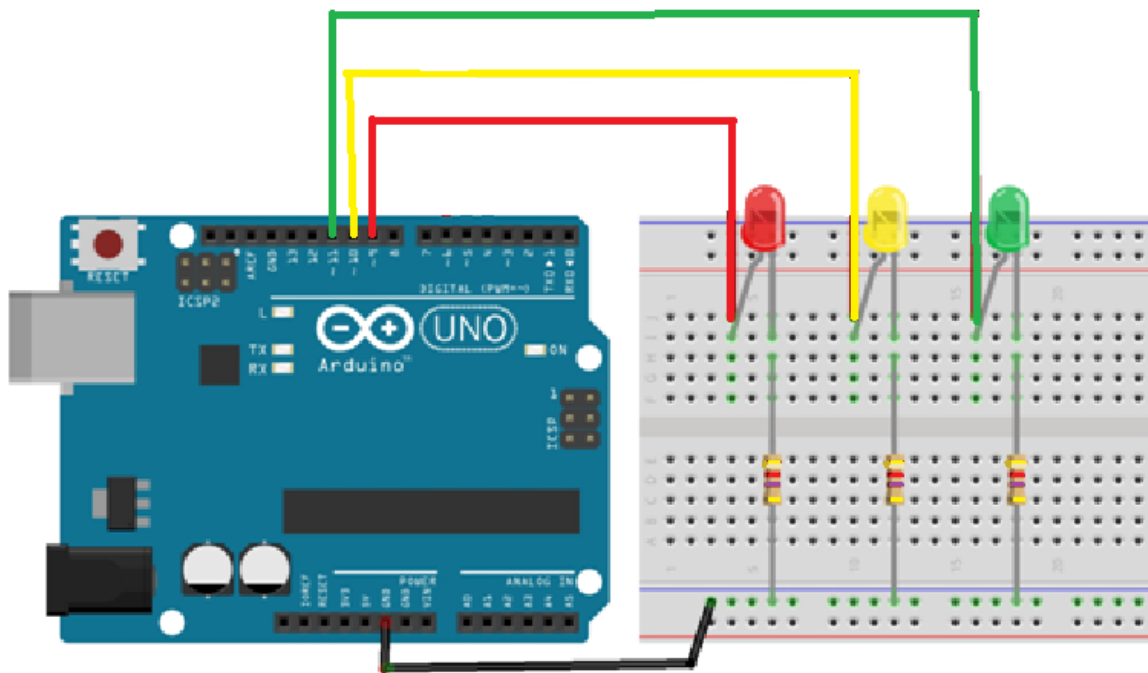
A continuación se muestra la **tabla REAS** del agente:

Agente	Rendimiento	Entorno	Actuadores	Sensores
Robot que pinta piezas mecánicas	Maximizar la coincidencia del color que pinta la pieza con el color que le indica el usuario	Fabrica de piezas	Brazo neumático para aplicar la pintura, aerosoles rojo, amarillo y verde	Sensor de color y de sonido.

Entorno	Observable	Determinista	Episódico	Estática	Discreto	Multiagente
Fabrica de piezas	Totalmente Observable	Sí	Sí	No. Dinámico	Sí	No. Agente individual

## DISEÑO DEL SISTEMA

Para el diseño del prototipo, se siguió el siguiente esquema de conexiones:



Se utilizó una placa **Arduino UNO**, usando el pin GND (tierra) representado por el cable negro, conectándose a las resistencias las cuales se van a la masa de cada LED. Los cables de colores rojo, amarillo y verde conectan los LED a los pines digitales números 9, 10 y 11.

El diseño del programa del agente fue realizado en **Matlab** y consiste en grabar audio, obtener el vector de características y aplicar los algoritmos para luego comparar su rendimiento. Al iniciar, el usuario tiene la posibilidad de reescribir la base de datos utilizando un micrófono externo y deberá decir tres veces cada color (rojo, amarillo y verde) para poder continuar. Si lo desea puede utilizar nueve pistas ya grabadas (tres veces cada color) para guardar en la base de datos en vez de tener que grabarlas el usuario mediante el micrófono. Cada vez que se agregue un audio de color, se calcula el “cepstrum” del mismo, para utilizarlo como vector de características y ese vector se guarda en la base de datos. Si ya tiene una base de datos guardada, puede elegir no reescribirla si acaba de iniciar el programa.

Una vez lista la base de datos, se pide al usuario que diga que color desea clasificar. Deberá indicar el color utilizando el micrófono y el programa calculará el cepstrum del mismo. Entonces el usuario puede elegir que algoritmo utilizar para realizar la clasificación y dependiendo de su elección, comenzará el algoritmo K-means o K-nn. Una vez que los

algoritmos terminan la clasificación, se encenderá el LED de acuerdo al color que el algoritmo haya determinado.

Por último, se pide al usuario que indique si el color determinado por el algoritmo es efectivamente el color que indicó la persona y cuando el usuario decide terminar el programa, se calcula cuál de los dos algoritmos resultó más efectivo.

## CÓDIGO

Se anexa el código completo al final del informe. En este apartado se explican sólo partes del mismo.

- ❖ Para grabar la voz, obtener el cepstrum de esa señal y guardarlo en la base de datos:

```
%cepstrum de 'AMARILLO'
disp('Diga AMARILLO')
recordblocking(recObj,1);
disp('Fin de la grabacion')
myRecording = getaudiodata(recObj);
cepstrum=ifft(log(abs(fft(myRecording))));
baseDatos=[baseDatos,cepstrum];
```

- ❖ **K-means:** se eligen como centroides puntos de la base de datos

```
centroide1=baseDatos(:,1);
centroide2=baseDatos(:,4);
centroide3=baseDatos(:,7);
```

- ❖ Se calcula la distancia euclidiana de cada dato de la base de datos a cada cepstrum y luego se comparan las distancias para colocar cada dato en un cluster (grupo)

```
for j=1:length(baseDatos(:,1)) %8000
    dist1=dist1+(centroide1(j)-baseDatos(j,i))^2;
    dist2=dist2+(centroide2(j)-baseDatos(j,i))^2;
    dist3=dist3+(centroide3(j)-baseDatos(j,i))^2;
end
dist1=dist1^0.5;
dist2=dist2^0.5;
dist3=dist3^0.5;

if(dist1<dist2 && dist1<dist3)
    if(length(grupo1(:,1))==1) && (length(grupo1(1,:))==1)
        grupo1=baseDatos(:,i);
    else
        grupo1=[grupo1,baseDatos(:,i)];
    end
end
end
```

- ❖ Se recalculan los centroides

```
for i=1:length(centroide1(:))
    centroide1(i)=0;
    for j=1:length(grupo1(1,:))
        centroide1(i)=centroide1(i)+grupo1(i,j);
    end
    centroide1(i)=centroide1(i)/length(grupo1(1,:));
end
```

- ❖ Una vez finalizadas las iteraciones, se busca en que cluster está el color que el usuario desea clasificar y se clasifica de acuerdo a que color predomine en el cluster. Por último, se enciende el LED correspondiente.

```
if ismember(cepstrumNuevo, grupo1)
    for j=1:(length(grupo1(1,:))-1)
        for i=1:3
            if grupo1(:,j)==baseDatos(:,i)
                cantR=cantR+1;
            end
        end
        for i=4:6
            if grupo1(:,j)==baseDatos(:,i)
                cantA=cantA+1;
            end
        end
        for i=7:9
            if grupo1(:,j)==baseDatos(:,i)
                cantV=cantV+1;
            end
        end
    end
end

if(cantR > cantA) && (cantR > cantV)
    disp('Dijo ROJO')
    writeDigitalPin(ard,9,1);
    pause(2);
    writeDigitalPin(ard,9,0);
end
```

- ❖ **K-nn:** Realiza iteraciones en las que va aumentando el radio hasta encontrar a K=2 vecinos y de acuerdo al color de ese vecinos, determina qué color dijo el usuario

```
for i=1:3 %Rojo
    dist=0;
    for j=1:length(baseDatos(:,1)) %8000
        dist=dist+(cepstrumNuevo(j)-baseDatos(j,i))^2;
    end
    dist=dist^0.5;
    if dist<radio
        vecinos=vecinos+1;
        cantR=cantR+1;
    end
    if vecinos==K
        break;
    end
end
radio=radio+0.03; %Aumento el radio
```

- ❖ Enciende el LED de acuerdo a la clasificación que se le dio por sus vecinos más cercanos

```
if(cantR > cantA) && (cantR > cantV)
    disp('Dijo ROJO')
    writeDigitalPin(ard,9,1);
    pause(2);
    writeDigitalPin(ard,9,0);
end
```



## EJEMPLO DE APLICACIÓN

Vemos que cuando el usuario inicia el programa puede elegir si reescribir o no la base de datos, y si quiere reescribirla puede hacerlo utilizando el micrófono o pistas previamente grabadas. En este caso se eligieron las pistas y el usuario grabó el color y probó la misma grabación con ambos algoritmos, primero K-means y luego K-nn, obteniendo el resultado correcto en ambas ocasiones:

```
Command Window
>> ProyectoFinal_TORRES
¿Desea reescribir la base de datos? (M)icrófono; pistas (G)rabadas; (N)o reescribir G
CLASIFICACIÓN
Presione la tecla "ENTER" y diga ROJO, AMARILLO o VERDE:
Fin de la grabación
Seleccione el algoritmo a utilizar para la clasificación: (1) K-means (2)K-nn 1
Dijo ROJO
Es correcto? (S)i o (N)o S
¿Desea Terminar el programa? (T)erminar
¿Desea volver a grabar el color a probar? (S)i o (N)o N
Seleccione el algoritmo a utilizar para la clasificación: (1) K-means (2)K-nn 2
Dijo ROJO
```

Si el usuario decide utilizar el micrófono para completar la base de datos, verá la siguiente pantalla, donde deberá grabar cada color tres veces:

```
Command Window
>> ProyectoFinal_TORRES
¿Desea reescribir la base de datos? (M)icrófono; pistas (G)rabadas; (N)o reescribir M
Presione la tecla "ENTER" cuando esté listo para grabar:
Diga ROJO
Fin de la grabación
Presione la tecla "ENTER" cuando esté listo para grabar:
Diga ROJO
Fin de la grabación
Presione la tecla "ENTER" cuando esté listo para grabar:
Diga ROJO
Fin de la grabación
Presione la tecla "ENTER" cuando esté listo para grabar:
Diga AMARILLO
Fin de la grabacion
Presione la tecla "ENTER" cuando esté listo para grabar:
Diga AMARILLO
Fin de la grabacion
Presione la tecla "ENTER" cuando esté listo para grabar:
Diga AMARILLO
Fin de la grabacion
Presione la tecla "ENTER" cuando esté listo para grabar:
Diga VERDE
Fin de la grabacion
Presione la tecla "ENTER" cuando esté listo para grabar:
Diga VERDE
Fin de la grabacion
Presione la tecla "ENTER" cuando esté listo para grabar:
```

## RESULTADOS

Cuando el usuario decide finalizar el programa, de acuerdo a la cantidad total de veces que corrió cada algoritmo y a los aciertos y errores cometidos, se indica cuál de ellos resultó más efectivo:

```
Command Window
REPORTE DE RESULTADOS

K-means corrió 7 veces, acertando 3 veces y errando 4 veces
K-nn corrió 9 veces, acertando 6 veces y errando 3 veces
Knn fue más efectivo
```

## CONCLUSIONES

Luego de realizar varias pruebas, se llegó a la conclusión que el algoritmo K-nn suele ser más efectivo que el algoritmo K-means. Sin embargo, se observó que esto puede variar si la base de datos no es buena, observación que llevó a incluir pistas grabadas al programa para mejorar su base de datos. Ambos algoritmos ven afectado su rendimiento si el lugar de prueba es muy ruidoso o si no se utiliza el micrófono correctamente. Además, la pronunciación clara y fuerte del usuario ayuda a obtener mejores resultados, mientras que si, por ejemplo, el usuario pronuncia la letra “R” de forma más suave que las grabaciones de las pistas de audio, los resultados de ambos algoritmos se ven afectados. Por lo tanto, se concluye que, si bien K-nn funciona mejor, se debe tener una buena base de datos y dar órdenes claras para mejorar los resultados del programa.

## BIBLIOGRAFÍA

<http://fing.uncu.edu.ar/catedras/inteligencia-artificial-i>

<https://www.mathworks.com>

*“Inteligencia Artificial. Un enfoque moderno”*, 2da edición. Russell, S. J; Norvig, P.

## ANEXO

### Programa del agente completo

```
ard=arduino('COM4','UNO')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% BASE DE DATOS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
prompt='¿Desea reescribir la base de datos? (M)icrófono; pistas
(G)rabadas; (N)o reescribir ';
str = input(prompt,'s');
if (str=='M')
    %Para poder grabar la voz
    recObj=audiorecorder;
    %borro la base de datos anterior
    clear baseDatos;
    baseDatos=zeros(1,1); %para mejorar performance del programa
    (reallocate)
    prompt='Presione la tecla "ENTER" cuando esté listo para grabar: ';
    contador=1;
    while (contador<10)
        if (contador<=3)
            str = input(prompt, 's');
            %cepstrum de 'ROJO'
            disp('Diga ROJO')
            recordblocking(recObj, 1); %grabo la voz
            disp('Fin de la grabación')
            myRecording = getaudiodata(recObj); %obtengo los datos con
            doble precision
            cepstrum=ifft(log(abs(fft(myRecording))));
            if (contador==1)
                baseDatos=cepstrum;
            else
                baseDatos=[baseDatos,cepstrum];
            end
            contador=contador+1;
        end
        if(contador>3 && contador<=6)
            str = input(prompt, 's');
            %cepstrum de 'AMARILLO'
            disp('Diga AMARILLO')
            recordblocking(recObj,1);
            disp('Fin de la grabacion')
            myRecording = getaudiodata(recObj);
            cepstrum=ifft(log(abs(fft(myRecording))));
            contador=contador+1;
            baseDatos=[baseDatos,cepstrum];
        end
        if(contador>6 && contador<=9)
            str = input(prompt, 's');
            %cepstrum de 'VERDE'
            disp('Diga VERDE')
            recordblocking(recObj,1);
            disp('Fin de la grabacion')
            myRecording = getaudiodata(recObj);
            cepstrum=ifft(log(abs(fft(myRecording))));
            contador=contador+1;
```

```
        baseDatos=[baseDatos,cepstrum];
    end
end

if (str=='G')
    %borro la base de datos anterior
    clear baseDatos;
    myRecording=audioread('Rojo1.wav');
    cepstrum=ifft(log(abs(fft(myRecording))));
    baseDatos=cepstrum;
    myRecording=audioread('Rojo2.wav');
    cepstrum=ifft(log(abs(fft(myRecording))));
    baseDatos=[baseDatos,cepstrum];
    myRecording=audioread('Rojo3.wav');
    cepstrum=ifft(log(abs(fft(myRecording))));
    baseDatos=[baseDatos,cepstrum];
    myRecording=audioread('Amarillo1.wav');
    cepstrum=ifft(log(abs(fft(myRecording))));
    baseDatos=[baseDatos,cepstrum];
    myRecording=audioread('Amarillo2.wav');
    cepstrum=ifft(log(abs(fft(myRecording))));
    baseDatos=[baseDatos,cepstrum];
    myRecording=audioread('Amarillo3.wav');
    cepstrum=ifft(log(abs(fft(myRecording))));
    baseDatos=[baseDatos,cepstrum];
    myRecording=audioread('Verde1.wav');
    cepstrum=ifft(log(abs(fft(myRecording))));
    baseDatos=[baseDatos,cepstrum];
    myRecording=audioread('Verde2.wav');
    cepstrum=ifft(log(abs(fft(myRecording))));
    baseDatos=[baseDatos,cepstrum];
    myRecording=audioread('Verde3.wav');
    cepstrum=ifft(log(abs(fft(myRecording))));
    baseDatos=[baseDatos,cepstrum];
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ELECCIÓN DEL ALGORITMO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
KmeansBien=0;
KmeansMal=0;
KnnBien=0;
KnnMal=0;

disp('CLASIFICACIÓN');
prompt='Presione la tecla "ENTER" y diga ROJO, AMARILLO o VERDE: ';
str=input(prompt,'s');
recordblocking(recObj,1);
disp('Fin de la grabación')
myRecording = getaudiodata(recObj);
cepstrumNuevo=ifft(log(abs(fft(myRecording))));
baseDatos(:,10)=cepstrumNuevo;

while (1)
    prompt='Seleccione el algoritmo a utilizar para la clasificación: (1)
K-means (2)K-nn ';
    algoritmo=input(prompt,'s');
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% K-MEANS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if algoritmo=='1'
    %k centroides (k=3)
    centroide1=baseDatos(:,1); %rojo
    centroide2=baseDatos(:,4); %amarillo
    centroide3=baseDatos(:,7); %verde

    %iteracion
    for k=1:10

        %k grupos
        grupo1=zeros(1,1); %centroide1
        grupo2=zeros(1,1); %centroide2
        grupo3=zeros(1,1); %centroide3

        for i=1:length(baseDatos(1,:)) %10
            %calculo las distancias euclidianas de cada dato al
            %centroide
            dist1=0;
            dist2=0;
            dist3=0;
            for j=1:length(baseDatos(:,1)) %8000
                dist1=dist1+(centroide1(j)-baseDatos(j,i))^2;
                dist2=dist2+(centroide2(j)-baseDatos(j,i))^2;
                dist3=dist3+(centroide3(j)-baseDatos(j,i))^2;
            end
            dist1=dist1^0.5;
            dist2=dist2^0.5;
            dist3=dist3^0.5;

            %a qué grupo pertenece cada punto de acuerdo al centroide
            if(dist1<dist2 && dist1<dist3)
                if(length(grupo1(:,1))==1) &&
                (length(grupo1(1,:))==1)
                    grupo1=baseDatos(:,i);
                else
                    grupo1=[grupo1,baseDatos(:,i)];
                end
            end
            if(dist2<dist1 && dist2<dist3)
                if(length(grupo2(:,1))==1) &&
                (length(grupo2(1,:))==1)
                    grupo2=baseDatos(:,i);
                else
                    grupo2=[grupo2,baseDatos(:,i)];
                end
            end
            if(dist3<dist1 && dist3<dist2)
                if(length(grupo3(:,1))==1) &&
                (length(grupo3(1,:))==1)
                    grupo3=baseDatos(:,i);
                else
                    grupo3=[grupo3,baseDatos(:,i)];
                end
            end
        end
    end
end
```

```
end

%recalculo los centroides
for i=1:length(centroide1(:))
    centroide1(i)=0;
    for j=1:length(grupo1(1,:))
        centroide1(i)=centroide1(i)+grupo1(i,j);
    end
    centroide1(i)=centroide1(i)/length(grupo1(1,:));
end
for i=1:8000
    centroide2(i)=0;
    for j=1:length(grupo2(1,:))
        centroide2(i)=centroide2(i)+grupo2(i,j);
    end
    centroide2(i)=centroide2(i)/length(grupo2(1,:));
end
for i=1:8000
    centroide3(i)=0;
    for j=1:length(grupo3(1,:))
        centroide3(i)=centroide3(i)+grupo3(i,j);
    end
    centroide3(i)=centroide3(i)/length(grupo3(1,:));
end

end %fin iteraciones
```

```
cantR=0;
cantA=0;
cantV=0;
if ismember(cepstrumNuevo,grupo1)
    for j=1:(length(grupo1(1,:))-1)
        for i=1:3
            if grupo1(:,j)==baseDatos(:,i)
                cantR=cantR+1;
            end
        end
        for i=4:6
            if grupo1(:,j)==baseDatos(:,i)
                cantA=cantA+1;
            end
        end
        for i=7:9
            if grupo1(:,j)==baseDatos(:,i)
                cantV=cantV+1;
            end
        end
    end
end
if ismember(cepstrumNuevo,grupo2)
    for j=1:(length(grupo2(1,:))-1)
        for i=1:3
            if grupo2(:,j)==baseDatos(:,i)
                cantR=cantR+1;
            end
        end
    end
end
```

```

        end
        for i=4:6
            if grupo2(:,j)==baseDatos(:,i)
                cantA=cantA+1;
            end
        end
        for i=7:9
            if grupo2(:,j)==baseDatos(:,i)
                cantV=cantV+1;
            end
        end
    end
end
if ismember(cepstrumNuevo,grupo3)
    for j=1:(length(grupo3(1,:))-1)
        for i=1:3
            if grupo3(:,j)==baseDatos(:,i)
                cantR=cantR+1;
            end
        end
        for i=4:6
            if grupo3(:,j)==baseDatos(:,i)
                cantA=cantA+1;
            end
        end
        for i=7:9
            if grupo3(:,j)==baseDatos(:,i)
                cantV=cantV+1;
            end
        end
    end
end

%resultados
if(cantR > cantA) && (cantR > cantV)
    disp('Dijo ROJO')
    writeDigitalPin(ard,9,1);
    pause(2);
    writeDigitalPin(ard,9,0);
end
if(cantA > cantR) && (cantA > cantV)
    disp('Dijo AMARILLO')
    writeDigitalPin(ard,10,1);
    pause(2);
    writeDigitalPin(ard,10,0);
end
if(cantV > cantA) && (cantV > cantR)
    disp('Dijo VERDE')
    writeDigitalPin(ard,11,1);
    pause(2);
    writeDigitalPin(ard,11,0);
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% KNN %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if(algoritmo=='2')
    radio=0;

```

```
K=2; %muestras mas cercanas
vecinos=0;
cantR=0;
cantA=0;
cantV=0;
while (vecinos~=K) %vecinos~=K
    if vecinos==K
        break;
    end
    for i=1:3 %Rojo
        dist=0;
        for j=1:length(baseDatos(:,1)) %8000
            dist=dist+(cepstrumNuevo(j)-baseDatos(j,i))^2;
        end
        dist=dist^0.5;
        if dist<radio
            vecinos=vecinos+1;
            cantR=cantR+1;
        end
        if vecinos==K
            break;
        end
    end
    for i=4:6 %Amarillo
        dist=0;
        for j=1:length(baseDatos(:,1)) %8000
            dist=dist+(cepstrumNuevo(j)-baseDatos(j,i))^2;
        end
        dist=dist^0.5;
        if dist<radio
            vecinos=vecinos+1;
            cantA=cantA+1;
        end
        if vecinos==K
            break;
        end
    end
    for i=7:9 %Verde
        dist=0;
        for j=1:length(baseDatos(:,1)) %8000
            dist=dist+(cepstrumNuevo(j)-baseDatos(j,i))^2;
        end
        dist=dist^0.5;
        if dist<radio
            vecinos=vecinos+1;
            cantV=cantV+1;
        end
        if vecinos==K
            break;
        end
    end
end

radio=radio+0.03; %Aumento el radio
if radio>3
    break;
end
end
```



```
%resultados
if(cantR > cantA) && (cantR > cantV)
    disp('Dijo ROJO')
    writeDigitalPin(ard,9,1);
    pause(2);
    writeDigitalPin(ard,9,0);
end
if(cantA > cantR) && (cantA > cantV)
    disp('Dijo AMARILLO')
    writeDigitalPin(ard,10,1);
    pause(2);
    writeDigitalPin(ard,10,0);
end
if(cantV > cantA) && (cantV > cantR)
    disp('Dijo VERDE')
    writeDigitalPin(ard,11,1);
    pause(2);
    writeDigitalPin(ard,11,0);
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% REPORTE DE RESULTADOS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
prompt='Es correcto? (S)i o (N)o ';
verificacion=input(prompt,'s');
if verificacion=='S'
    if algoritmo=='1'
        KmeansBien=KmeansBien+1;
    else
        KnnBien=KnnBien+1;
    end
else
    if algoritmo=='1'
        KmeansMal=KmeansMal+1;
    else
        KnnMal=KnnMal+1;
    end
end
end

fin='¿Desea Terminar el programa? (T)erminar ';
terminar=input(fin,'s');
if terminar=='T'
    break
end

prompt='¿Desea volver a grabar el color a probar? (S)i o (N)o ';
str=input(prompt,'s');
if str=='S'
    prompt='Presione la tecla "ENTER" cuando esté listo para grabar:
';
    str = input(prompt, 's');
    recordblocking(recObj,1);
    disp('Fin de la grabación')
    myRecording = getaudiodata(recObj);
```

```
        cepstrumNuevo=ifft(log(abs(fft(myRecording))));  
        baseDatos(:,10)=cepstrumNuevo;  
    end  
  
end  
  
disp('REPORTE DE RESULTADOS')  
KmeansTotal=KmeansBien+KmeansMal;  
fprintf('\nK-means corrió %d veces, acertando %d veces y errando %d  
veces',KmeansTotal,KmeansBien,KmeansMal)  
KnnTotal=KnnMal+KnnBien;  
fprintf('\nK-nn corrió %d veces, acertando %d veces y errando %d  
veces',KnnBien+KnnMal,KnnBien,KnnMal)  
fprintf('\n')  
erroresKmeans=KmeansMal/(KmeansMal+KmeansBien);  
erroresKnn=KnnMal/(KnnMal+KnnBien);  
if erroresKmeans > erroresKnn  
    disp('Knn fue más efectivo')  
elseif erroresKnn > erroresKmeans  
    disp('Kmeans fue más efectivo')  
else  
    disp('Ambos algoritmos fueron igualmente efectivos')  
end
```