

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



**Facultad de Ingeniería en Electricidad y Computación**

**“DETECCION DE ANOMALIAS SONORAS APLICADO A LA  
SEGURIDAD EN AMBIENTES CERRADOS USANDO MATLAB”**

**TESINA DE SEMINARIO**

Previa a la obtención del Título de:

**INGENIERO EN ELECTRONICA Y TELECOMUNICACIONES**

Presentado por

**MIGUEL DELGADO DELGADO**

**DANIEL CORDOVA PONCE**

Guayaquil - Ecuador

2011

## AGRADECIMIENTO

A Dios, por estar conmigo en cada paso que doy; a mi familia por su apoyo y amor incondicional, por darme la fortaleza necesaria para seguir adelante; a mis amigos, por todo el ánimo y paciencia, y sobre todo por su valiosa amistad y apoyo que me brindaron en todo momento.

**Miguel Delgado Delgado**

A Jehová Dios por la dicha de haberme bendecido con Padres y Hermanos que en todo momento de mi vida estuvieron siempre a mi lado dándome ánimo y apoyo incondicional, así como a mis maestros quienes supieron mostrarme todo su talento sin reservas.

**Daniel Córdova Ponce**

## DEDICATORIAS

Dedico este trabajo principalmente a Dios. A mis padres, por ser quienes me apoyaron durante mi carrera universitaria. A todos mis amigos y personas queridas a los cuales guardaré siempre mi eterna gratitud.

**Miguel Delgado Delgado**

A Jehová Dios, a mi Papa, a mi Mama y a mis Hermanos quienes con sus sabios consejos y amor hicieron de mí la persona que hoy soy. A mis maestros y amigos incondicionales, quienes también ayudaron en gran forma a mi formación profesional

**Daniel Córdova Ponce**

## DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este Trabajo de Graduación, nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la

**Escuela Superior Politécnica del Litoral”**

(Reglamento de Graduación de la ESPOL)

---

Miguel A. Delgado Delgado

---

Daniel Córdova Ponce

# TRIBUNAL DE SUSTENTACION

---

Msc. Patricia Chávez Burbano  
**PROFESOR DEL SEMINARIO DE GRADUACIÓN**

---

Ing. Juan Carlos Avilés Castillo  
**PROFESOR DELEGADO POR EL DECANO DE LA FACULTAD**

## RESUMEN

En el presente proyecto se va a desarrollar un sistema de detección de anomalías sonoras en ambientes cerrados optimizando recursos tales como procesos de adquisición de datos y algoritmos eficientes. A continuación se detalla lo que se va a realizar en cada capítulo del presente reporte.

En el **primer capítulo** se realiza un análisis del problema, con el cual se especifica todos los objetivos ya sean específicos y generales, y la razón del por qué hemos realizado nuestro proyecto.

En el **segundo capítulo** se analizan todas las herramientas necesarias para tener la capacidad de proponer una solución válida que tenga bases teóricas, se hará referencia a los conocimientos obtenidos durante el proceso de aprendizaje y que de alguna manera ayuden a entender un poco mejor la problemática.

En el **tercer capítulo** veremos la implementación de la solución con las herramientas escogidas y las consideraciones necesarias para poder llevar a cabo el desarrollo de nuestro problema.

Por último, en el **cuarto capítulo**, realizaremos las pruebas de comprobación de la solución diseñada, para corroborar que cumpla con las expectativas propuestas al inicio de nuestro proyecto, y con todos los objetivos planteados.

## INDICE GENERAL

<b>AGRADECIMIENTO.....</b>	<b>II</b>
<b>DEDICATORIAS .....</b>	<b>III</b>
<b>DECLARACIÓN EXPRESA .....</b>	<b>IV</b>
<b>TRIBUNAL DE SUSTENTACION .....</b>	<b>V</b>
<b>RESUMEN.....</b>	<b>VI</b>
<b>INDICE GENERAL .....</b>	<b>VII</b>
<b>INDICE DE FIGURAS .....</b>	<b>X</b>
<b>INDICE DE TABLAS .....</b>	<b>XI</b>
<b>INTRODUCCION.....</b>	<b>XII</b>
<b>CAPITULO 1 .....</b>	<b>1</b>
1. ANALISIS CONTEXTUAL.....	1
1.1. ANTECEDENTES .....	1
1.2. OBJETIVO GENERAL .....	2
1.3. OBJETIVOS ESPECIFICOS .....	3
<b>CAPITULO 2 .....</b>	<b>4</b>
2. TEORÍA DEL SONIDO.....	4
2.1. ONDAS SONORAS .....	4
2.2. MAGNITUDES FÍSICAS DEL SONIDO.....	5
2.2.1. LONGITUD DE ONDA .....	6
2.2.2. FRECUENCIA DE LA ONDA .....	7

2.2.3. PERIODO DE LA ONDA .....	8
2.2.4. AMPLITUD DE LA ONDA .....	8
2.2.5. AMPLITUD ACÚSTICA .....	9
<b>CAPITULO 3 .....</b>	<b>10</b>
<b>3. IMPLEMENTACION .....</b>	<b>10</b>
3.1. REQUERIMIENTOS DEL HARDWARE .....	10
3.2. DIAGRAMA DE FLUJO .....	11
3.3. DESCRIPCIÓN DEL DIAGRAMA DE FLUJO.....	13
3.3.1. DIAGRAMA DE FLUJO PARTE A.....	13
3.3.2. DIAGRAMA DE FLUJO PARTE B.....	15
3.4. IMPLEMENTACIÓN .....	18
3.4.1. CUADRO DE CONTROL.....	19
3.4.2. CUADRO DE MENSAJES .....	20
3.4.3. PANEL .....	21
3.4.4. CUADRO DE REPORTE.....	21
<b>CAPITULO 4 .....</b>	<b>22</b>
<b>4. PRUEBAS Y RESULTADOS .....</b>	<b>22</b>
4.1. PRUEBAS.....	23
4.1.1. SENSIBILIDAD 25% A 0.5 METROS.....	24
4.1.2. SENSIBILIDAD 25% A 1.5 METROS.....	25
4.1.3. SENSIBILIDAD 25% A 3 METROS.....	25
4.1.4. SENSIBILIDAD 50% A 0.5 METROS.....	26
4.1.5. SENSIBILIDAD 50% A 1.5 METROS.....	26
4.1.6. SENSIBILIDAD 50% A 3 METROS.....	26
4.1.7. SENSIBILIDAD 75% A 0.5 METROS.....	27
4.1.8. SENSIBILIDAD 75% A 1.5 METROS.....	27



4.1.9. SENSIBILIDAD 75% A 3 METROS.....	27
4.2. RESUMEN DE PRUEBAS .....	28
<b>CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>30</b>

## INDICE DE FIGURAS

FIGURA 1 ESQUEMA DE ENTRADA Y SALIDA .....	1
FIGURA 2 ESQUEMA FUTURO DE ENTRADA Y SALIDA.....	2
FIGURA 3 ESPECTRO AUDIBLE DE LAS ONDAS SONORAS .....	5
FIGURA 4 LONGITUD DE ONDA ( $\lambda$ ) .....	6
FIGURA 5 FRECUENCIA DE ONDA (F).....	7
FIGURA 6 PERIODO Y AMPLITUD DE ONDA (T) .....	8
FIGURA 7 DIAGRAMA DE FLUJO PARTE A .....	11
FIGURA 8 DIAGRAMA DE FLUJO PARTE B .....	12
FIGURA 9 INTERFAZ GRAFICA (GUI).....	19
FIGURA 10 ESTRUCTURA IDEAL DEL LUGAR A MONITOREAR.....	22

## INDICE DE TABLAS

TABLA 1 REQUERIMIENTOS DEL SISTEMA PARA WINDOWS.....	10
TABLA 2 PRUEBAS REALIZADAS .....	28

## INTRODUCCION

El presente proyecto trata de optimizar un método en la seguridad en ambientes cerrados, ya los muy comúnmente conocidos tales como video cámaras de seguridad, detectores de movimientos, hasta sensores infrarrojos han intentado dar completa satisfacción, pero incluso con estos sistemas sofisticados no se lo ha conseguido cabalmente.

Es por este motivo que hemos dado atención a este proyecto, detección de anomalías por medio del sonido, ya que muy difícilmente se puede atenuar, y es perceptible en todo el ambiente cerrado, lo que nos proporciona una muy buena información de detección.

Se desarrolló este método usando la plataforma MATLAB, en la cual usando métodos comparativos de señales de audio, entre una señal de audio en “tiempo real” y una señal “base” preestablecida, lograremos detectar las anomalías existentes. Además el sistema tendrá la capacidad de generar reportes y hacer una copia del audio, de la anomalía registrada.

# CAPITULO 1

## 1. ANALISIS CONTEXTUAL

### 1.1. ANTECEDENTES

La alarma por detección de audio se puede usar como complemento a la detección de movimiento de vídeo, ya que puede reaccionar a eventos que se produzcan en áreas demasiado oscuras en las que la función de detección de movimiento pueda no funcionar correctamente. También se puede utilizar para detectar actividad en las áreas que quedan fuera de la visión de la cámara.



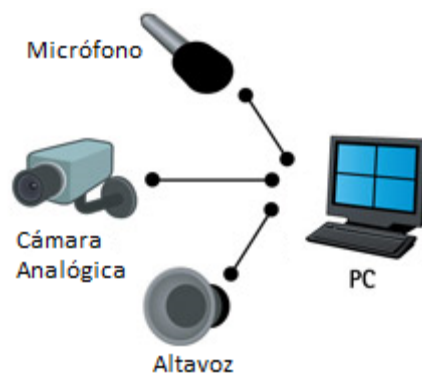
**FIGURA 1 Esquema de Entrada y Salida**

A pesar de que el uso de audio en sistemas de vigilancia todavía no se ha extendido, el hecho es que puede mejorar la capacidad de un sistema para detectar e interpretar eventos.

El audio integrado en un sistema de video vigilancia puede suponer una gran ventaja para un sistema a la hora de detectar e interpretar eventos y situaciones de emergencia. La capacidad del audio de cubrir un área de 360 grados permite que el sistema de

video vigilancia amplíe su cobertura más allá del campo de visión de la cámara.

En nuestro proyecto podremos implementar en un futuro un sistema completo de vigilancia en cual se monitoree todo lo que un simple sistema de video vigilancia no lo podía hacer, ya que con la implementación de nuestro proyecto de vigilancia por audio se cubren los huecos que las cámaras no pueden detectar como se había mencionado anteriormente.



**FIGURA 2 Esquema Futuro de entrada y salida**

En el desarrollo de este sistema de seguridad, hemos implementado uno en el cual se supla, las carencias de otros sistemas, conocidos como puntos ciegos, por eso nos hemos trazado los siguientes objetivos:

## **1.2. OBJETIVO GENERAL**

- Desarrolla un sistema de detección de anomalías sonoras en ambientes cerrados optimizando los recursos empleados.

### **1.3. OBJETIVOS ESPECIFICOS**

- Usar el software Matlab y su interfaz gráfica para la implementación del sistema
- Investigar cómo se realiza la adquisición de audio usando el software Matlab
- Analizar diferentes métodos de comparación de señales de audio y seleccionar el más adecuado para nuestro sistema
- Realizar suficientes pruebas de sensibilidad y veracidad para nuestro sistema

# CAPITULO 2

## ANALISIS DE LAS HERRAMIENTAS Y CONOCIMIENTOS DISPONIBLES

### 2. TEORÍA DEL SONIDO

El sonido es producido por las vibraciones de un cuerpo en un medio elástico, como el aire o el agua; y estas se propagan en forma de **ondas**, estas ondas son las que van hasta nuestro oído, y es el que se encarga de transferirla a nuestro cerebro.

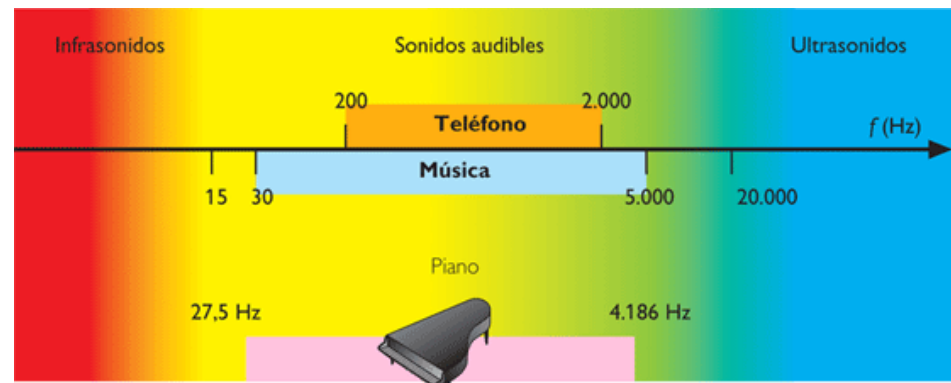
#### 2.1. ONDAS SONORAS

Estas ondas son de tipo mecánica, estas vibraciones se producen por el desplazamiento de las moléculas del aire debido a la acción de una presión externa. Cada molécula transmite la vibración a la que está a su lado provocándose un movimiento en cadena. Así pues, el sonido es el resultado de estos desplazamientos.

Es de notar que el oído humano tiene limitaciones para percibir las ondas sonoras, es decir solo puede percibir cierto rango de frecuencias, esta va desde los 20Hz hasta los 20KHz, también conocido como espectro audible.



Cabe destacar que ondas sonoras inferiores a 20Hz se las denomina infrasonido y las superiores a 20KHz se las denomina ultrasonido.



**FIGURA 3 Espectro Audible de las Ondas Sonoras**

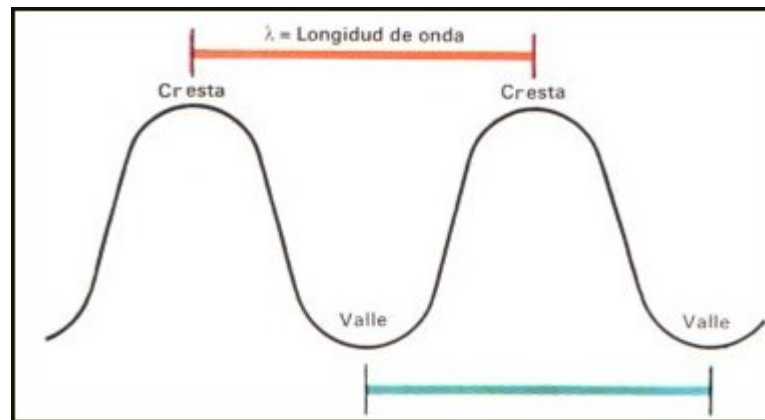
## 2.2. MAGNITUDES FÍSICAS DEL SONIDO

Las ondas sonoras pueden representarse como la suma de armónicos sinusoidales, de tal forma, que tienen las mismas propiedades de estas, por tanto las ondas sonoras tienen:

- ❖ Longitud de onda
- ❖ Frecuencia
- ❖ Periodo
- ❖ Amplitud

[Figura 3] Tomado de imágenes de [www.google.com](http://www.google.com), "espectro de las ondas sonoras"

### 2.2.1. LONGITUD DE ONDA



**FIGURA 4** Longitud de onda ( $\lambda$ )

Se define así a la distancia que recorre la onda entre dos máximos consecutivos de la onda, es representado con la letra griega ( $\lambda$ ) en las ondas sonoras estas pueden variar desde 2 cm hasta los 17 mts.

La longitud de onda es inversamente proporcional a la frecuencia de la onda. Su representación matemática es

$$\lambda = \frac{v}{f}$$

Donde:

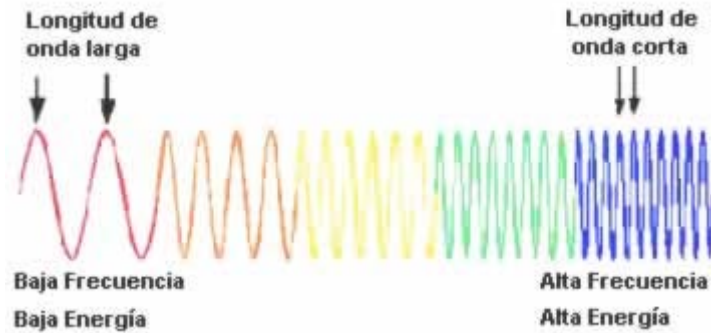
$\lambda$ : Longitud de onda

$v$ : Velocidad de la onda

$f$ : Frecuencia de la onda

[Figura 4] Tomado de imágenes de [www.google.com](http://www.google.com), "longitud de onda"

### 2.2.2. FRECUENCIA DE LA ONDA



**FIGURA 5** Frecuencia de onda ( $f$ )

Esta mide el número de repeticiones de algún evento en una unidad de tiempo, en el caso de las ondas sonoras mide el número de veces que se repite la onda en un segundo, se la representa por la letra (**f**) y su unidad de medida es el Hertzio (**Hz**).

La frecuencia es el inverso del Periodo de la onda (**T**) y su representación matemática es la siguiente:

$$f = \frac{v}{\lambda}$$

Donde:

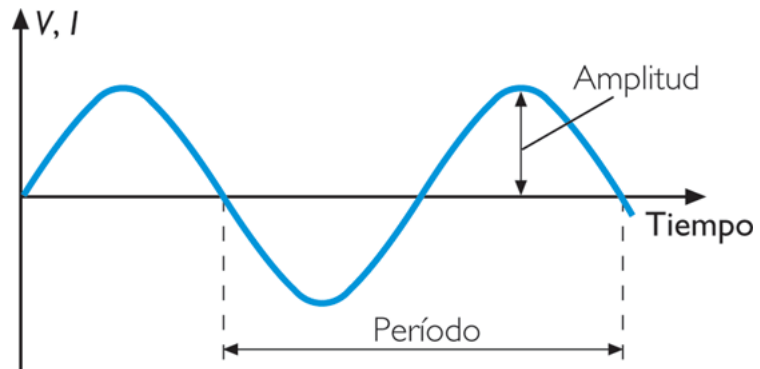
$f$ : Frecuencia de la onda

$v$ : Velocidad de la onda

$\lambda$ : Longitud de onda

[Figura 5] Tomado de imágenes de [www.google.com](http://www.google.com), "frecuencia de onda"

### 2.2.3. PERIODO DE LA ONDA



**FIGURA 6** Periodo y Amplitud de onda ( $T$ )

Se lo define como el tiempo transcurrido entre os puntos equivalentes de una onda. Es representado por la letra (**T**), y es medido en unidades de tiempo, segundos (**s**).

Se lo representa como el inverso al periodo esto es:

$$T = \frac{1}{f}$$

Donde:

$T$ : Periodo de la onda.

$f$ : Frecuencia de la onda.

### 2.2.4. AMPLITUD DE LA ONDA

La amplitud de una onda se refiere a la variación máxima de desplazamiento, es decir a la distancia desde un punto intermedio hasta el nivel más alto de la onda.

[Figura 6] Tomado de imágenes de [www.google.com](http://www.google.com), "periodo de onda"

Aquí es donde nos diferenciamos con ondas acústicas y entramos a definir, **Amplitud acústica**.

### 2.2.5. AMPLITUD ACÚSTICA

La percepción del sonido no es lineal sino es de tipo logarítmica, lo que hace necesario un método diferente de expresión del mismo.

La potencia acústica se la mide en decibelios, (un submúltiplo del Belio), estas representan la relación entre dos señales y se basa en un logaritmo de base 10 del cociente entre dos amplitudes sonoras o presiones.

Y esta es medida de la siguiente forma:

$$L_p = 20 \log_{10} \frac{P_1}{P_0} (dB)$$

Donde:

$L_p$ : Potencia de la onda (dB)

$P_1$ : Potencia del sonido a estudiar (Pa)

$P_0$ : Potencia de referencia  $2 \times 10^{-5}$  (Pa)

## CAPITULO 3

### 3. IMPLEMENTACION

#### 3.1. REQUERIMIENTOS DEL HARDWARE

A continuación se detallan los requerimientos mínimos que necesita la plataforma de Matlab para una buena implementación, usando el sistema operativo de Windows:

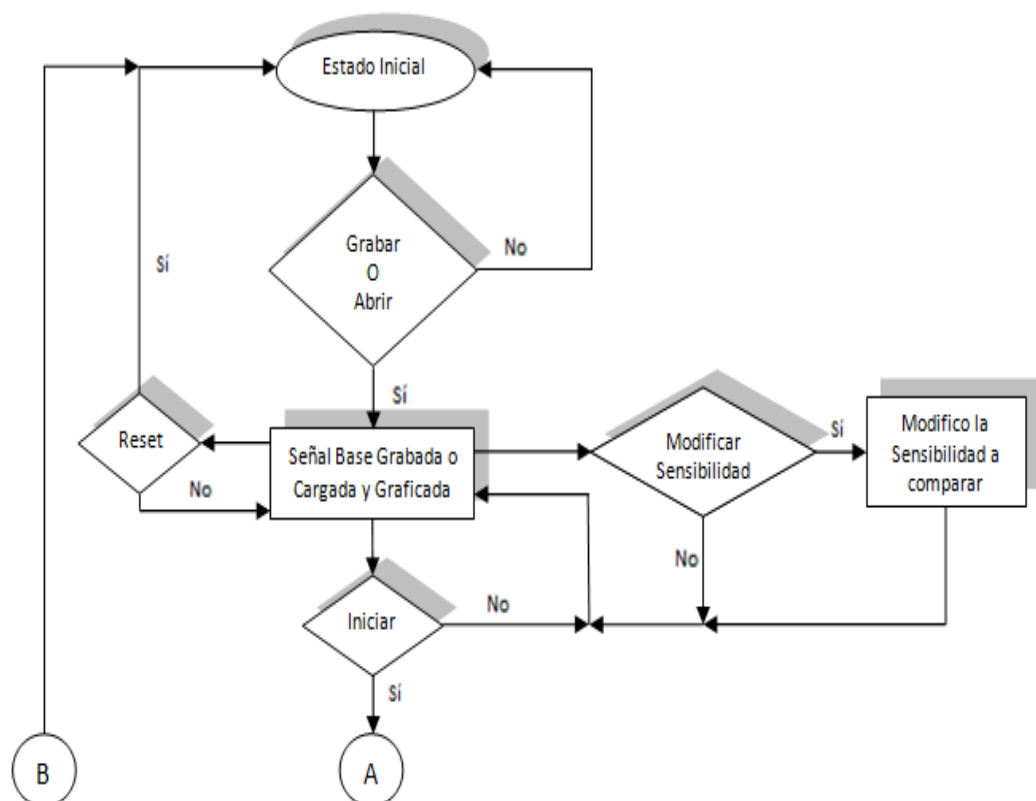
<b>Sistema Operativo</b>	<b>Procesador</b>	<b>Espacio en el disco</b>	<b>RAM</b>
32-Bit y 64-Bit MATLAB y Simulink Product Families			
Windows XP Service Pack 3	Cualquier procesador Intel o AMD x86	1 GB solo para MATLAB, 3–4 GB para instalación normal	1024 MB (2048 MB recomendado)
Windows XP x64 Edition Service Pack 2			
Windows Server 2003 R2 Service Pack 2			
Windows Vista Service Pack 1 or 2			
Windows Server 2008 Service Pack 2 or R2			
Windows 7			

**TABLA 1** Requerimientos del Sistema para Windows

Las pruebas realizadas fueron en una PC Xtratech son sistema operativo Windows 7 de 32 bits (6.1, compilación 7600), con 2048MB de memoria RAM, y se uso un micrófono Cisco para la adquisición de datos.

### 3.2. DIAGRAMA DE FLUJO

Estos serán los pasos a seguir para la ejecución del programa.



**FIGURA 7 Diagrama de Flujo parte A**

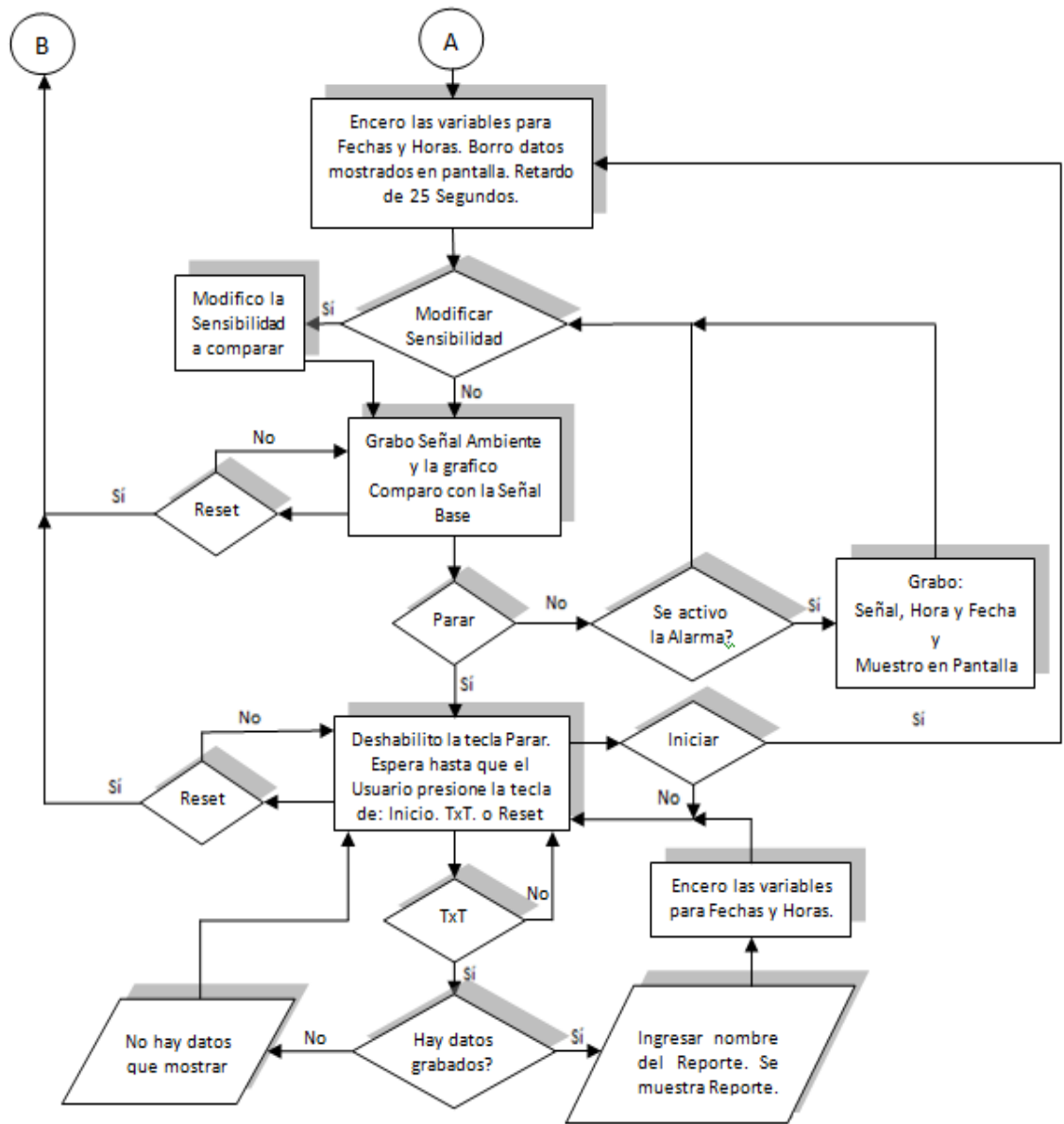


FIGURA 8 Diagrama de Flujo parte B



### **3.3. DESCRIPCIÓN DEL DIAGRAMA DE FLUJO**

Nuestro diagrama de flujo lo hemos dividido en dos partes como se muestran en las figuras 7 y 8 que corresponden al diagrama de flujo parte A y diagrama de flujo parte B respectivamente, por tal motivo se va a explicar en forma separada cada uno de ellos para brindar mayor claridad y comprensión al usuario.

#### **3.3.1. DIAGRAMA DE FLUJO PARTE A**

Nuestro sistema empieza con su estado inicial después de haber puesto a ejecutar el programa.

En este estado el sistema esperará hasta que el usuario presione las teclas de Grabar o Abrir, ya que se necesita obligatoriamente una Señal del Ambiente que se va a monitorear, y ésta se la conocerá como Señal Base; si el Usuario no presiona ninguna de las dos teclas el sistema no hará nada y se mantendrá en el Estado Inicial.

Cuando se haya presionado la tecla de Iniciar, se procederá a ir al siguiente estado que lo denominaremos como un estado “Estacionario”, ya que simplemente se mostrará gráficamente la Señal Base en el dominio del tiempo. Como observamos en este estado el sistema esperará hasta que el usuario presione cualquier tecla que

se encuentre habilitada, estas serán específicamente las teclas de “Reset”, “Iniciar”, ya que las teclas de “Grabar” y “Abrir” se deshabilitan automáticamente después de presionarlas. También se puede cambiar la sensibilidad a utilizar con la barra deslizante (conocida como “Slider” en la interfaz gráfica de Matlab).

En el caso de presionar “Reset” en cualquiera que sea de los casos posteriores y en este el sistema interrumpirá cualquier proceso que se esté ejecutando y borrará todos los datos almacenados y luego retornará al Estado Inicial del programa.

En el caso de modificar la Sensibilidad, el sistema cogerá el valor máximo a comparar de la Señal Base y lo multiplicará por un coeficiente, éste resultado me va a representar el umbral en el que las Señales de audio deberán sobrepasar para activa la alarma. Por ejemplo, si el valor máximo de la señal base es 0.9, y escogemos una sensibilidad de 10%, el sistema automáticamente multiplicará  $0.9 * 1.1$  y obtenemos como resultado 0.99, que tendrá 10% más que el valor máximo de la señal base antes mencionado, y este se comparará con las Señales adquiridas. EL valor de 1.1 lo obtengo de la barra

deslizante, haciendo que los valores de ésta sean entre 1 y 2, y varíen de acuerdo a la posición que el usuario la ubique, superior como máxima “2” e inferior como mínima “1”. Por defecto la barra deslizante está en la mitad (“coeficiente 1.5”) que me va a representar el 50% de la Sensibilidad

Si presionamos la tecla “Iniciar” el sistema comenzará hacer lo más importante, esto se lo explicará en el diagrama de flujo Parte B.

### **3.3.2. DIAGRAMA DE FLUJO PARTE B**

Luego de haber presionado Iniciar el Sistema procederá a encerrar todas las variables a utilizar, las más importantes son las de tipo estructuras que se las usarán para almacenar las horas y fechas de los sucesos para presentarlos en los reportes que se van a generar. Luego antes de empezar el proceso se aplicará un retardo de veinticinco segundos exactamente para que el usuario tenga el suficiente tiempo para salir de la oficina o cuarto a monitorear y no lo detecte el sistema en esa transición de salida.

Luego el sistema estará constantemente evaluando la sensibilidad que nosotros apliquemos usando la barra deslizante, si no la presionamos simplemente no pasa nada y continua con el siguiente bloque como se muestra en la Figura 8.

El siguiente Bloque es el más importante, y en sí es lo que hará que funcione la alarma y sea efectiva; en este se aplicarán los códigos de adquisición de audio para adquirir una señal de tres segundos exactos y luego compararla con el valor de umbral de la Señal Base que se genera al modificar la posición de la barra deslizante y luego se graficará la señal recién grabada en el dominio del tiempo. Aquí se va a estar evaluando constantemente los estados de las teclas de "Reset" y de "Parar" ya que cuando nosotros queramos las podemos presionar, y automáticamente interrumpirá cualquier proceso que se esté realizando.

Si no se presiona ninguna de estas dos teclas el sistema estará evaluando si se activó la alarma o no; en el caso de que no se active el sistema grabará una nueva señal de tres segundos y comparará de nuevo con la señal base,

esto ocurrirá hasta que el usuario interrumpa esta operación por medio de las teclas “Parar” o “Reset”.

Si se activo la alarma, el sistema grabará la fecha y la hora en la que ocurrió este suceso. Adicionalmente grabará el sonido de tres segundos antes adquirido y que ocasionó dicha activación en una carpeta dentro de la que se está trabajando.

Si el proceso anterior se interrumpe por medio de la tecla “Parar”, el sistema pasará a otro bloque estacionario en el que se deshabilitará la tecla de “Parar” y se esperara hasta que el usuario presione cualquiera de las teclas habilitadas que son “Inicio”, “TxT”, o “Reset”.

Ya anteriormente se explicó lo que pasa si presionamos las teclas de “Inicio” y ”Reset”, así que si presionamos la tecla de TxT lo primero que se evaluara es si es que hay datos grabados, es decir, en otras palabras si es que se activo la alarma, ya que estos datos corresponden a los de la fecha y hora que se mostraran en el Reporte.

Si no existen datos el sistema me mostrara un mensaje indicando que no existen datos para mostrar, caso contrario el sistema me mostrara una ventana para

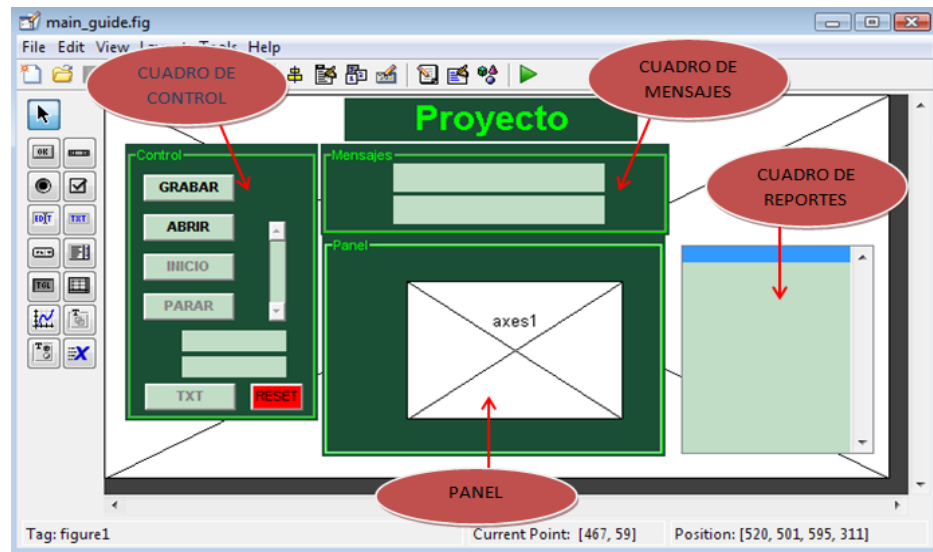
ingresar el nombre que creamos conveniente del reporte con todas las anomalías si es que estas fueron varias y no solo una.

Luego de haber presionado TxT y de realizar el proceso anterior, se regresará al último bloque estacionario y el sistema esperará si se desea “Resetear” el sistema o “Iniciar” nuevamente el proceso de comparación usando la misma señal base que se obtuvo al principio.

### **3.4. IMPLEMENTACIÓN**

La implementación del detector de anomalías sonoras se la ha desarrollado en la plataforma de Matlab, por lo tanto, seguida explicando el desarrollo de la misma a través de las líneas de comandos que se usaron, en este trabajo.

Una de las primeras cosas que vamos a pasar explicar, en la elaboración de la interfaz gráfica, conocida como GUI de Matlab, para lo cual presentamos la siguiente figura.



**FIGURA 9 Interfaz Grafica (GUI)**

Como se puede observar, tiene cuatro cuadros:

- ❖ Cuadro de control
- ❖ Cuadro de mensajes
- ❖ Panel
- ❖ Cuadro de reportes

### 3.4.1. CUADRO DE CONTROL

En el cuadro de control, observamos cinco botones, dos cuadros de mensajes, y una barra deslizante; lo cual pasaremos a describir.

**Grabar**: con este botón el programa nos da la oportunidad de poder grabar la señal base.

**Abrir**: con este botón de programa nos permite abrir un archivo de audio “.wav” pregrabado, para poder usarlo como base.

**Inicio**: con este botón se comienza la ejecución del programa.

**Parar**: con este botón, se detiene la ejecución del programa.

**TXT**: con este botón, el sistema permite crear un archivo en formato “.TXT”, con los informes, de las anomalías detectadas por el sistema.

**RESET**: este botón permite, regresar el sistema condiciones iniciales, el uso del mismo se lo especificará en el manual de usuario.

#### **3.4.2. CUADRO DE MENSAJES**

En estas dos líneas, se presentará, el estado en el que se encuentre el sistema, es decir, que mostrará cuadros de diálogos como por ejemplo: “grabando”, “comparando”, así como una cuenta regresiva, que indicara un tiempo de retardo antes de iniciar el funcionamiento del sistema.



### **3.4.3. PANEL**

En el panel se mostrará, la grafica en el tiempo, de la señal que esté procesando el sistema en ese momento, esto será: la señal que se ha grabado ó la señal pregrabada que se ha cargado y la señal ambiente adquirida.

### **3.4.4. CUADRO DE REPORTE**

En el cuadro de mensajes, se hará un listado en forma cronológica, de las diferentes anomalías detectadas por el sistema, detallando la hora y fecha cuando ocurra el suceso.

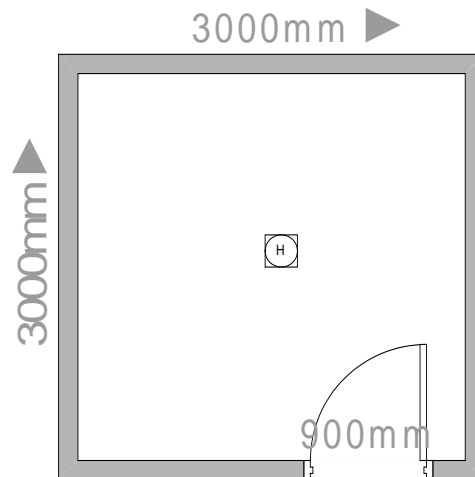
## CAPITULO 4

### 4. PRUEBAS Y RESULTADOS

En este capítulo, pondremos a prueba nuestro sistema antes descrito, pero antes puntualizaremos algunas condiciones.

Primero, como se trata de un ambiente cerrado (es decir una habitación) esta será de dimensión corta, es decir asumiremos una habitación de 3x3 metros; también se asumirá que el dispositivo (micrófono) estará ubicado en una posición central de la habitación, para poder abarcar el sonido de forma simétrica en toda la habitación.

Segundo, se harán ajustes en la sensibilidad del sistema estos serán al 25, 50 y 75 % respectivamente.



**FIGURA 10 Estructura Ideal del lugar a monitorear**

#### 4.1. PRUEBAS

Para comprobar que nuestro sistema funciona de una manera eficiente nosotros hemos hecho varias pruebas, con diferentes tipos de sonidos que podrían darse en cualquier circunstancia, con el fin de brindar mayor confianza, y seguridad al usuario que vaya a usar nuestro sistema de seguridad.

Las pruebas parecen sencillas pero se las han realizado varias veces, para corroborar todos nuestros objetivos

La primera prueba que se hizo es dar un golpe con un esferográfico sobre cualquier escritorio que se encuentre en el lugar que se monitorea.

Las siguientes dos pruebas son que alguna persona este caminando sin hacer nada de ruido, es decir, silenciosamente, y la otra prueba, que esa misma persona, mientras camine y por algún descuido, haga una pisada enérgica sin querer, en otras palabras una pisada enérgica hecha involuntariamente ocasionada por algún tropiezo.

Las dos últimas pruebas que se realizaron fueron simplemente golpear un cuaderno cualquiera o libro, y dejar caer una lata metálica ya sea esta un envase de cualquier bebida.

Cabe recalcar que cada una de las pruebas se las realizaron varias veces para comprobar que nuestro sistema funcionaba de manera correcta.

#### **4.1.1. SENSIBILIDAD 25% A 0.5 METROS.**

En esta primera prueba se golpea con una esferográfica el escritorio donde está trabajando el equipo, y el sistema se activó con el sonido; continuamos dando pasos, como cuando uno camina normalmente, pero el sistema no se activó; seguimos con pisadas enérgicas, es decir, asentar el pie con fuerza sobre el piso (simulando saltos o caídas), el sistema se activó; se golpeó el cuaderno de notas con la misma esferográfica, y el sistema se activa; como última prueba la caída de un objeto al suelo, (usamos una lata metálica), lo que produce un gran estruendo y en efecto nuestro sistema se activó

#### **4.1.2. SENSIBILIDAD 25% A 1.5 METROS.**

En esta segunda fase de pruebas, las pruebas a cumplir son las mismas que en la primera, y serán las mismas en todas las pruebas siguientes, con la excepción de la primera, que no es aplicable, debido a que el equipo se encuentra instalado en un escritorio y la distancia siempre será de 0.5 m siempre. De esta manera: El caminar normal no es detectado por el sistema; al realizar las pruebas de pisadas enérgicas el sistema reacciona favorablemente; el golpear el cuaderno de notas con el esferográfico, también es detectado por el sistema; y la prueba de la caída de la lata metálica al suelo, también el sistema reacciona favorablemente.

#### **4.1.3. SENSIBILIDAD 25% A 3 METROS.**

En esta tercera fase de pruebas, nuevamente el caminar normal no es detectado; aunque si detecta las pisadas enérgicas, así como el golpe al cuaderno de notas y la caída de la lata metálica.

#### **4.1.4. SENSIBILIDAD 50% A 0.5 METROS.**

En la cuarta fase, ahora variamos la sensibilidad del sistema a un 50%; y en esta primera parte, con la nueva sensibilidad, podemos retomar el primer experimento, de golpear el escritorio con una esferográfica, dándonos un resultado positivo, es decir, el sistema se activa; el caminar normal el sistema no lo detecta, en el caso de las pisadas enérgicas si es detectado por el sistema y también se detecta el golpe al cuaderno de notas y la caída de la lata metálica al suelo.

#### **4.1.5. SENSIBILIDAD 50% A 1.5 METROS.**

Quinta fase de pruebas; no detecta los pasos normales; si detecta las pisadas enérgicas; detecta el golpe al cuaderno de notas, así como la caída de la lata metálica

#### **4.1.6. SENSIBILIDAD 50% A 3 METROS.**

Sexta fase de pruebas; no detecta los pasos normales; detecta las pisadas enérgicas; detecta el golpe al cuaderno de notas; y detectó la caída de la lata metálica

#### **4.1.7. SENSIBILIDAD 75% A 0.5 METROS.**

Séptima fase de pruebas; aquí tomamos nuestra última prueba de sensibilidad, es decir al 75%, y como es la primera muestra retomamos el experimento de golpear el escritorio con un esferográfico, y el sistema se activa; mas no al caminar. El sistema se activa con las pisadas fuertes, con el golpe al cuaderno de notas y con la caída del objeto metálico.

#### **4.1.8. SENSIBILIDAD 75% A 1.5 METROS.**

Octava fase de pruebas; no detecta el caminar, mas si lo hace con las pisadas enérgicas, el golpe al cuaderno de notas y con la caída de objetos metálicos.

#### **4.1.9. SENSIBILIDAD 75% A 3 METROS.**

Novena y última fase de pruebas; no de detecta la caminata, pero si lo hace con las pisadas enérgicas, el golpe al cuaderno de notas y con la caída de la lata metálica.

## 4.2. RESUMEN DE PRUEBAS

<b>SENSIBILIDAD</b>	25%	50%	75%
<b>DISTANCIA</b>	<b>GOLPE SOBRE EL ESCRITORIO</b>		
0,5 m	SI	SI	SI
1,5 m	X	X	X
3,0 m	X	X	X
<b>CAMINAR NORMAL</b>			
0,5 m	NO	NO	NO
1,5 m	NO	NO	NO
3,0 m	NO	NO	NO
<b>PISADAS ENÉRGICAS</b>			
0,5 m	SI	SI	SI
1,5 m	SI	SI	SI
3,0 m	SI	SI	SI
<b>GOLPE A CUADERNO DE NOTAS</b>			
0,5 m	SI	SI	SI
1,5 m	SI	SI	SI
3,0 m	SI	SI	SI
<b>CAÍDA DE OBJETOS AL SUELO</b>			
0,5 m	SI	SI	SI
1,5 m	SI	SI	SI
3,0 m	SI	SI	SI

**TABLA 2 Pruebas Realizadas**



En la primera prueba en la que se ejecuto un golpe sobre el escritorio únicamente puede ser detectado a distancias hasta los 1.5 metros, y se puede observar que el sistema responde a las sensibilidades descritas.

Podemos comprobar que el sistema no detecta a una persona que camine normal en ninguna de sus sensibilidades.

En cambio si detecta pisadas enérgicas, esto se refiere a golpear el piso con el pie, fue detectado a medio metro un metro y medio, tres metros y en todas las sensibilidades.

Otro experimento que si lo detectó es el golpear el cuaderno de notas con un esferográfico, fue detectado en toda la distancias, y en todas las sensibilidades.

El último experimento realizado y el más importante de todos, fue la caída de un objeto metálico al suelo, dando como resultado, la activación del sistema en todas las sensibilidades y en todas las distancias descritas.

## CONCLUSIONES Y RECOMENDACIONES

Las conclusiones son:

- 1) Como primera conclusión, podemos notar que sí es viable la creación de un sistema de seguridad usando detección de anomalías sonoras, y esto sirve de complemento a muchos sistemas de seguridad que solo usan sensores de calor, infrarrojos, o cámaras que dejan muchos puntos ciegos que nuestro sistema muy fácilmente puede complementarlo.
- 2) También podemos notar, que sí fue factible la detección de las anomalías, tomando en cuenta las amplitudes de las señales adquiridas, ya que se usaron los valores máximos y mínimos de la señal base y señal a comparar respectivamente, y variando los niveles de sensibilidad para una mejor y óptima comparación.
- 3) También es importante, la grafica de la señal, aún más en la señal de la base, debido a que este gráfico nos permite ver si la señal está bien tomada o no, ya que rápidamente y de forma visual podemos observar si existen picos demasiados pronunciados que hagan que el sistema no funcione como es debido, es decir, de forma óptima.
- 4) Es también importante la generación de los reportes, ya que podemos visualizar las anomalías que ocurrieron, y aún mejor escuchar la anomalía, puesto que se asume que el sistema estará activo cuando no

haya personal dentro de las habitaciones, esto lo hace generando archivos de texto y archivos de Audio con extensión (.wav).

- 5) Asimismo una señal pregrabada, deberá tener similares características a la señal base tomada en ese momento de la edificación o estructura que se vaya a monitorear, para evitar que la alarma se active innecesariamente, por eso es importante observar la grafica, ya que nos ayuda a visualizar si son o no correctas.
- 6) Por último podemos decir que se ha cumplido con todos los objetivos propuestos desde un principio, usando métodos de comparación efectivos, investigando como usar de manera correcta el software de Matlab con su interfaz gráfica, y comprobar las veces necesarias y suficientes que nuestro sistema funcione de manera óptima.

Las recomendaciones son:

- 1) Como primera recomendación es el uso del equipo, que deberá ser, uno que soporte la plataforma de Matlab. En la Tabla 1 del Capítulo 3 se muestran los requerimientos mínimos y necesarios para ejecutar el software de forma correcta.
- 2) El importante usar un micrófono de una buena calidad, que no cause ningún tipo de ruido, o si lo llega a ocasionar que este llegue a ser el mínimo posible, para realizar una comparación óptima entre señales puras de audio.
- 3) Se recomienda, al obtener la señal base, mirar su grafica en el panel, si esta tuviera algún pico, se recomienda volver a grabar la señal de audio, ya que ocasionará que el sistema no funcione de manera correcta y eficiente.
- 4) Se recomienda leer el manual de usuario que se adjunta en los anexos, para que en un principio se sepa lo que se tiene que hacer, desde como abrir el software de Matlab hasta que hacer para generar los reportes y donde quedan guardados para su pronto análisis.
- 5) Con respecto a la sensibilidad, ésta deberá ser tomada de acuerdo a la aplicación que se le quiera dar y también de esta dependerá la efectividad del sistema cómo vemos en los resultados. Por lo tanto en ambientes que haya mucho ruido se deberá escoger una sensibilidad

alta, caso contrario se escogerá una sensibilidad baja, recomendable no menor de un 10%.

- 6) En caso de implementar el sistema, se recomienda ubicar el dispositivo de detección que es el micrófono en una posición central, para que de esta forma, se tomen muestras de audio de forma simétrica y evitar los puntos ciegos auditivos.

# ANEXOS

## Código en Matlab del detector de anomalías sonoras

```
function varargout = main_guide(varargin)
    % Begin initialization code - DO NOT EDIT
    gui_Singleton = 1;
    gui_State = struct('gui_Name',       mfilename, ...
                       'gui_Singleton',  gui_Singleton, ...
                       'gui_OpeningFcn', @main_guide_OpeningFcn, ...
                       'gui_OutputFcn',  @main_guide_OutputFcn, ...
                       'gui_LayoutFcn',  [], ...
                       'gui_Callback',    []);
    if nargin && ischar(varargin{1})
        gui_State.gui_Callback = str2func(varargin{1});
    end

    if nargin
        [varargout{1:nargout}] = gui_mainfcn(gui_State,
        varargin{:});
    else
        gui_mainfcn(gui_State, varargin{:});
    end
    % End initialization code - DO NOT EDIT

    %% --- CONDICIONES INICIALES DEL PROGRAMA
function main_guide_OpeningFcn(hObject, eventdata, handles,
varargin)
    % Leer imagen de fondo y graficar
    image(imread('onda.jpg'),'parent',handles.axes2)
    % Quitar ejes
    set(handles.axes2,'XTick',[],'YTick',[])
    % Centrar GUI
    movegui(hObject,'center')
    % Leer sonido de alarma
    [x,f]=wavread('siren.wav'); %carga el sonido de alarma
    % Almacenar en la estructura la señal y la frecuencia
    handles.player = audioplayer(x, f);
    % Ejes de la gráfica
    set(handles.axes1,'YTick',[0 .5 1],'YTickLabel',[' 0'; ' M'; '
2M'],'Xtick',[])
    % Matriz que almacena el nivel de la señal que está por encima
    del umbral
    handles.senal_up=[ ];
    handles.las_horas=[ ];
    handles.dif=[ ];
```

```

% Fin condiciones iniciales
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);

%% --- Outputs from this function are returned to the command
line.
function varargout = main_guide_OutputFcn(hObject, eventdata,
handles)
    varargout{1} = handles.output;

%% --- GRABAR SONIDO DESDE EL MICRÓFONO
function grabar_b_Callback(hObject, eventdata, handles)
% Desactivar botones
set(handles.abrir_b, 'Enable', 'off')
set(handles.grabar_b, 'Enable', 'off')
set(handles.reset_b, 'Enable', 'off')
pause(0.1)
% Frecuencia de muestreo
Fs=8e3;%8 KHz
% Mensaje que se está grabando
set(handles.lcd, 'String', 'Grabando...')
pause(0.1)
% Grabar 3 segundos de voz
tiempo_de_grab=3;
handles.voz= abs(wavrecord(tiempo_de_grab*Fs,Fs));
% Mensaje que se grabó
set(handles.lcd, 'String', 'OK')
% Graficar la señal
axes(handles.axes1)
handles.Fs=Fs;
% Tiempo
tiempo=0:1/Fs:(length(handles.voz)-1)/Fs;
% Ploteo
plot(tiempo,handles.voz)
handles.mas=max(handles.voz);% ;

set(handles.text5, 'String', handles.mas) %*****
pause(0.1)
%energía de la señal base
handles.energia=sum(handles.voz.^2);
%Muestro el margen en el umb_s
porcentaje=get(handles.sensible, 'Value');
handles.porcentaje=porcentaje;
margen=porcentaje*(handles.mas);
set(handles.umb_s, 'string', margen) %*****
% Activar botones
set(handles.inicio_b, 'Enable', 'on')
%set(handles.parar_b, 'Enable', 'on')
set(handles.sensible, 'Enable', 'on')
set(handles.reset_b, 'Enable', 'on')
% Almacenar datos en la estructura de la GUI

```

```

guidata(hObject, handles);

%% --- ABRIR SEÑAL DE AUDIO .WAV
function abrir_b_Callback(hObject, eventdata, handles)
% Desactivar botones
set(handles.grabar_b, 'Enable', 'off')
set(handles.abrir_b, 'Enable', 'off')
set(handles.reset_b, 'Enable', 'off')
% GUI para seleccionar archivo wav
[name path]=uigetfile('*.wav', 'Leer archivo wav');
if name == 0%No hacer nada si se presiona el botón de
cancelación
    return
end
% Leer archivo wav
[handles.voz, handles.Fs]=wavread(fullfile(path,name));%LOAD MAT
FILE
% Mensaje que se está grabando
set(handles.lcd, 'String', 'Señal Cargada')
pause(0.1)
% Graficar la señal
axes(handles.axes1)
Fs=8e3;%8 KHz
handles.voz=abs(handles.voz);
% Tiempo
tiempo=0:1/Fs:(length(handles.voz)-1)/Fs;
% Ploteo
plot(tiempo,handles.voz)
% Energía de la señal base
handles.mas=max(handles.voz);

set(handles.text5, 'String', handles.mas)%*****
pause(0.1)
%energía de la señal base
handles.energia=sum(handles.voz.^2);
%Muestro el margen en el umb_s
porcentaje=get(handles.sensible, 'Value');
handles.porcentaje=porcentaje;
margen=porcentaje*(handles.mas);
set(handles.umb_s, 'string', margen)%*****
% Activar botones
set(handles.inicio_b, 'Enable', 'on')
%set(handles.parar_b, 'Enable', 'on')
set(handles.sensible, 'Enable', 'on')
set(handles.reset_b, 'Enable', 'on')
% Guardar en la estructura handles
guidata(hObject, handles)

%% --- INICIO DEL PROGRAMA PRINCIPAL
function inicio_b_Callback(hObject, eventdata, handles)

```



```

%borro los datos que se han ppresentado en el ListBox "horas y
fechas"
set(handles.registro,'String',' ')
handles.las_horas=[];
handles.senal_up=[];
handles.dif=[];
% Axes a graficar
axes(handles.axes1)
% Desactivar botones
set(handles.parar_b,'Enable','off')
set(handles.reset_b,'Enable','off')
set(handles.sensible,'Enable','on')
% Valor de parada del ciclo de adquisición
set(handles.parar_b,'UserData',0)
set(handles.reset_b,'UserData',0) %*****
% Energía de la señal ambiente
mas=handles.mas;
% Tiempo de espera hasta iniciar el programa
for n=5:-1:1 %25:-1:1
    % Mostrar contador
    set(handles.lcd,'String',n)
    % Pausa de 1 segundo
    pause(1)
end
%Habilito los botones de parar y reset
set(handles.parar_b,'Enable','on')
set(handles.reset_b,'Enable','on')
% Frecuencia de muestreo
Fs=8e3;%8 KHz
% Tiempo de la señal
tiempo=0:1/Fs:(length(handles.voz)-1)/Fs;
% Ciclo de adquisición
while 1

    % Rutina IF de parada del ciclo while
    if get(handles.reset_b,'UserData'),break,end%*****
    if get(handles.parar_b,'UserData'), break, end
    pause(0.1)
    % Detener sonido de alarma (si está activo)
    stop(handles.player);
    % Tomar valores de sensibilidad del slider
    porcentaje=get(handles.sensible,'Value');
    handles.porcentaje=porcentaje;
    margen=porcentaje*(handles.mas);
    %Muestro el margen en el umb_s
    set(handles.umb_s,'string',margen)%
    % Mensaje indicando que se está grabando
    set(handles.lcd,'String','Grabando ambiente')
    % Pausa para presentar el mensaje
    pause(0.1)
    % Grabar 3 segundo de audio
    Signal1= wavrecord(3*Fs,Fs);
    % Mostrar mensaje que se está comparando los valores con el
    margen

```

```

set(handles.lcd,'String','Comparando...')
pause(0.1)
% Acondicionar la señal
senal=abs(Signal1);
% Energía de la señal de entrada
el_m=max(senal);
handles.ener=sum(senal.^2);
ener=handles.ener;
% Si es 0, no se ha grabado señal (mic desconectado)
if el_m==0, warndlg('Señal en 0','AVISO'), break, end
% Graficar señal adquirida
plot(tiempo,senal)
grid on
% Ejes de la gráfica
set(handles.axes1,'YTick',0:1/2*el_m:el_m)
% Límites de graficación
ylim([0 max(senal)])
% Título: el valor máximo de la señal
title(['Valor Maximo: ',num2str(el_m)])
% Código de comparación
if any(senal>margen) % ¿Hay algún valor en la señal mayor
que el margen?
    % Buscar el valor mayor al umbral
    indice=find(senal>margen);
    % Tomar el primero de estos valores
    de_mas=senal(indice(1));
    % Almacenar este valor
    handles.senal_up=[handles.senal_up; ener];
    % Reproducir sonido
    play(handles.player);
    % Hora de activación de la alarma
    F=datestr(clock);
    L=num2str(F);
    FN=[num2str(F),'.wav'];
    FN1=strrep(FN,':','-');
    handles.las_horas=[handles.las_horas; L];
    set(handles.registro,'String',handles.las_horas)
    % Almacenar sonido que activo la alarma
    wavwrite(Signal1,['audios\',FN1])
    % Mostrar en statix-text la hora del evento
    set(handles.hora,'String','"Anomalia"')
    handles.dif=[handles.dif; de_mas];
    pause(0.5)
end
set(handles.hora,'String',' ')
end
% Almacenar el valor del margen en un "handles"
handles.margen=margen;
% Mensaje de parada de adquisición
set(handles.lcd,'String','Stop')
% Grabar los datos handles
guidata(hObject, handles)

```

```

%% --- FUNCIÓN DE PARADA
function parar_b_Callback(hObject, eventdata, handles)
% Valor de parada del ciclo de adquisición
set(handles.parar_b, 'UserData', 1)
% Detener ejecución del sonido de alarma
pause(0.1)
stop(handles.player);
% Activar botones
set(handles.inicio_b, 'Enable', 'on')
set(handles.txt_b, 'Enable', 'on')
set(handles.parar_b, 'Enable', 'off')
set(handles.sensible, 'Enable', 'off')
pause(0.1)

%% --- FUNCIÓN DEL SLIDE DEL SENSIBILIDAD
function sensible_Callback(hObject, eventdata, handles)
% Tomar valor
v=get(hObject, 'Value');
% Sensibilidad entre 1 y 100
valor=(v-1)*100;
% Escribir umbral
set(handles.umb_s, 'string', handles.mas*v);
% Colocar valor de porcentaje en el slider
set(hObject, 'Tooltipstring', num2str(valor))

%% --- GRABAR DATOS EN UN TXT
function txt_b_Callback(hObject, eventdata, handles)
pause(0.1)
% Verificar si hay señal a graficar
if isempty(handles.senal_up)
    msgbox('No hay datos para mostrar')
    return
end
% Nombre del archivo a guardar eventos
[n r]=uiputfile('*.txt', 'Guardar eventos');
if n~=0, return, end
reporte=fullfile(r,n);
% Guardar el procesamiento en un archivo de texto
fid = fopen(reporte, 'wt');%

fprintf(fid, '%s\n\n', '*****');
fprintf(fid, '%s\n', 'Energia de la Señal Base');
fprintf(fid, '%12.8f\n\n', handles.energia);%

fprintf(fid, '%s\n\n', '*****');
fprintf(fid, '%s', ['Margen del ', num2str(((handles.porcentaje)-
1)*100)]);
fprintf(fid, ' por ciento\n');
fprintf(fid, '%12.8f\n\n', handles.margen);%

```

```

fprintf(fid,'%s\n\n','*****
*****');
    fprintf(fid,'%s\n','Valores > Margen que activaron la alarma');
    for n=1:size(handles.dif,1)
        fprintf(fid,'%12.8f\n',handles.dif(n,:));%
    end
    fprintf(fid,'\n');

fprintf(fid,'%s\n\n','*****
*****');
    fprintf(fid,'    %s \t\t ','Hora y fecha');
    fprintf(fid,'%s\n','Energia de las Señales');
    for n=1:size(handles.las_horas,1)
        fprintf(fid,'%s \t\t\t %f\n',
handles.las_horas(n,:)','handles.senal_up(n));%
    end
    fprintf(fid,'%s\n',handles.las_horas');%
    fclose(fid);
    % Abrir archivo de texto
    winopen(reporte)%Open 'text.txt' file
    %Borro los parametros que se encuentran en las estructuras
    handles.las_horas=[ ];
    handles.senal_up=[ ];
    handles.dif=[ ];
    guidata(hObject,handles)

%% --- Funcion de Reset
function reset_b_Callback(hObject, eventdata, handles)
    set(handles.reset_b,'UserData',1)
    pause(0.1)
    % Detener sonido de alarma (si está activo)
    stop(handles.player);
    %Borro todo lo que se muestra en los Statics text
    set(handles.lcd,'String',' ')
    set(handles.hora,'String',' ')
    set(handles.text5,'String',' ')
    set(handles.umb_s,'String',' ')
    set(handles.registro,'String',' ')
    %Reseteo las variables
    handles.senal_up=[ ];
    handles.las_horas=[ ];
    handles.dif=[ ];
    handles.voz=0;
    handles.mas=0;
    handles.margen=0;
    handles.diferencia=0;
    %Borro el grafico de la ultima señal ploteada
    set(handles.axes1)
    title(' ')
    tiempo=0:1/3:3;
    x=0;
    % Ploteo

```

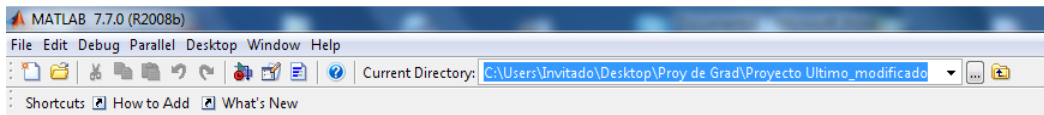
```
plot(tiempo,x)
ylim([0 1])
%Volver al estado inicial
set(handles.grabar_b,'Enable','on')
set(handles.abrir_b,'Enable','on')
set(handles.inicio_b,'Enable','off')
set(handles.parar_b,'Enable','off')
set(handles.txt_b,'Enable','off')

guidata(hObject, handles)
% hObject    handle to reset_b (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

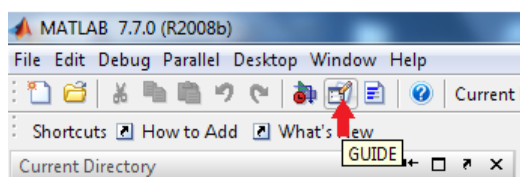
%% --- Funcion del Registro de Reportes
function registro_Callback(hObject, eventdata, handles)
```

# Manual de Usuario

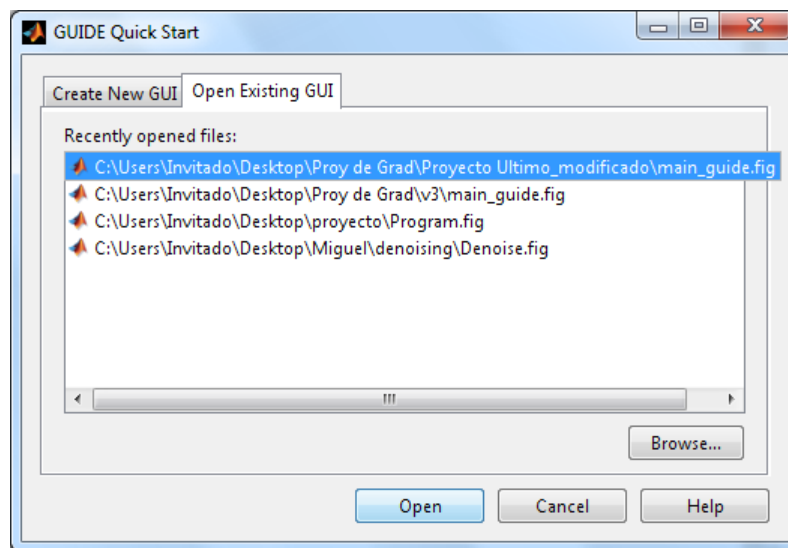
Cargar la dirección donde se encuentra la carpeta del proyecto en el 'Current Directory' de MatLab



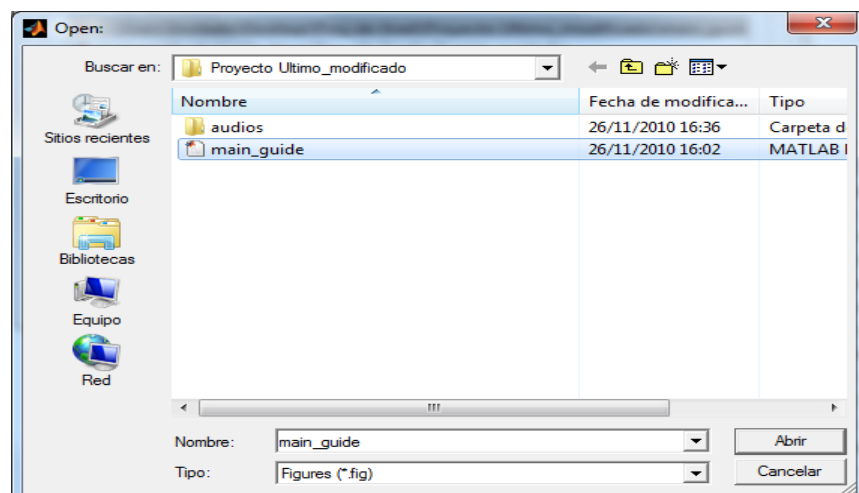
Pulsar el icono del GUI de MatLab



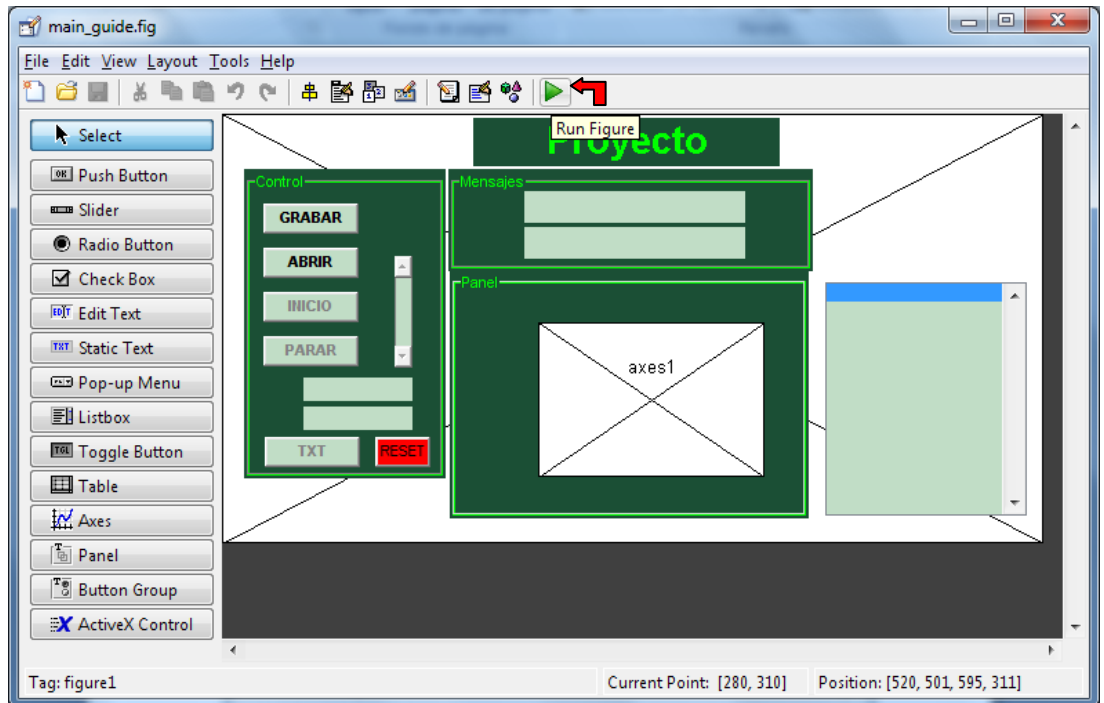
Damos click en "Browse..." y buscamos nuestro archivo .fig que corresponde a nuestro programa.



Damos click en Abrir: main\_guide.fig



Damos click en Run Figure



Aparecerá una pantalla como la siguiente:



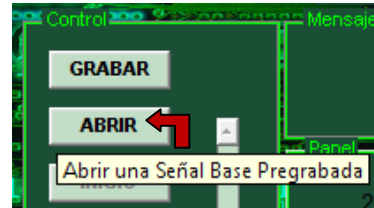
Esta ventana es la interfaz del programa con el Usuario, con la cual se va a interactuar.

El primer paso que hay que realizar es grabar una Señal Base de las cuales hay dos opciones:

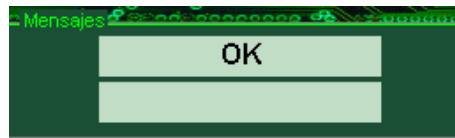
1. Grabar una nueva Señal Base



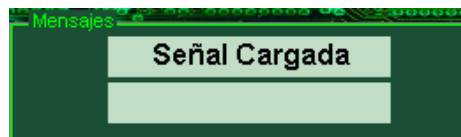
2. Abrir una Señal Base Pregrabada



Si se escogió la primera opción, al dar clic, esperar (3 segundos) hasta que se muestre el mensaje de "OK".

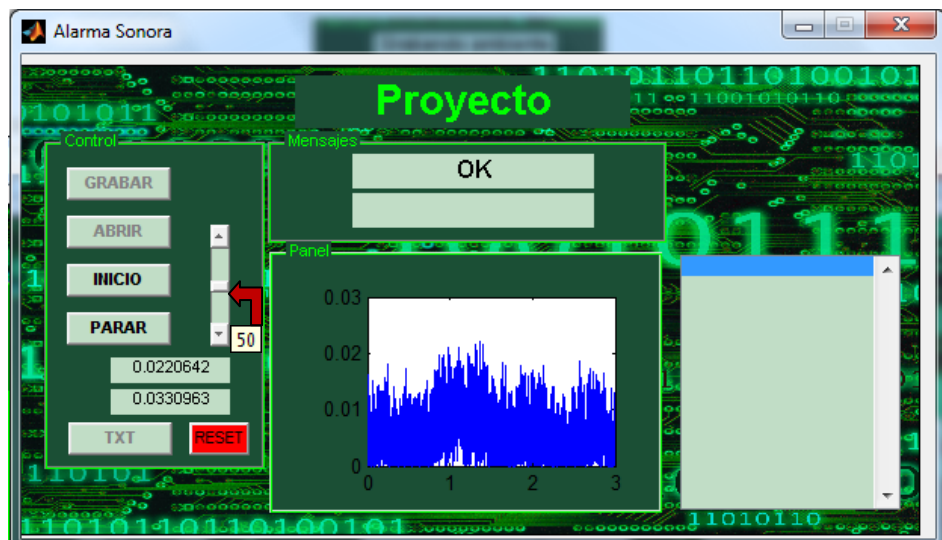


Al escoger la segunda opción, de deberá escoger el archivo pregrabado en la carpeta de trabajo, entonces se mostrará un mensaje indicando que se ha cargado una Señal Base



A continuación se grafica la Señal Base y los indicadores debajo de los botones muestran su valor máximo en Magnitud, y el valor multiplicado por el margen que muestra el Slider respectivamente.

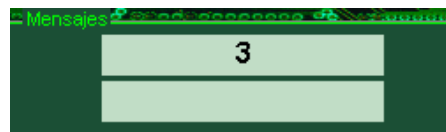
Por defecto el Slider esta al 50%, que representa la Sensibilidad de la Alarma. Esta puede ser modificada a conveniencia del usuario.





Una vez grabada o cargada la Señal Base, los botones de “GRABAR” y “ABRIR” se deshabilitan automáticamente para evitar problemas posteriores. Si se desea grabar o abrir una Señal Base nueva, es necesario pulsar el botón de “RESET”.

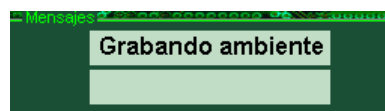
Luego para activar el sistema, damos click en INICIO.  
Se mostrara un contador descendente que representa un retardo en Segundos.  
Al llegar a cero se activa la alarma.



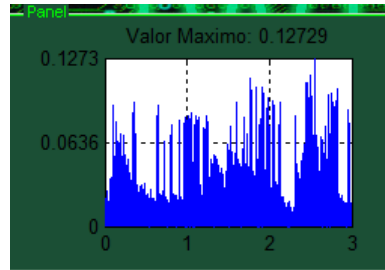
Al activarse el sistema este comienza a adquirir y procesar las señales.



En el panel de “Mensajes” se muestra un mensaje “Grabando ambiente”, ya que se adquiere la señal durante 3 segundos que dura la grabación y se la procesa.



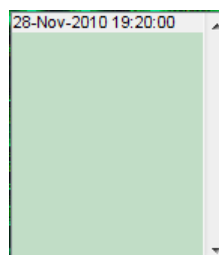
En la ventana del grafico se muestra la señal que se ha grabado y que al procesarla se compara con la Señal Base.



Si hay un sonido extraño o fuera de lo normal, la alarma se Activará y en el panel de "Mensajes" se indicara que hubo una anomalía.



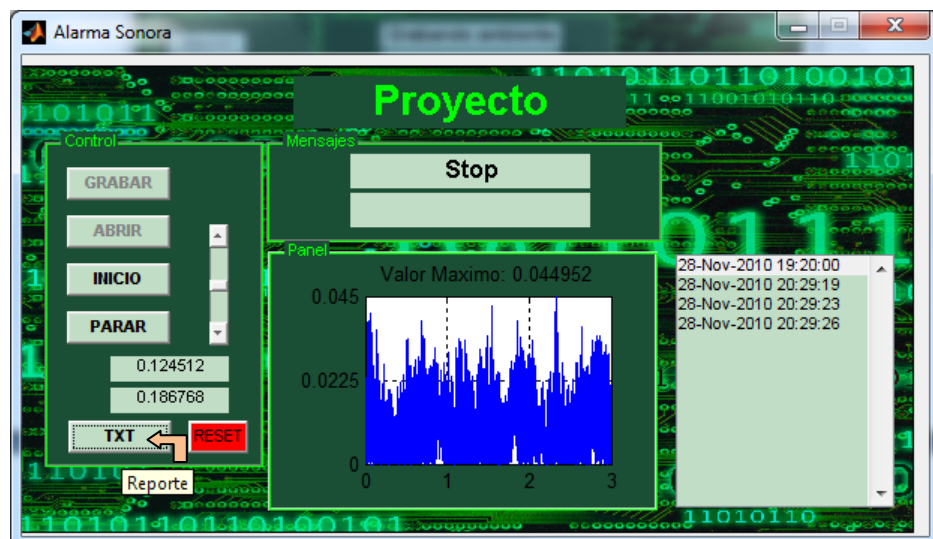
Adicionalmente en el "Cuadro de mensajes" que se presenta en la parte derecha, se muestra la fecha y hora en el que ocurrió la anomalía. Posteriormente éste se presentará en el Reporte, que más adelante se explicará.



El sistema estará grabando y comparando hasta que el usuario pulse el botón de “PARAR” y cada vez que suene la alarma se grabara en el “Cuadro de mensajes” la fecha y la hora de las anomalías. Estos se mostraran en el Reporte posteriormente.

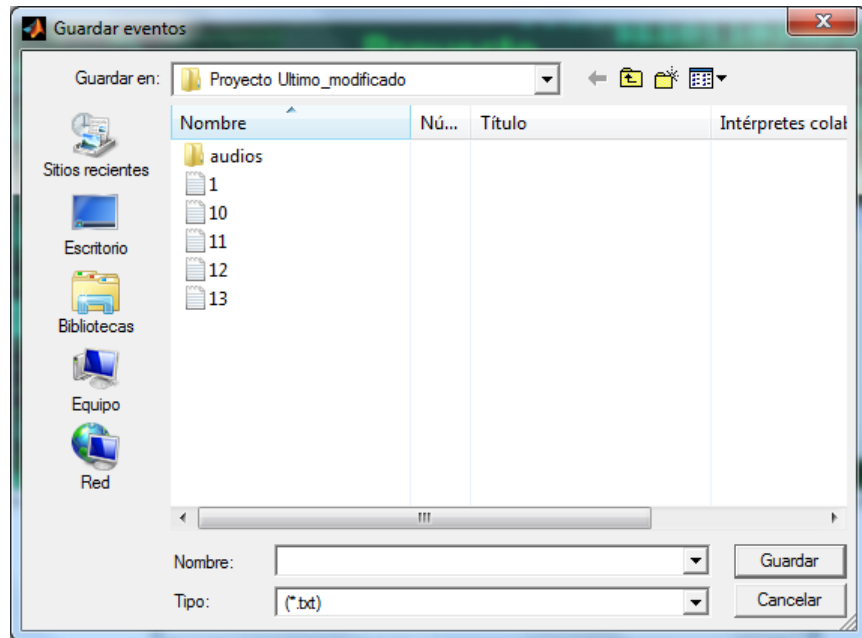


Para detener el sistema el usuario deberá pulsa “PARAR”, y se mostrará un mensaje que dice “Stop” que nos indica que ha parado la adquisición de sonido y el proceso a realizar. Se habilita el botón de “TXT”

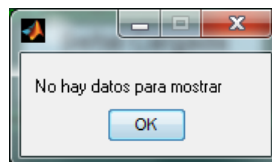


Nos podemos dar cuenta que el botón de “INICIO” sigue activo, por lo tanto se puede volver activar el Sistema, pero OJO, los datos en el “ListBOX” se perderán y por ende no saldrán en el reporte a menos que el Usuario pulse el botón de “TXT” y guarde los datos en cualquier carpeta que se crea conveniente.

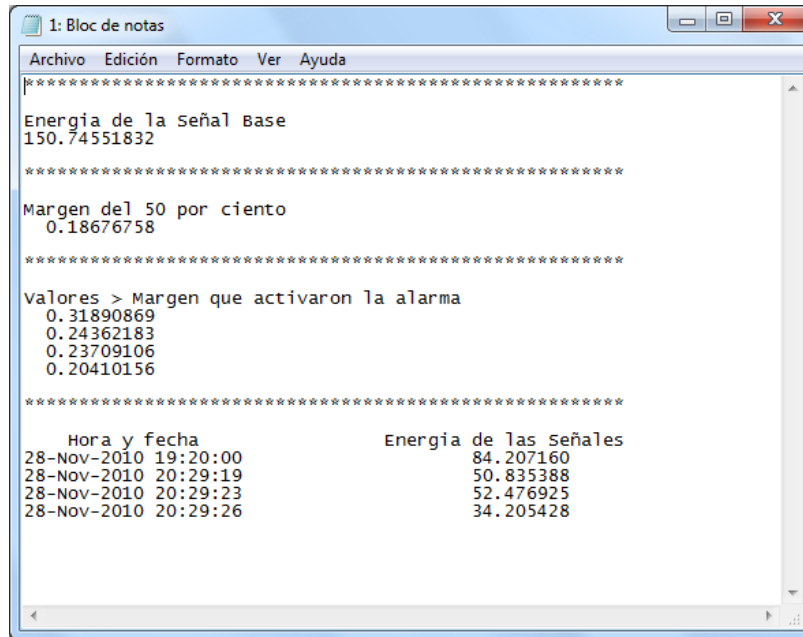
Cuando pulsamos “TXT” aparecerá una ventana que nos permite guardar el Reporte donde el Usuario desee. El nombre del Reporte depende del Usuario y el formato es un archivo .txt. La ventana es la siguiente:



Si se pulsa el botón de “TXT” y nunca se activo la alarma saldrá un mensaje indicando que no hay datos que mostrar y en consecuencia no se guarda nada.



Un ejemplo del Reporte es el siguiente:

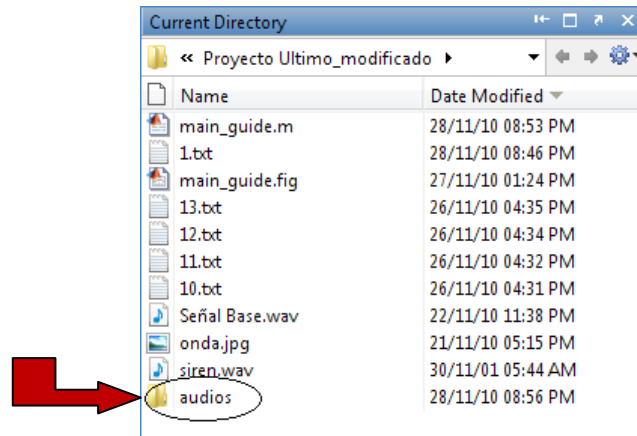


```
1: Bloc de notas
Archivo Edición Formato Ver Ayuda
*****
Energía de la Señal Base
150.74551832
*****
Margen del 50 por ciento
0.18676758
*****
valores > Margen que activaron la alarma
0.31890869
0.24362183
0.23709106
0.20410156
*****
Hora y fecha                                Energía de las Señales
28-Nov-2010 19:20:00                          84.207160
28-Nov-2010 20:29:19                          50.835388
28-Nov-2010 20:29:23                          52.476925
28-Nov-2010 20:29:26                          34.205428
```

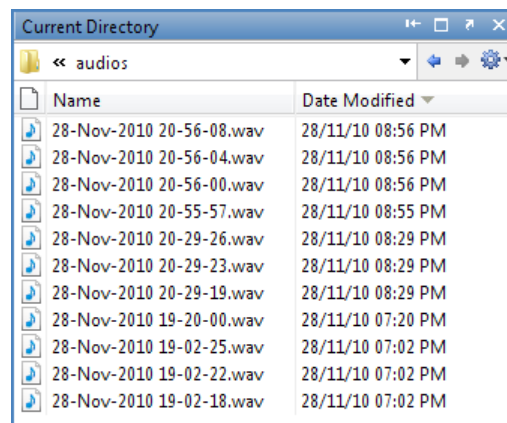
El reporte consta de lo siguiente:

1. Energía de la Señal Base
2. Margen que se ha escogido. En este caso del 50 por ciento.
  - a. Este valor es el resultado de multiplicar el valor máximo de la Señal Base por 1.5 que sería el 50% más. Este resultado es el valor a comparar con las señales pos grabadas.
3. Los siguientes valores son las magnitudes de las señales que activaron la alarma. Nos podemos dar cuenta que son mayores que el Margen del 50 por ciento mostrado anteriormente.
4. Lo último que se muestra son las Fechas y Horas que ocurrieron las anomalías con sus respectivas Energías (Energías de las Señales). Si nos damos cuenta estas horas y fechas son las mismas que se mostraron en el “Cuadro de mensajes” de la interfaz.

Adicionalmente, en la carpeta de trabajo, en la que cargamos el programa al principio, en el “Current Directory” hay una carpeta llamada “Audios”

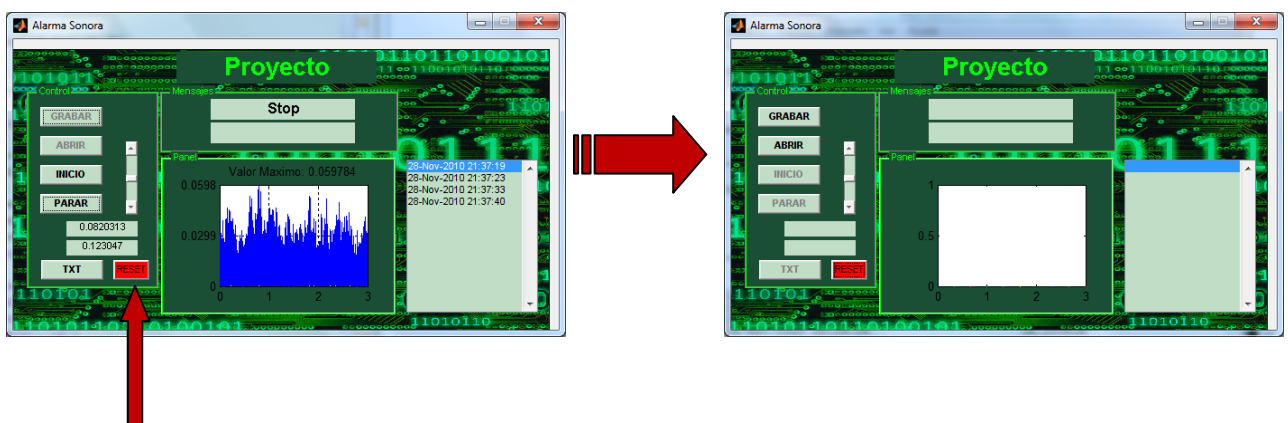


Esta carpeta contiene los archivos de sonido “.WAV” en los que hubo anomalías y por ende los causantes de que la alarma se activara



Como podemos observar tiene el mismo formato que se presenta en el Reporte, FECHA y HORA para una mejor identificación, y brindar una facilidad de búsqueda al usuario.

Por Ultimo si queremos Reiniciar el programa y borrar todo lo que tenemos graficado y mostrado en la INTERFAZ, basta con presionar el botón de “RESET”



## REFERENCIAS

- [1] BURRUS SÍDNEY C, ET AL, EJERCICIOS DE TRATAMIENTO DE LA SEÑAL, PRENTICE HALL, SEGUNDA EDICIÓN, 1998.
- [2] ENDER JOSÉ LÓPEZ MÉNDEZ, ET AL, TUTORIAL ELECTRÓNICO DEL PAQUETE MATLAB, FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DEL ZULIA MARACAIBO VENEZUELA, 20.
- [3] JAMES EASTHAM, ET AL, MATLAB GUI: CREATING A GRAPHICAL USER INTERFACE WITH MATLAB, [HTTP://WWW.MATHWORKS.COM/PRODUCTS/FEATURED/VIDEOS/](http://www.mathworks.com/products/featured/videos/) 2009.
- [4] HIGHAM, D.J., HIGHAM, N.J., ET AL, MATLAB GUIDE. SOCIETY FOR INDUSTRIAL AND APPLIED MATHEMATICS, 2000.
- [5] EDWARD W. KAMEN AND BONNIE S. HECK, FUNDAMENTALS OF SIGNALS AND SYSTEMS, PRENTICE HALL, INC., SECOND EDITION, 2000.
- [6] J.G. PROAKIS, D.G. MANOLAKIS. DIGITAL SIGNAL PROCESSING: PRINCIPLES, ALGORITHMS AND APPLICATIONS". PRENTICE-HALL, INC. 1996.
- [7] SANJIT K. MITRA, PROCESAMIENTO DE SEÑALES DIGITALES: UN ENFOQUE BASADO EN COMPUTADORA, 3RA EDICIÓN, MCGRAW-HILL, 2007.
- [8] RAYMOND A. SERWAY, ET AL, FÍSICA TOMO 1 MC GRAW HILL INTERAMERICANA DE EDITORES, CUARTA EDICIÓN, 1997.