División de Estudios de Posgrado e Investigación

Transformada rápida de Fourier (FFT) en aplicación móvil sin internet para control por detección con comandos de voz

TESIS

PRESENTADA COMO REQUISITO PARCIAL PARA
OBTENER EL GRADO DE:

Maestría en Ingeniería Electrónica

Ing. Leonor García Gámez

Director:

M. C. Carlos Alberto Pereyra Pierre

Resumen

Este trabajo presenta un sistema de reconocimiento de comando de voz utilizando la trasformada rápida de Fourier. Este sistema está compuesto por una aplicación desarrollada para dispositivos móviles con sistema operativo Android, que interactúa con el ambiente y el usuario, utilizando los recursos del equipo móvil, tales como micrófono y control de volumen. Con la identificación de un comando corto de voz, independientemente del interlocutor, se tiene acceso al control de bajar el volumen de un reproductor de música al reconocer el comando “oye”.

Abstract

This work presents a voice command recognition system using the fast Fourier transform. This system is composed of an application developed for mobile devices with Android operating system, which interacts with the environment and the user, using the resources of the mobile equipment, such as microphone and volume control. With the identification of a short voice command, regardless of the speaker, you have access to the control of lowering the volume of a music player by recognizing the command "oye".

Dedicatoria

A mis hijos, Ian e Isaac, a quienes les dejo el legado de disfrutar la vida, de atreverse a crecer y desarrollarse con alegría, pero sobre todo con perseverancia. Cada vez que le pides algo a Dios, a la vida, te preguntarán que si de verdad lo deseas, y ustedes contestaran con alma, corazón y vida que sí. Un sí que los lleva a esforzarse hasta el último aliento para lograr obtener eso tan querido.

Al gran ausente, mi príncipe, mi primer amor, mi papá, por poner en mi la fortaleza de tomar con alegría cada proyecto emprendido.

A mi mamá, por su amor infinito, por enseñarme a no darme por vencida, a ser tenaz y perseverante.

A cada uno de mis hermanos, Alma Sofía, María Luisa, Haydée, Clarissa, Eduardo Antonio, Sandra Guadalupe, por hacer de este viaje de la vida algo maravilloso.

Agradecimientos

"Yo, si tuviera hambre y estuviera desvalido en la calle
no pediría un pan; sino que pediría medio pan y un libro"
Arantza Echaniz Barrondo

Mi agradecimiento a la vida que me ha dado a manos llenas, brindándome oportunidades que jamás imaginé, y que disfruto cada día. A Dios, primero que nada y que nadie, por darme esta vida y todo lo que con ella viene.

En este caminar descubrí grandes amigos, empezaré por mis compañeros de esta aventura, Adrián Francisco Gallardo Antúnez, Fernando Joaquin Ramírez Coronel, Horacio Valencia Córdova, Juan Antonio Borboa Griego, Jesús Naim León Ortega, quienes me apoyaron los días de risas y los días de lágrimas. Pasamos los mejores momentos entretejiendo este sueño que juntos emprendimos hace dos años.

A la institución de la que egresé, en la que trabajo y me dio la oportunidad de estudiar esta maestría. A la subdirectora Martha Díaz, quien me apoyo desde el inicio, para que esto fuera posible.

A todos mis maestros, ya que todos dejan una gran enseñanza tanto en mi espíritu como de conocimiento. Maestro José Manuel Chávez, gracias por ser ese gran apoyo de siempre, desde la licenciatura, como profesionista, y hoy nuevamente su alumna. Al maestro Rafael Galaz, por ser mi guía en este oficio de la educación, José Antonio Hoyo Montaño, que siempre me invitas a proyectos que me hacen crecer como persona y seguir desarrollándome profesionalmente. Doctores Jorge Sakanasi, fue una muy grata experiencia aprender de usted; así como Rosalía Gutiérrez y Guillermo Palomo, muchas gracias por sus enseñanzas.

Un agradecimiento muy especial a mí asesor, Carlos Alberto Pereyda Pierre, quien con su Fe en mí, me permitió dar lo mejor y concluir con alegría este proyecto, haciéndolo una grata experiencia.

Por último, a mi gran amiga, la maestra Ana Luisa Millán, que me contagió con su entusiasmo y pasión por hacer cada día mejor las cosas, en constante alabanza a Dios.

Índice General

Contenido

El misterio es la cosa más bonita que podemos experimentar. Es la fuente de todo arte y ciencia verdaderos

Albert Einstein

Capítulo I: Introducción

1.1. Antecedentes

La Trasformada Rápida de Fourier ó FFT por sus siglas en inglés, (Fast Fourier Transform), es de gran importancia en un amplio ramo de aplicaciones en física, ingeniería, óptica, espectroscopia, ingeniería eléctrica, electrónica, comunicaciones y ciencias de la computación. [1]

El 90 % de los eventos físicos tienen que ver con vibraciones y formas de onda de un tipo o de otro. El sonido es un ejemplo de ello, cuando un instrumento musical se hace sonar y un micrófono recibe la señal que no es otra cosa que una presión de aire instantánea y con ello produce un voltaje proporcional, mientras en un osciloscopio se puede observar una gráfica de presión contra el tiempo, una función del tiempo periódica, f(t), y el recíproco es una frecuencia de la nota musical. [1]

La forma de onda no es una sinusoidal pura, contiene armónicos múltiplos de la frecuencia, con varias amplitudes y en varias fases, dependiendo del timbre de la nota. La forma de onda puede ser analizada para encontrar las amplitudes y tonos, a través de una serie de amplitudes y fases de la sinusoidal que la comprende.[1]

Así mismo, las señales de voz pueden analizarse de la misma forma, a través de métodos y algoritmos que permitan transformar las señales en el dominio del tiempo al dominio de la frecuencia, obteniendo así el sonograma de la voz, a partir del cual se pueden realizar caracterizaciones espectrales del sonido y con ello el reconocimiento del habla. Un algoritmo ampliamente utilizado para el reconocimiento del habla es la Transformada de Fourier.[2]

La fortaleza del análisis de Fourier consiste en que puede descomponer una señal compleja en un conjunto de componentes de frecuencia única; sin embargo, no indica el instante en

que han ocurrido. Esta descomposición es útil para señales estacionarias donde las componentes de la frecuencia que forman la señal compleja no cambian a lo largo del tiempo. En cambio, para señales no estacionarias se deben tomar tramos o ventanas donde se puede considerar estacionaria y así poder aplicar la transformada de Fourier. [3]

Existen diversos trabajos de tesis a nivel licenciatura y posgrado que han incursionado en el reconocimiento de voz. Algunos de ellos, Hernández Mora, Huerta de la Fuente, utilizaron la transformada de Fourier y proponen el uso de alguna de las funciones de ventana para un filtro digital tipo FIR (Finite Impulse Response) [4]. Así mismo, como Bonnet, Gutiérrez, en 2013, utilizaron la transformada rápida de Fourier para reconocimiento de voz.[5]

1.2. Justificación del tema de tesis

El uso de audífonos especialmente en los jóvenes, los aísla auditivamente y se quedan prácticamente incomunicados. El intentar hablar con una persona que porta audífonos puede tornarse difícil y a veces un problema. En un estudio a los alumnos del Colegio Nacional Bartolomé de Mitre, el 95% de los estudiantes lo utilizan.[6]

Se ha documentado el riesgo de andar en la calle con audífonos. El Centro de Investigación Pew afirmó que se triplicaron los accidentes mortales entre 2004 y 2011, debido al uso de audífonos en las vialidades públicas. En el 29% de los casos se realizó una advertencia sonora, como un silbato, sirena o claxon antes de la colisión. El Doctor Richard Lichenstein, director de la investigación, explica que la causa más probable del accidente fue la privación sensorial que provoca el uso de audífonos. [7], [8]

Los dispositivos móviles inteligentes cuentan con micrófono, por lo que se pudiera aprovechar este recurso en conjunto con las herramientas matemáticas de la transformada rápida de Fourier y el uso de alguna de las ventanas tipo Hamming, Kaiser, Blackman o Dolph-Chebyshev para crear una aplicación móvil que reconozca comandos de voz, independientemente del interlocutor. Sería importante poder gestionar estos recursos para

crear una aplicación móvil que reconozca comandos fonéticos y poder tener acceso a las funciones del dispositivo, para reducir el volumen del reproductor de música y de esta manera contribuir a reducir los índices de los accidentes de la problemática expuesta, debido al uso de audífonos. [9]

1.3. Planteamiento del problema

Considerando que se torna difícil comunicarse con una persona que utiliza audífonos para escuchar música, contar con una aplicación que baje el volumen totalmente cuando se utilice el comando corto de voz "oye", le ayudaría al usuario a mejorar su estado de alerta y recuperar su escucha activa del entorno. Dado que se ha citado el uso del móvil como un distractor en conductores y peatones [10], el problema se centra en el estado de alerta del usuario.

1.4. Hipótesis

Es posible desarrollar una aplicación que detecte fonemas específicos y controle el volumen de un reproductor de música en un dispositivo móvil inteligente.

1.5. Objetivos generales y particulares

1.5.1. Objetivo general

Desarrollar una aplicación de procesamiento digital de señales que detecte fonemas específicos y controle la operación del volumen de un dispositivo móvil sin requerir acceso a internet.

1.5.2. Objetivos particulares

» Establecer el modelo de procesamiento con herramientas matemáticas que permita el mejor reconocimiento de fonemas.

» Diseñar la arquitectura del Software.

» Desarrollar un simulador en Java con capacidad de reconocer un comando corto de voz “oye” y baje totalmente el volumen de música, corriendo en dos hilos: por un hilo el reproductor de música y en el segundo hilo, el reconocimiento de voz y la interrupción del primer hilo.

» Desarrollar una aplicación móvil en Android Studio que funcione sin la necesidad de conexión a internet, utilizando dos planos, la aplicación y un servicio.

1.6. Metas y alcances

» Establecer un modelo matemático adecuado para el reconocimiento de voz a través de la transformada rápida de Fourier y el método de ventana apropiado para minimizar el error.

» Diseñar la propuesta de una arquitectura de software.

» Realizar una simulación en Java de la arquitectura de software planteada utilizando el paradigma de la programación orientada a objetos en dos hilos; en el primero el reproductor de música y en el segundo, el reconocimiento del comando de voz “oye” que modifique la acción del primer hilo.

» Desarrollar una aplicación móvil en dos planos, obteniendo una aplicación y un servicio.

1.7. Organización de la Tesis

En el primer capítulo se dan a conocer antecedentes, justificación de éste trabajo, hipótesis, objetivos generales y específicos, metas y alcances, para terminar con la exposición de la organización del mismo. El segundo capítulo desarrolla el marco teórico, y explica las bases teóricas implementadas en el diseño del reconocedor de voz, esto es, la FFT y se define el tipo de ventana adecuada para una mejorar calidad en el reconocimiento de voz. Además, se expone la utilización de un modelo unificado (UML) para el análisis y diseño de software orientado a objetos. Más aún se presenta la forma de traducir este modelo en Java manejando dos hilos para posteriormente implementarlo en Android Studio, en dos planos. En el tercer capítulo se explican los pasos a seguir para desarrollar la propuesta de solución. Se explica la conformación de la FFT en la plataforma Android Studio, la forma en que se requieren los permisos para controlar el volumen del reproductor de música del dispositivo y el permiso para la validación de que los audífonos estén propiamente conectados, todo ello a través de una aplicación y un servicio que permita el trabajo en segundo plano, para la ejecución simultanea del reproductor de música y el control por comando de voz. En el capítulo cuarto se documenta la comprobación del sistema mediante la realización de la prueba de confiabilidad bajo el modelo estadístico del límite central para obtener índice total de fallas y el intervalo de confianza y se muestran los resultados obtenidos. En el capítulo V y último, se analizan los resultados obtenidos y el trabajo futuro que puede desarrollarse a partir de ello.

Capítulo II: Marco Teórico

Existen diversos trabajos de tesis a nivel licenciatura y posgrado que han incursionado en el reconocimiento de voz. Algunos de ellos, Hernández, Mora, Huerta de la Fuente, en el cual proponen utilizar series de Fourier y FIR (Finite Impulse Response).[11]

Así mismo, como Bonnet, Gutiérrez, en 2013, utilizaron la transformada rápida de Fourier para reconocimiento de voz. [5]

Torres Hernández, en su tesis de maestría en Ciencias en Computación, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, en el 2013, utiliza en su investigación la trasformada rápida de Fourier, aunque en su conclusión propone utilizar la transformada Wavelet.[9]

2.1. Características acústicas

Los primeros trabajos en sistemas de reconocimiento de voz fueron guiados por la teoría fonético-acústica, los elementos básicos del sonido del lenguaje, intentando explicar cómo estos realizan una expresión hablada. Se determina las principales regiones de concentración de energía en el espectro de potencia del habla. Estos modelos de resonancia, llamados frecuencia formante, se manifiestan como la región de mayor concentración de energía en el espectro de potencia de la señal de la onda registrada del habla. [11]

En 1952, en los Laboratorios Bell, K. H. Davis, R. Biddulph y S. Balashek, construyeron un sistema de reconocimiento de dígitos del 0 al 9, independiente del interlocutor, solo de las características del espectro de cada número. En 1953, Walter Lawrence creó el primer sintetizador de voz basado en frecuencias formantes, al que llamó PAT (Parametric

Artificial Talker). En 1956, Oslo y Belar de los laboratorios RCA, construyeron un sistema de reconocimiento de 10 sílabas, independientemente del interlocutor y únicamente utilizando el análisis espectral. En esa misma década, el MIT Lincoln Lab Forgi y Forgie construyeron un reconocedor de 10 sílabas independientemente del interlocutor. Así mismo, en el University College of London, los investigadores Dennis B. Fry y Peter Denes, desarrollaron un reconocedor de 4 vocales y 9 consonantes, utilizando el análisis espectral y comparación de patrones, utilizando información estadística para determinar posibles secuencias de fonemas en inglés, enmarcando el uso por primera vez la sintaxis estadística (en el nivel de fonema).[12]

En 1960 varios laboratorios Japoneses diseñaron mejoras en el reconocimiento de voz, destacando la Universidad de Kyoto ya que incursionaron en el análisis discreto, empleando vocabulario limitado, dependiente del locutor, e incursiona en el sistema de reconocimiento de voz continuo. Para 1962, el físico Lawrence Kersta de Bell Laboratories, utiliza por primera vez el término "voiceprint" para un espectrograma generado por un dispositivo electromecánico. Tom Martin de Laboratorios RCA, reconoce la necesidad de detección de puntos finales, lo que mejoró la fiabilidad, mientras que Vintsyuk en la Unión Soviética propuso el uso de programación dinámica para la alineación de tiempo entre dos expresiones con el fin de derivar una evaluación de una similitud significativa. [5]

Estos avances fincaron las bases de la década de los 70´s para germinar técnicas como "*time warping*", "modelado probabilístico" que son aplicaciones de los modelos ocultos de Markov, y el "algoritmo de retropropagación". A finales de los 70´s, las publicaciones de Sakoe y Chiba sobre programación dinámica, en numerosas variantes se han convertido en técnicas indispensables para el reconocimiento automático de voz.[13]

En 1980 se trabajó en el tamaño del vocabulario, que en algunos casos alcanzaron las 20,000 palabras utilizando técnicas probabilísticas con cadenas de Markov. [9]

En la década de los 90´s se siguió trabajando en la ampliación de vocabulario y se hicieron más comunes las aplicaciones independientes del locutor y del flujo continuo.[9]

Los primeros modelos neuronales, como por ejemplo el perceptrón propuesto en los 50´s, se retoman para desarrollarse con algoritmos de aprendizaje más eficaces, utilizando lenguaje de programación de Matlab, que cuenta con una herramienta para procesamiento digital de señales, tales como procesamiento de voz y/o imágenes.[9]

En referencia [3] se menciona como la Transformada de Fourier ha sido utilizada para el estudio de la fonética acústica y cita que "Transforma una señal representada en el dominio del tiempo al dominio de la frecuencia sin alterar su contenido de información, solo es una forma diferente de representarla. La ventaja del análisis de Fourier radica en que permite mapear una señal compleja en un conjunto de sus componentes espectrales de frecuencia única. En contraparte, no indica el instante en que han ocurrido. Por ello, la descomposición es útil para señales estacionarias, las componentes de las frecuencias que forman la señal compleja no cambian a los largo del tiempo."

La mayoría de los fonólogos utilizan espectrogramas para la realización de sus investigaciones. Para señales no estacionarias tienen que ser tratadas por tramos o ventanas de la señal en donde se pueda considerar estacionaria y así poder aplicar la Transformada de Fourier. Para realizar el análisis completo debemos tomar una secuencia de ventanas para observar la evolución de las frecuencias de la señal original.[3]

El sonido se puede filtrar en el dominio de la frecuencia. Los filtros en el dominio de la frecuencia se usan, principalmente, para eliminar altas o bajas frecuencias de la señal de sonido, lo que se traduce en suavizar el sonido, o bien, realzar o detectar amplitud. [3]

El avance en los equipos computacionales, la telefonía celular y la incursión de los *Smartphone* han revolucionado la tecnología de reconocimiento de voz. Las capacidades de los *Smartphone* permiten utilizarlo como la principal interfaz física para aplicaciones de cómputo. [14]

La señal de voz se presenta como una forma continua. El lenguaje transmitido está formado por palabras que se pueden dividir en fonemas, que representan la unidad básica del habla. En cualquier lenguaje hablado, el conjunto de fonemas puede caracterizarse de acuerdo a sus vocales (anterior, central y posterior), diptongos, semivocales y consonantes (nasales, oclusivos, fricativas, africativas, flaps, y trill). Según se muestra en la Figura 1. [15]

Vocales	Los sonidos producidos por las vocales se generan cuando el aire pasa por los pulmones a la laringe y después a la boca, no existe ninguna obstrucción audible en ninguna de las vocales.	Anterior: /iy/ y /ey/ Central: /aa/ Posterior /ow/ y /uw/
Diptongos	Son vocales en las que la lengua se está moviendo durante la duración del fonema. Cuando el locutor reduce la duración del conjunto formado por dos vocales y las pronuncia de una sola vez, se forma un diptongo.	/ay/ empieza como una /a/ y termina como una /i/ /oy/ empieza como una /o/ y termina como una /i/
Semivocales	Este grupo tiene similitud con las vocales, es por eso que se les llama así. Se producen como las vocales y los diptongos, pero la lengua en posición muy extrema.	/y/ es como una /i/ extrema /w/ es como una /u/ extrema. /l/ es raro encontrarla.
Fricativas	Sonidos producidos por un cierre parcial de la boca.	Labial: /f/ Alveolar: /s/ Velar: /hx/
Stops o Oclusivos	Son sonidos dinámicos, producidos por un cierre total y después una salida repentina de aire. Se clasifican en voiced y unvoided (hablado o no-hablado), depende de como estén vibrando las cuerdas vocales.	Labial: /b/ y /p/ Alveolar: /d/ y /t/ Velar: /g/ y /k/
Flaps y trill	Los flaps son producidos cuando la lengua cierra por un momento corto el tracto vocal.	/r/ y el trill es una secuencia de flpas /rr/.
Africativos	Empiezan como un oclusivo y terminan como un fricativo.	Ejemplo: /ch/.
Nasales	Se producen cuando se cierra el tracto vocal mientras que baja el volumen del habla, dejando pasar el aire por la nariz.	Labial: /m/ Palatal /ny/ Alveolar: /n/ Velar: /ng/

Figura 1. Tabla de Fonemas.[15]

2.1.1. Representación de la señal de voz

Los sonidos son variaciones en la presión del aire a través del tiempo y a frecuencias que podemos escuchar. Una forma de representar el sonido es a través de una onda.[15]

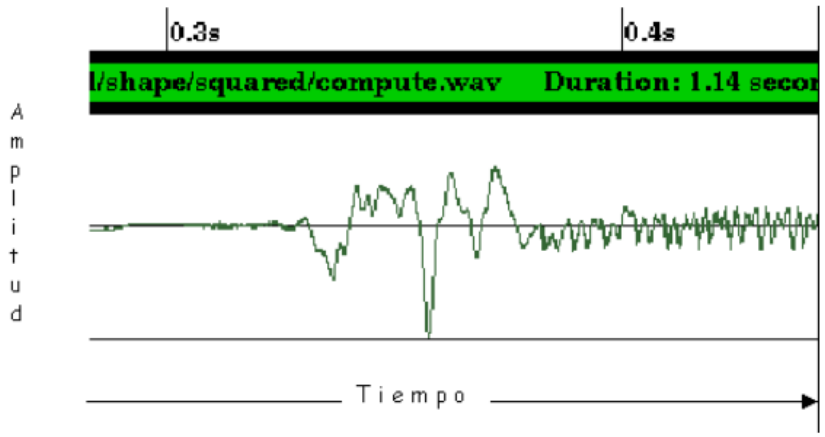


Figura 2. Ejemplo de una señal de voz. [15]

Este tipo de gráficas requieren de poca memoria, pero no describe explícitamente el contenido de la señal en cuanto a sus propiedades. Los espectrogramas contienen mayor información sobre los datos de voz, son una transformación que muestra la distribución de los componentes de la frecuencia de la señal.[15]

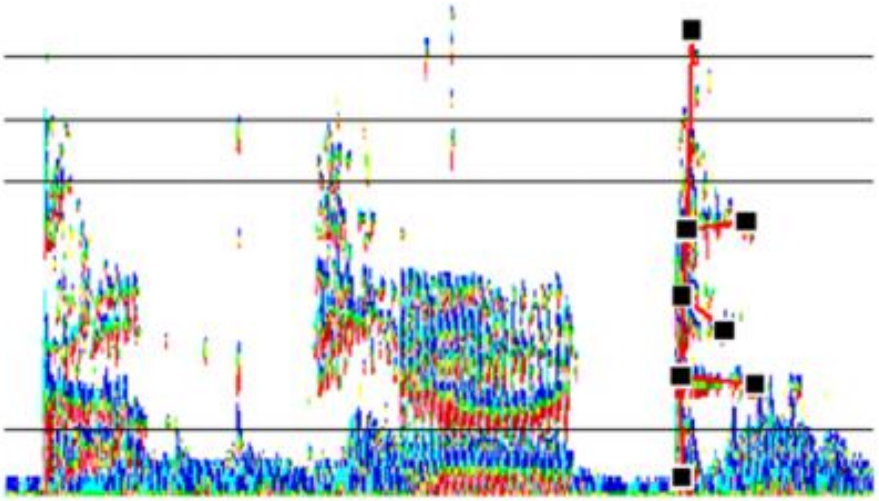


Figura 3. Ejemplo de un Espectrograma.[15]

Los puntos negros de la figura 3, ejemplo de Espectrograma, representan la concentración de energía y son denominadas formantes. El rango de frecuencias de la capacidad auditiva del ser humano varía en un rango de frecuencias de 20 Hz a 20 kHz, mientras que el habla emite sonidos desde 100 Hz y hasta 20 kHz. Se considera que la mayoría de la información sonora que se transmite se ubica por debajo de los 8 kHz. [15]

2.1.2. Arquitectura de un sistema de reconocimiento de voz

Se identifican los siguientes componentes en un sistema de reconocimiento de voz:[16]

1. Señal de entrada: Compuesto por el interlocutor, micrófono y ruido.

2. Representación y/o Extractor de Características:

    a. La señal se divide en una colección de segmentos.

    b. Se aplica alguna técnica de procesamiento de señales para obtener una representación de las características acústicas más distintivas del segmento.

    c. En base a las características obtenidas se construye un conjunto de vectores que constituyen la entrada al siguiente módulo.

3. Se plantea el algoritmo matemático:[16]
    a. Se plantea una manipulación para preénfasis

    b. Se aplica la ventana de Hamming

    c. Aplicación de la FFT

4. Decodificador General: Alineamiento en el tiempo, igualación de los patrones de articulación. [17]

La adquisición de señales de voz se compone de la captura de la señal a través de un micrófono y su digitalización por medio de alguna tarjeta de sonido.

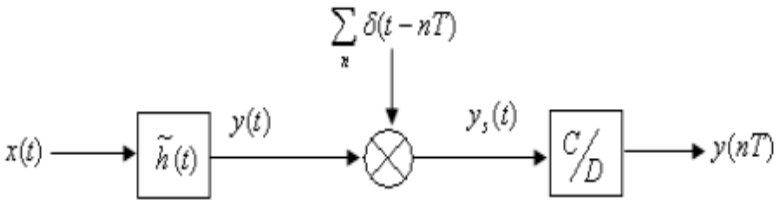


Figura 4. Diagrama de bloques del muestreo de una señal
de voz continua en el tiempo x(t)

Donde x(t) es la forma de onda original, es decir, la señal de voz, entregando el micrófono una señal, resultado de la convolución de las señales x(t) y h(t), y(t)=x(t)*h(t), muestreado a un intervalo T, a una frecuencia según el teorema de Nyquist f_c=2f, y la salida *y(*nT) estaría dada por [5] [16][18][19]

$$y(nT) = \int_{-\infty}^{\infty} h(t-nT)x(t)dt \qquad (1)$$

Donde h(t) es la respuesta impulso del micrófono, y(t) la salida del micrófono, T intervalo del muestreo, y(nT) es la salida y $n \in Z$ es el número de muestras. La adquisición de las señales se realiza utilizando una tarjeta de Adquisición de Datos DAQ (por sus siglas en inglés Data Acquisition).[16]

2.1.3. Tratamiento de las señales

Las herramientas más utilizadas para el análisis y tratamiento de señales de voz son los bancos de filtros y la transformada de Fourier. [18]

Los filtros más comunes usados para el reconocimiento de voz es un banco de filtros uniforme para una frecuencia central f_i, para un filtro pasabanda definido por: [18]

$$f_i = \frac{Fs}{N} i \quad , \quad 1 \leq i \leq Q \qquad (2)$$

donde F_s es la frecuencia de muestreo de la señal de voz, y N es el número de filtros uniformemente espaciados requeridos, y que se satisface a través de la relación $Q \leq \frac{N}{2}$ Equitativo para toda la gama de frecuencia de la voz usada en el análisis.[5]

El ancho de banda b_i, del i-enésimo filtro, generalmente satisface la propiedad: $b_i \geq \frac{F_s}{N}$ [5]

Equitativo significa que ninguna frecuencia sobrepasa los canales de los filtros adyacentes; si esto sucediera, una parte del espectro de voz se perdería y dejaría de ser un espectro significativo.[5]

La alternativa para un banco de filtros uniforme es un banco de filtros no uniforme, asignando valores según el criterio de frecuencia logarítmica, que es aceptable para la percepción auditiva humana.[18]

Por tanto, para un conjunto de Q filtro pasabanda, con frecuencia central f_i y ancho de banda b_i,

$$1 \leq i \leq Q \qquad (3)$$

Se establece que:

b_1=C (4)

bi= αb_{i-1} (5) $2 \leq i \leq Q$ (6)

$$f_i = f_1 + \sum_{j=1}^{i-1} b_j + \frac{(b_i - b_1)}{2} \quad (7)$$

donde C y f_1 son un ancho de banda arbitrario y una frecuencia central del primer filtro, y α es el logarítmico del factor creciente.[18]

El valor más común para α es α=2, el cual da una separación de banda de una octava entre los filtros adyacentes y α= 4/3 resulta en 1/3 de octava de separación entre filtros. Considérese un diseño de una banda de cuatro filtros, espaciados en octavas, sin sobre posicionar para cubrir una frecuencia a partir de 200 y hasta 3200 Hz, con una tasa de muestreo de 6.67 KHz.[19]

Filtro 1: f1= 300 Hz, b_1= 200Hz

Filtro 2: f2= 600 Hz, b_1= 400Hz

Filtro 3: f3= 1200 Hz, b_3= 800Hz

Filtro 4: f1= 2400 Hz, b_1= 1600Hz

Un criterio alternativo de diseño es la escala de banda crítica. Se han utilizado algunas variantes de este criterio con resultados muy similares.[18]

2.1.4. Interpretación de la Transformada Rápida de Fourier (FFT) en el dominio de la frecuencia

La idea básica de este algoritmo, es la descomposición de una señal compleja en la sumatoria de señales simples. El oído humano, por medio del caracol, descompone las señales auditivas que le llegan en sus frecuencias fundamentales y esta es la información

básica a partir de la cual se elaboran las señales que le llegan al cerebro. Entonces se puede afirmar que el proceso de audición se fundamente en la descomposición en frecuencias de la señal sonora, como se puede observar en las siguientes figuras:[18]

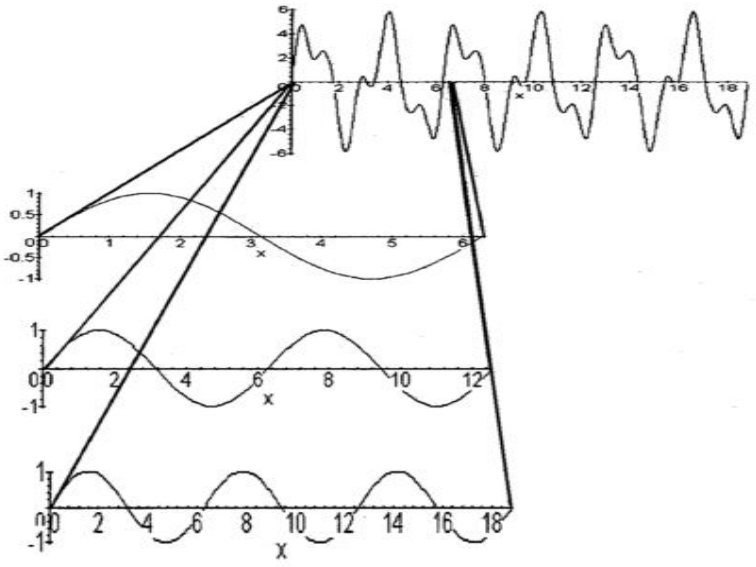
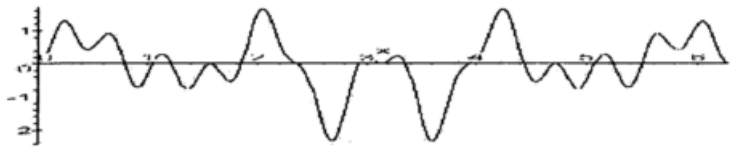
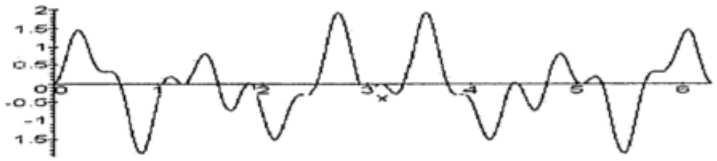


Figura 5. Comparación de funciones seno de diferentes frecuencias
con un bloque de la señal que se pretende descomponer en sus armónicos o parciales.

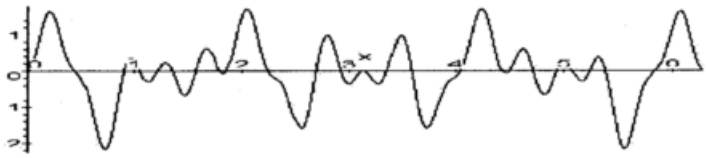
$$\big(seno(x) + seno(3x) + seno(10\,x)\big) * seno(4x))$$



$$\big(seno(x) + seno(3x) + seno(10\,x)\big) * seno(5x))$$



$$\big(seno(x) + seno(3x) + seno(10\,x)\big) * seno(7x))$$



$$\big(seno(x) + seno(3x) + seno(10\,x)\big) * seno(10x))$$

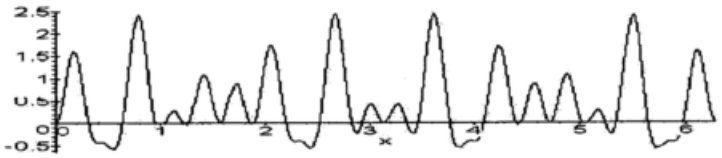


Figura 6. Señal que se pretende descomponer y resultado de compararla con senos de diferentes frecuencias.

2.1.5. Ventaneado

Se selecciona la ventana de Hamming por tener una forma similar a la señal de voz, por lo que podrá dar mejor seguimiento a la señal que una ventana rectangular o una triangular. Se le suele llamar también ventana de coseno elevado.[16]

Esta ventana tiene un comportamiento temporal de medio ciclo de una señal cosenoidal y normalizado en amplitud a la unidad. Esta ventana se caracteriza por el argumento n impar. Su ecuación es la siguiente y genera n número de muestras:

$$w[n] = 0.54 - 0.46\cos\left(\frac{2\pi n}{L-1}\right) \quad (8) \quad , \quad 0 \leq n \leq L\text{-}1 \quad (9)$$

$$y_i(k) = \sum_i^L y_i(\mathrm{k}) w_i(\mathrm{k}) \quad (10)$$

i= 1, 2, 3. . ., L y k= 1, 2, 3, K

Se obtiene la FFT de cada tramo con el objetivo de generar una superficie que pueda observar las frecuencias y su variación en el tiempo. Se promedian las FFT de cada tramo, para obtener un patrón de la palabra seleccionada, "Oye"

$$Y(k) = \sum_{n=0}^{N-1} X(n) e^{-j2\pi\left(\frac{Kn}{N}\right)} \quad (11)$$

n= 0, 1,. . ., N-1 y k= 0, 1, 2, N

2.1.6. Algoritmo FFT

La FFT se obtiene dividiendo la DFT (Transformada de Fourier discreta) de 4 puntos en dos TDF de dos puntos y combinando sus coeficientes. El algoritmo empleado se representa mediante el siguiente diagrama de flujo [19]

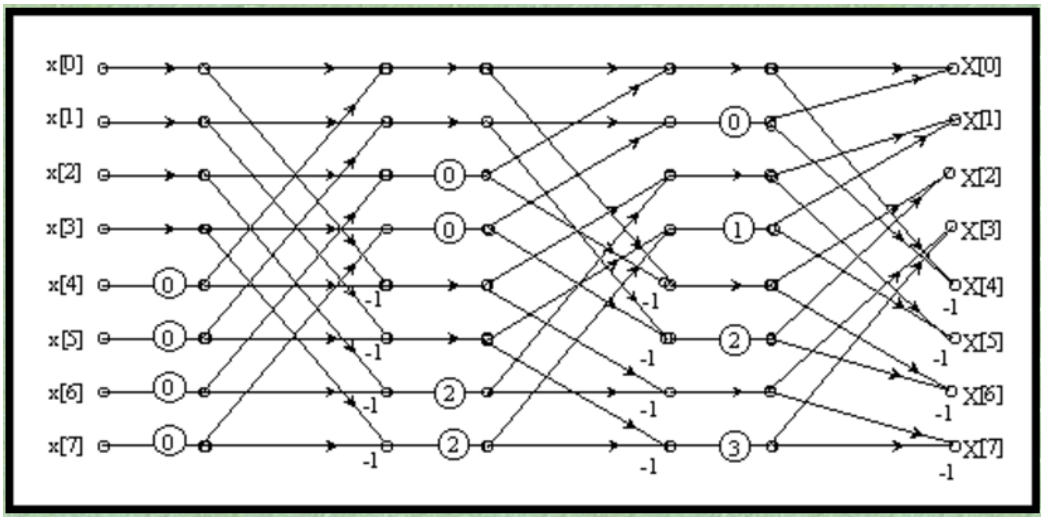


Figura 7: Algoritmo de FFT.

2.1.7. Reconocimiento de Voz con FFT

La secuencia de la transformada rápida de Fourier $Sn(\ e^{jw_i})$ se define como:[19]

$$X[k]\ = \sum_n^{N_F-1} X[n]\, e^{-j2\pi(\frac{kn}{N_F})} \quad (12)$$

Es conveniente usar la notación $W = e^{(-j2\pi(k_n/N_F\ ))}$

Si L, que es el número de tramos, es grande, relativo a la periodicidad de la señal (pitch), entonces $X[k]$ tendría una buena resolución en la frecuencia. Esto significaría que se puede resolver de forma individual los harmónicos, pero solo una aproximación en general de la sección de voz dentro de la ventana.

Si L es pequeña, relativo a la periodicidad de la señal, entonces $X[k]$ daría una resolución pobre de la frecuencia pero una buena estimación bruta sobre la forma espectral.[18]

Se plantea para este trabajo realizar un muestreo con una frecuencia de 20 KHz, para analizar segmentos de 40000 datos, de los que se discriminan los datos significativos mediante un umbral de 0.1 con respecto al tiempo. Se aplicará un filtrado preénfasis para acentuar las frecuencias altas de la señal de voz, y se segmentará la señal en tramos de 20 a 30 ms, puesto que en este período de tiempo se considera la señal de voz como estacionaria. Se analiza utilizando un traslape.

$$y(k) = X[n](MI + k) \quad (13)$$

Donde k= 1, 2, 3,. . .K-1, I=1, 2, 3, L, n= 0, 1,. . ., N-1

L= total de Tramos

M= traslape

2.1.8. Coeficientes Cepstrales

Los coeficientes Cepstrales en la escala de frecuencias de Mel (MFCC), adaptan las frecuencias de fonemas a la manera que el oído humano percibe los sonidos. Se ubican más robustos que los coeficientes de predicción lineal, llamados LPC (Lineal Predictive Code, por sus siglas en inglés, Código de Predicción Lineal) y Cepstrums.[18]

La capacidad del oído humano tiene una resolución en frecuencia de 1/9 tono, en un rango de frecuencia de 20 a 20,000 Hz, con una resolución en el tiempo de aproximadamente 400 muestras por segundo, limitado por el tiempo de relajación de células ciliadas y terminaciones nerviosas. La resolución en intensidad es mejor de 1 dB y cuenta con mecanismos de adaptación.[20]

2.1.9. Parametrización de la voz

La señal de la voz es "estacionaria a trozos". Durante la pronunciación de un fonema es quasi-estacionaria. Puede ser estacionaria durante 20 a 40 milisegundos, lo que facilita el análisis de "trozos de voz estacionarios" en ventanas.[5]

Los parámetros característicos de señales de voz están formados por el período fundamental (Pitch) y los formantes.

El período fundamental Pitch, es el tiempo transcurrido entre dos aperturas sucesivas de las cuerdas vocales. La velocidad de vibración de las cuerdas, se denomina Frecuencia fundamental de la fonación y es el inverso del Pitch.[5]

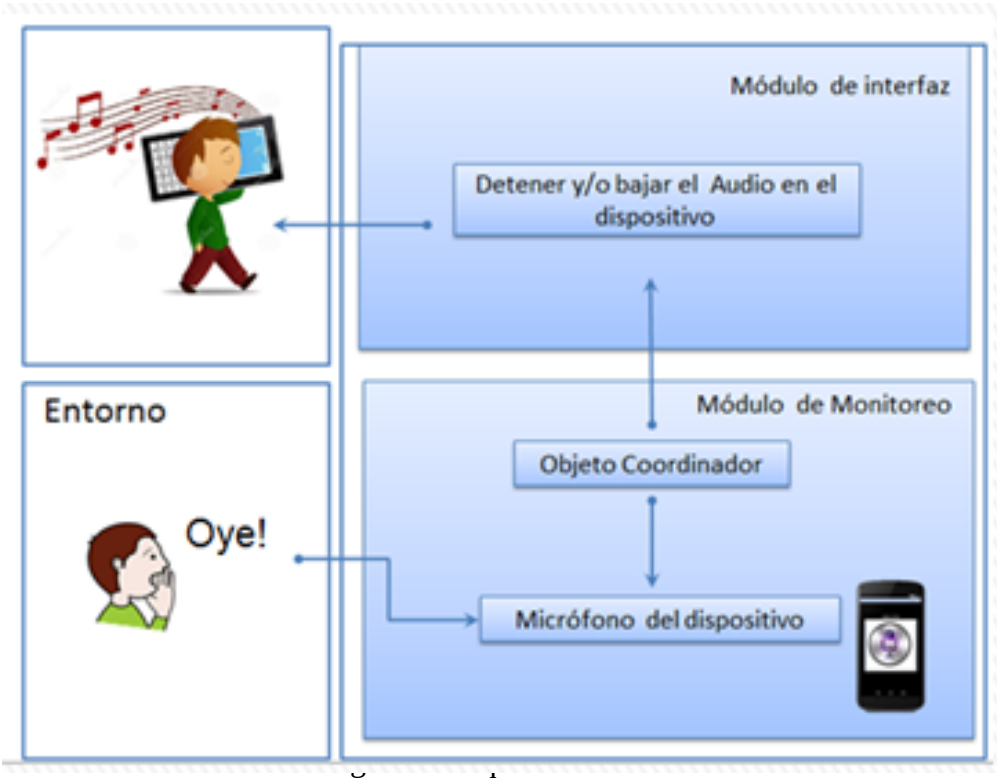
Los formantes son resonancias del tracto vocal. Por las dimensiones y velocidades de propagación del sonido aparece en promedio 1 formante por cada KHz. El tracto vocal

filtra la onda glotal, amplifica cada componente de frecuencia con una determinada ganancia.

2.2. Programación UML

2.2.1. Arquitectura

El usuario recibirá del entorno una llamada de alerta, a través de un comando “oye”, y todo lo que se esté hablando se estará captando a través del micrófono del celular. El objeto coordinador revisará si el usuario tiene los audífonos puestos, si se está reproduciendo música, al mismo tiempo que estará recibiendo el muestreo que realiza constantemente el micrófono, realiza un muestreo y recibe la confirmación si la palabra “Oye”, y este mismo objeto coordinador bajará el volumen de audio en el celular.



2.2.2. Diagramas UML

Son 3 los diagramas UML que se desarrollaron. El primero un diagrama representa los requisitos de usuario, de tal forma que nos ayude a identificar a los actores y la interacción con el sistema.

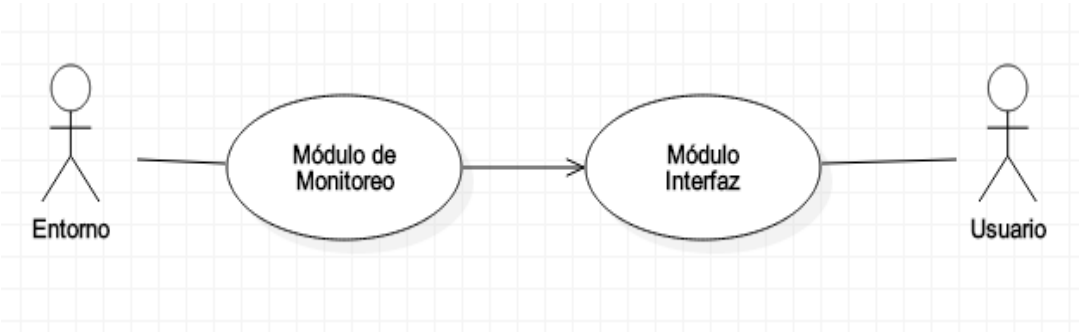


Figura 9. Diagrama de Caso de Uso.

El segundo diagrama es el Diagrama de clase, un modelado de estructura estática, que permite plasmar de forma clara y objetiva, las clases y objetos que estarán interactuando. Se reconocen 5 objetos, que son el dispositivo móvil que su función es revisar el estatus del reproductor de música, del audífono y del micrófono. La acción que puede realizar es la de bajar totalmente el nivel de volumen, así como las asociaciones con los otros objetos e identificación de subtipos, en su caso. Los diagramas de clase muestran también los atributos y operaciones de una clase y las restricciones a que se ven sujetos, según la forma que se conecten los objetos. El dispositivo muestra una relación 1 a 1 con el reproductor de música, el comando coordinador, el micrófono y los audífonos.. El Comando Coordinador es el responsable de reconocer la señal, realizando el muestreo, un preénfasis, aplicando el algoritmo de FFT, enviando la alerta de bajar el volumen del dispositivo. El micrófono se dispondrá a recibir de forma continua las señales sonoras del entorno y las enviará al Comando Coordinador. La clase audífonos revisa si éstos están conectados o no, enviando al objeto coordinador el estatus de la señal. Si se encuentran conectados los audífonos, entonces correrá el programa, de lo contrario, no realizará el algoritmo. De esto se obtuvieron las siguientes 5 clases con sus funciones y la interrelación entre ellas.

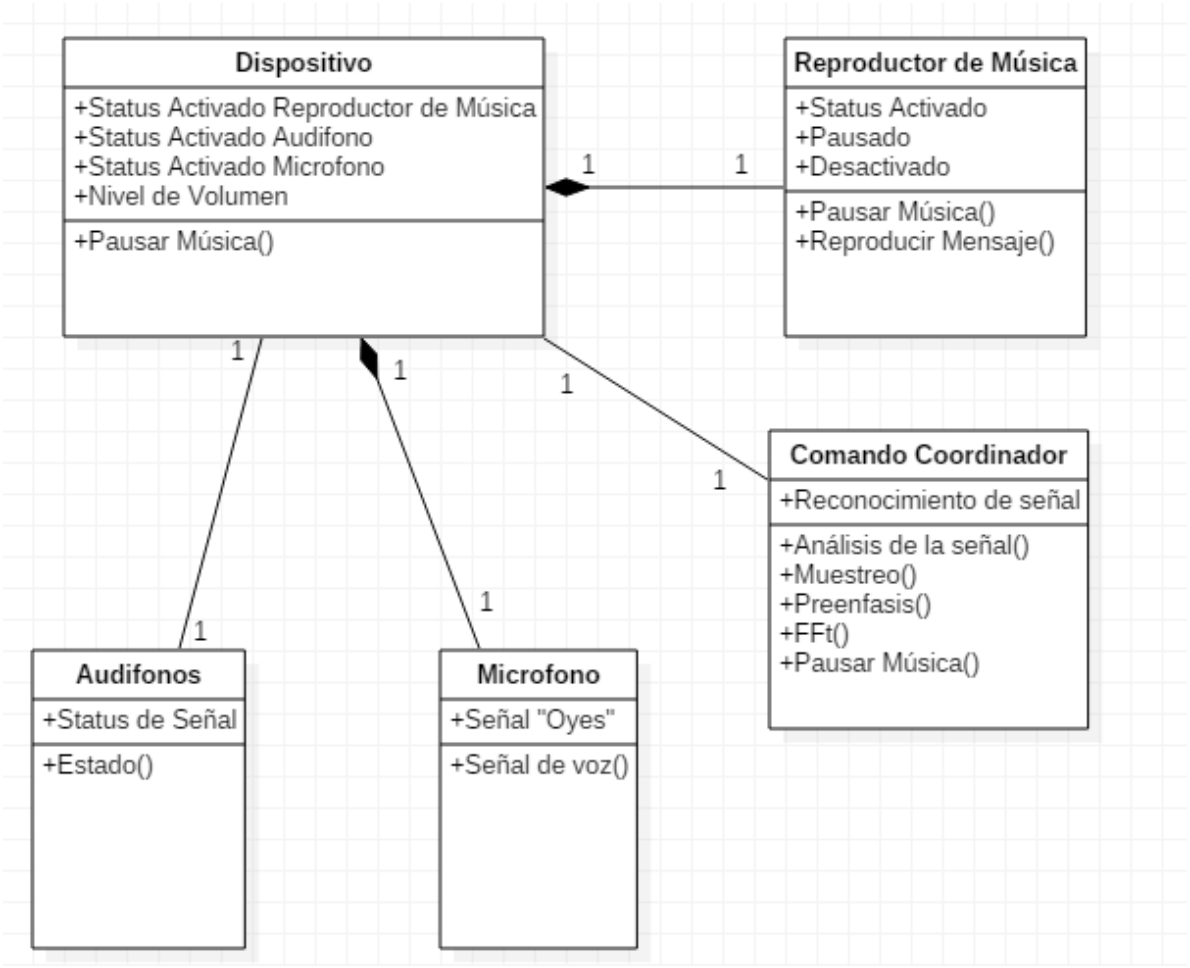


Figura 10. Diagrama de Clases.

El tercer diagrama, mostrado en la figura 11, es el de actividades, un modelado dinámico, el cual es útil para describir métodos complicados. Este muestra el disparador de entrada asociado con la actividad. Se puede tener actividades paralelas cuando se dispara la misma actividad por medio de un disparador múltiple. En este caso, se usa el disparador múltiple, cuando le corresponde verificar que hay las condiciones para correr la aplicación.

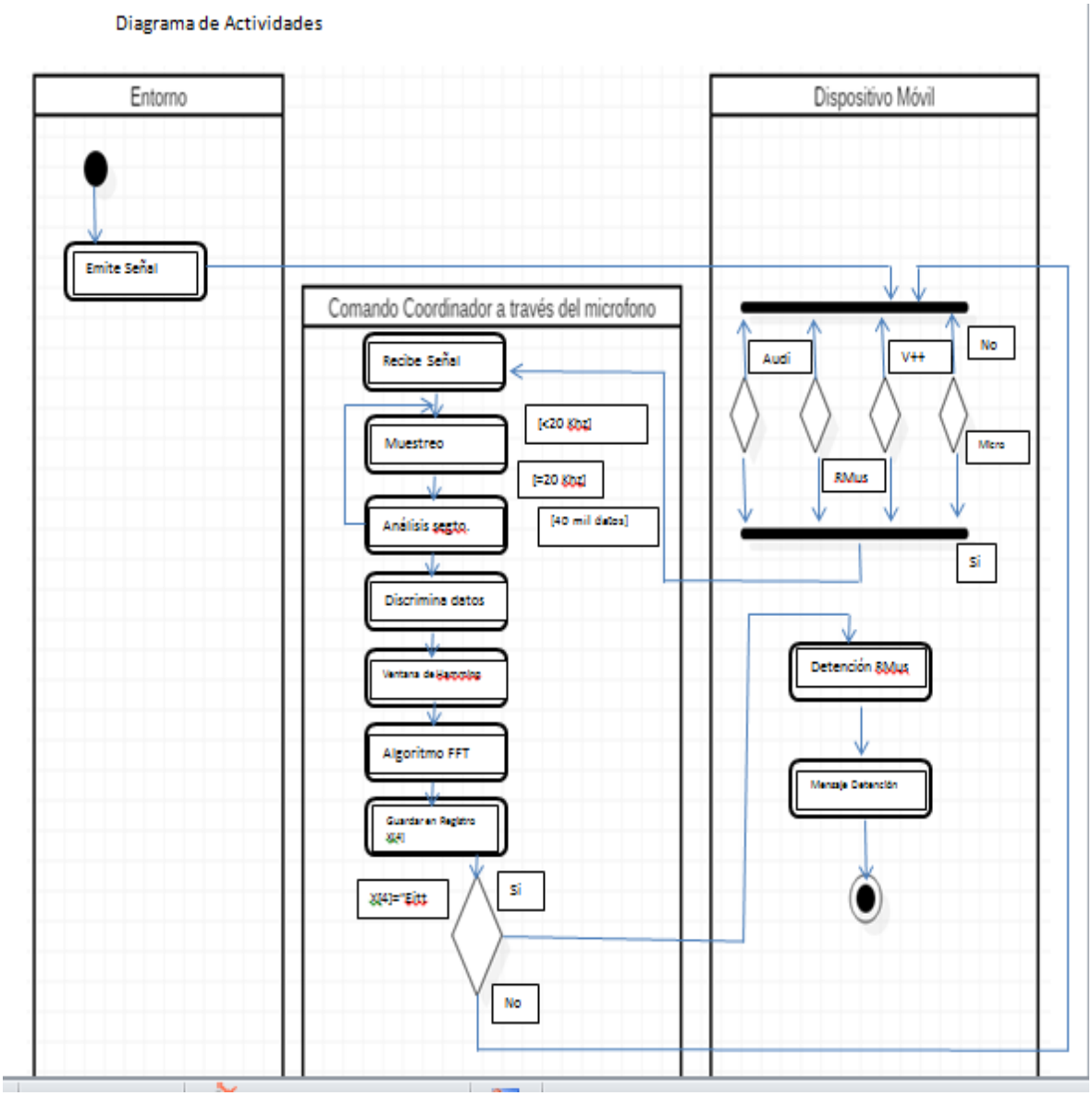


Figura 11. Diagrama de Actividades

Se utilizaron tres columnas. En la primera se encuentra señalado el entorno como punto de partida, que en forma paralela revisa si se encuentran encendidos el micrófono, el reproductor de música y los auriculares conectados. De ser así, pasa al comando coordinador que se encarga de ejecutar las órdenes. [21]

2.3. Propuesta de solución en Java

2.3.1. Programación en Java

Se determina realizar una simulación en Java, ya que puede facilitar el acceso a la programación en Android Studio. Por un lado se debe realizar un programa que reconozca un comando con voz, y por otro, un reproductor de música para ser probado, por lo que hay que utilizar un programa con dos *Threads* ó hilos.

Multihilos

Cada hilo controla un solo proceso dentro de un programa, uno de ellos se utilizará para reproducir la música y el otro para el reconocimiento del comando, y todo ello debe estar en un solo dispositivo, por lo que comparten los mismos recursos, con datos separados y su propio código.

Multihilo permite ejecutar dos o más tareas de forma simultánea, pero en realidad se hace de forma concurrente, se ejecutan en paralelo. [21]

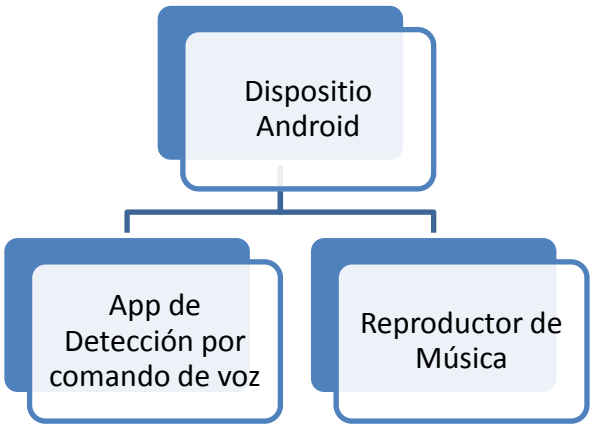


Figura 12. Estructura Multihilos Java

Comunicación entre hilos

La comunicación entre hilos se realiza a través de un modelo productor-consumidor. El hilo que realiza el comando por voz detendrá el hilo reproductor de música, entonces el primero se identificará como un productor, que irá sacando caracteres por su salida. El reproductor de música, será las veces de un consumidor, esto es, que tomará los caracteres que salgan del productor. Se crea también un monitor que controlará el proceso de sincronización entre estos.

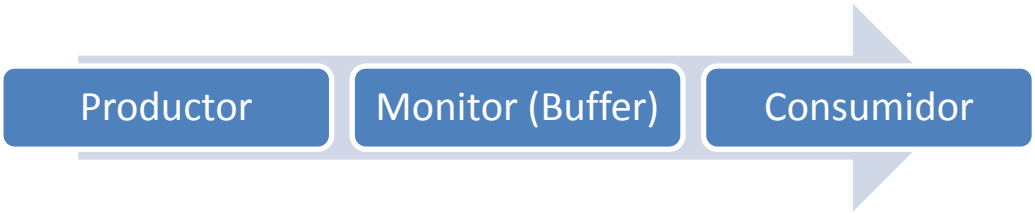


Figura 13. Comunicación entre hilos

2.3.1. Simulación de la arquitectura propuesta en Java

Programa principal, llama al comando coordinador para verificar el inicio del programa.

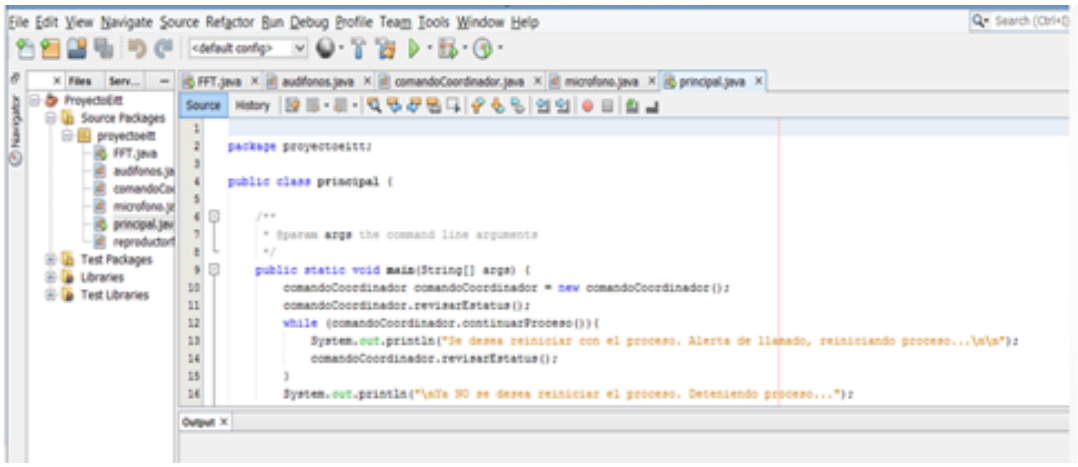


Figura 14. Pantalla del programa principal en Java

Clase audífonos, comando coordinador, micrófono

Se hace una verificación, si están conectados los audífonos si tendrá que bajar el volumen en caso de detección del comando "oye", de otra forma, enviará un mensaje indicando que no están conectados los audífonos, por lo cual no bajará el volumen del dispositivo.

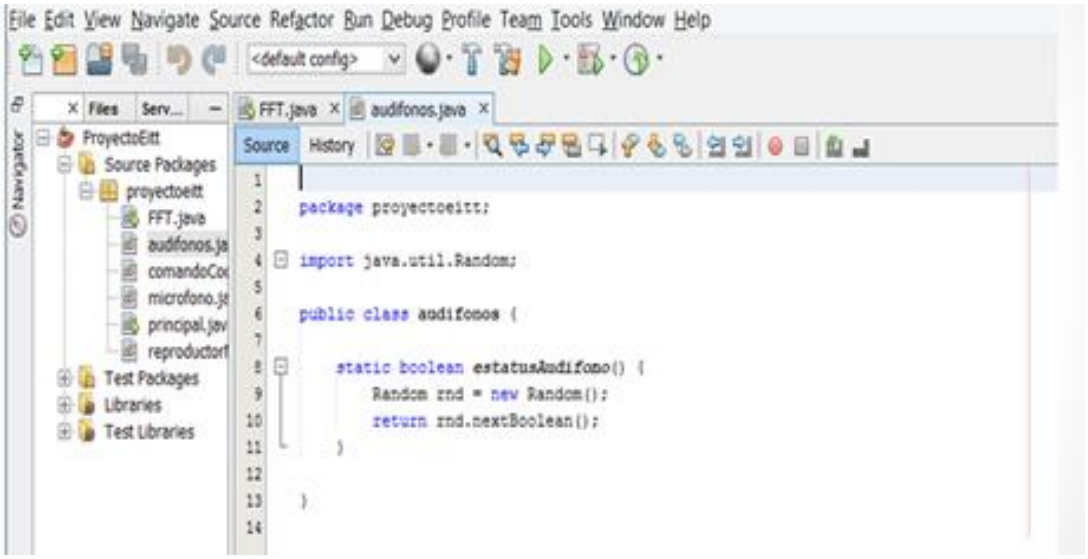


Figura 15. Pantalla de la Clase audífonos

Reproductor de música

Se utilizó código abierto para esta sección. Se necesita solo con la finalidad de probar el comando con voz.

Desarrollar un reproductor de música con detención por comando de voz corto "OYE" utilizando interrupción por hilos de primer y segundo plano.

Se realizó la simulación del proyecto "Oye", se reproduce la música, y se hace correr la simulación, donde aleatoriamente da un estado a los audífonos y se da una serie de comandos, y solo se detiene la música al reconocer el comando "oye"

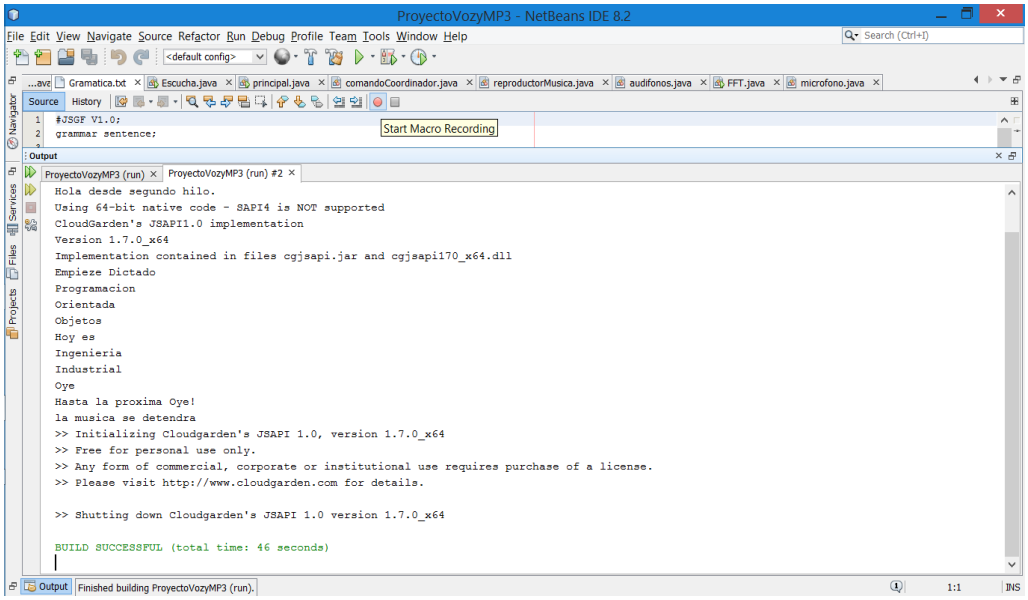


Figura 16. Pantalla de simulación de detección de comando “oye”

Código FFT en Java

```
public class FFT {
  int n, m;
  double[] cos;
  double[] sin;
  public FFT(int n) {
     this.n = n;
     this.m = (int) (Math.log(n) / Math.log(2));

     if (n != (1 << m))
        throw new RuntimeException("FFT debe ser potencia de 2");
     cos = new double[n / 2];
     sin = new double[n / 2];

     for (int i = 0; i < n / 2; i++) {
        cos[i] = Math.cos(-2 * Math.PI * i / n);
        sin[i] = Math.sin(-2 * Math.PI * i / n);
     }
  }
  public void fft(double[] x, double[] y) {
     int i, j, k, n1, n2, a;
     double c, s, t1, t2;
     // Bit-reverse
     n2 = n / 2;
     for (i = 1; i < n - 1; i++) {
        n1 = n2;
        while (j >= n1) {
```

```
            j = j - n1;
            n1 = n1 / 2;
        }
        j = j + n1;
        if (i < j) {
            t1 = x[i];
            x[i] = x[j];
            x[j] = t1;
            t1 = y[i];
            y[i] = y[j];
            y[j] = t1;
        }
    }
    // FFT
    n1 = 0;
    n2 = 1;
    for (i = 0; i < m; i++) {
        n1 = n2;
        n2 = n2 + n2;
        a = 0;
        for (j = 0; j < n1; j++) {
            c = cos[a];
            s = sin[a];
            a += 1 << (m - i - 1);
            for (k = j; k < n; k = k + n2) {
                t1 = c * x[k + n1] - s * y[k + n1];
                t2 = s * x[k + n1] + c * y[k + n1];
                x[k + n1] = x[k] - t1;
                y[k + n1] = y[k] - t2;
                x[k] = x[k] + t1;
                y[k] = y[k] + t2;
            }
        }
    }
  }
}
```

Capítulo III: Aplicación móvil controlada por comando corto de voz

3.1. Propuesta de solución en Android Studio

El prototipo de aplicación desarrollada funciona de la siguiente manera: Se abre un reproductor de música que pueda correr en segundo plano, para ello se seleccionó la aplicación de música. Seguidamente, se abre la aplicación App para detener el audio por control con comando de voz. Se entiende por comando de voz por una sola palabra aislada, que este caso se seleccionó la palabra "oye".

Una vez abierta la aplicación, se puede observar un botón con el ícono de un micrófono y dos leyendas, la primera en la parte superior "Se requiere iniciar audio y audífonos conectados, no es necesario el internet" y en la parte inferior se presiona en el botón en el micrófono para iniciar el dictado de palabras. La palabra "oye" detendrá el audio.

Si se pulsa el botón al ícono de micrófono pueden suceder las siguientes situaciones:

a. Si no están los audífonos conectados, aparecerá una leyenda en la parte inferior "¡No están conectados los audífonos!".

b. Si están los audífonos conectados, pero el comando es diferente de "oye", no bajará el volumen del móvil.

c. Si están los audífonos conectados, pero el comando es "oye", bajará el volumen del móvil. Seguidamente, se podría reanudar el escuchar música con sólo subir el volumen del mismo.

Figura 17. Pantallas de la aplicación que muestran la respuesta como se describe en los incisos b y c.

3.2. FFT en Android Studio

El primer paso a observar es el muestreo. El tamaño del buffer de datos de muestreo permitido es 8000, entonces eso lleva a que la frecuencia máxima se debe establecer dividiendo el tamaño de la muestra entre 2, para cumplir con el teorema de muestreo de Nyquist. Se tratan los tramos para que sean en potencia de 2.

Para calcular la FFT se utilizó lo siguiente:

```
for (k=0;k<N;k++){

   //fft

   for (j=0;j<N;j++){
      xc = new Complex(x[j],0);
      F = new Complex(0,(2*Math.PI*(j-1)*(k-1))/N);
      Xc =xc.times(F.exp());
      Xre = Xre + Xc.re();
      Xim = Xim + Xc.im();

   }
```

```
            tograph[k] = Math.sqrt(Math.pow(Xre,2)+Math.pow(Xim,2));

        //reset Xim & Xre

        Xim=0;
        Xre=0;
    }
```

Para la manipulación del número complejo se utiliza una librería complex de "java.util.Objects"

3.3. Estructura de la aplicación de control de volumen de música con detención por comando de voz corto "OYE" utilizando interrupción por hilos de primer y segundo plano.

La estructura del programa de la aplicación, consiste primeramente en el archivo AndroidManifest.xml, para establecer la configuración, las actividades, permisos, servicios entre otros; el archivo MainActivity, en donde se coordinan las diferentes clases del programa. Incluye la clase MusicIntentReceiver, ServicioRecVoz, en java.

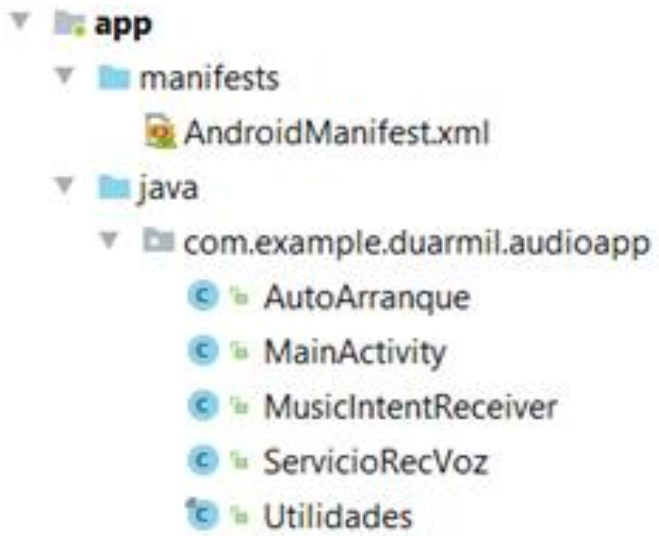


Figura 18. Estructura de la aplicación.

3.4. Desarrollo de un servicio de segundo plano

En el archivo AndroidManifest.xml, se declara el servicio a través de la instrucción <service android:name="ServicioRecVoz" /> y a través del arhcivo MainActivity, se controla el inicio con la instrucción <startService(service);> y su detención < stopService(service);>

El servicio de reconocimiento de voz se realizó en Java. Cuando ha detectado el comando "oye" envía el dato a la aplicación para que reduzca el volumen.

3.5. Manipulación de componentes

3.5.1. Volumen del móvil inteligente

En el archivo MainActivity de Java se declara el uso de la librería Android.media.AudioManager y con la instrucción "import static android.media.AudioManager.ADJUST_LOWER;", además la instrucción "myAudioManager.adjustVolume(ADJUST_LOWER,0); y la instrucción "myAudioManager.setStreamVolume(AudioManager.STREAM_MUSIC,maxMusicVolume, 0); se reduce el volumen.

3.1.3. Audífonos

Se verifica si están conectados los audífonos al móvil inteligente. Si no están conectados, no reducirá el volumen, aun cuando reconozca el comando "oye". En caso contrario, si detecta que los audífonos si están conectados, permitirá que se baje el volumen del audio hasta 0. Todo ello lo hace a través de la clase MusicIntentReceiver, realizado en Java, utilizando las siguientes instrucciones:

MusicIntentReceiver.java

```
package com.example.duarmil.audioapp;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;

public class MusicIntentReceiver extends BroadcastReceiver {

    public static int state;

    @Override
    public void onReceive(Context context, Intent intent){
        if (intent.getAction().equals(Intent.ACTION_HEADSET_PLUG)) {
            state = intent.getIntExtra("state", -1);
            switch (state) {
                case 0:
                    Utilidades.mostrarToastText(context, "Los audífonos NO están conectados!");
                    break;
                case 1:
                    Utilidades.mostrarToastText(context, "Los audífonos están conectados!");
                    break;
                default:
```

Capítulo IV: Comprobación del sistema y Resultados

4.1. Modelo de prueba

4.1.1. Modelo Estadístico para la validación del funcionamiento del sistema

Para comprobar el funcionamiento de la aplicación, se realizaron pruebas a la aplicación móvil. Se utilizó el teorema del límite central.

Dicho teorema afirma que, para una población con cualquier distribución, la distribución de las medias muestrales se aproxima a una distribución normal conforme aumenta el tamaño de la muestra. Así mismo, establece que si una muestra es lo bastante grande para un tamaño de n > 30. [22]

La falla es un evento que cambia el estado de un producto de operacional a no operacional. En este sentido la Tasa de Falla (TF) puede ser expresada como un número de fallas observadas en un tiempo de operación, por lo que tenemos: $TF = \frac{Número\ de\ fallas}{Número\ de\ Examinados}$ [23].

4.1.2. Metodología

Para la comprobación de la aplicación se utilizó una Tablet Samsung Galaxy Tab4, con Android 5.0.2, y con un celular Samsung Galaxy 4, Android 5. 0. 1.

El objetivo de esta prueba es medir la confiabilidad de la aplicación, esto es, que tiene la capacidad de reconocer el comando de voz “oye”, y con ello disminuye el volumen del reproductor de música del dispositivo.

La prueba toma los datos de edad y sexo. Se procede como se describe a continuación

Edad: _____ Sexo: F ( ) Masculino ( )

1. Se inicia el reproductor de música.

2. Se inicia la aplicación “Oye”.

3. Se dicta el comando “Oye”, si la aplicación reconoce el comando y disminuye el volumen, se termina el experimento, con una puntuación de 0 falla.

4. Se puede repetir hasta 3 veces el punto 3, de lo contrario se da un valor de 1 falla.

5. Se da un valor de 1 si el evento fue exitoso en la primer intento, se da un valor de 2 si el evento fue exitoso en el segundo intento y se da un valor de 3 si el evento fue exitoso en la tercer intento.

Se calculó el error de estimación en intervalo de confianza de 95%, dada una muestra grande, que en este caso fueron 35 muestras. Dicha muestra es una proporción de la población de intentos en n variables de Bernoulli aleatorias independientes con promedio común p. Así, $E(Y) = np$, o sea, $E\left(\frac{Y}{n}\right) = p$ y Y/n es un estimador insesgado de p.

El intervalo

$$\hat{p} \pm z_{\frac{\propto}{2}}\sqrt{\frac{\hat{p}}{n}(1-\hat{p})} \qquad (14)$$

Es la representación de un intervalo de confianza de muestra grande para p con coeficiente de confianza aproximadamente igual a $1-\propto$, donde $\hat{p} = y/n$, con un intervalo de confianza de 95%.[24]

4.1.3. Resultados

Número de muestra	Edad	Sexo	Número de Repeticiones	Resultado
1	21	M	1	1
2	21	M	1	1
3	19	F	1	1
4	19	M	1	1
5	20	M	2	1
6	19	M	1	1
7	19	M	1	1
8	19	M	1	1
9	48	F	1	1
10	20	M	1	1
11	20	M	1	1
12	23	M	1	1
13	21	F	1	1
14	22	F	1	1
15	29	F	1	1
16	53	F	1	1
17	21	M	1	1
18	21	F	1	1
19	22	M	1	1
20	20	M	1	1
21	23	M	1	1
22	21	F	2	1
23	21	F	1	1
24	21	F	1	1
25	21	M	1	1
26	20	M	1	1
27	22	M	2	1
28	22	M	1	1
29	22	M	1	1
30	23	M	1	1
31	22	M	1	1
32	21	M	1	1
33	24	M	1	1
34	21	M	1	1
35	21	M	1	1

Se comprobó con 35 personas, en rango de edades comprendidos entre los 19 y 53 años. De los cuales 9 del sexo femenino y 26 del sexo masculino.

Total de eventos exitosos en primer intento, 32 de 35, por lo que ocurrieron 3 fallas en primer intento.

Total de eventos exitosos en segundo intento, 3 de 3, por lo que ocurrieron 0 fallas.

$$TF(en\ primer\ intento) = \frac{Número\ de\ fallas}{Número\ de\ Examinados} = \frac{3}{35} = 0.08571429 = 8.5714\%$$

$$TF(en\ segundo\ \ intento) = \frac{Número\ de\ fallas}{Número\ de\ Examinados} = \frac{0}{35} = 0 = 0\%$$

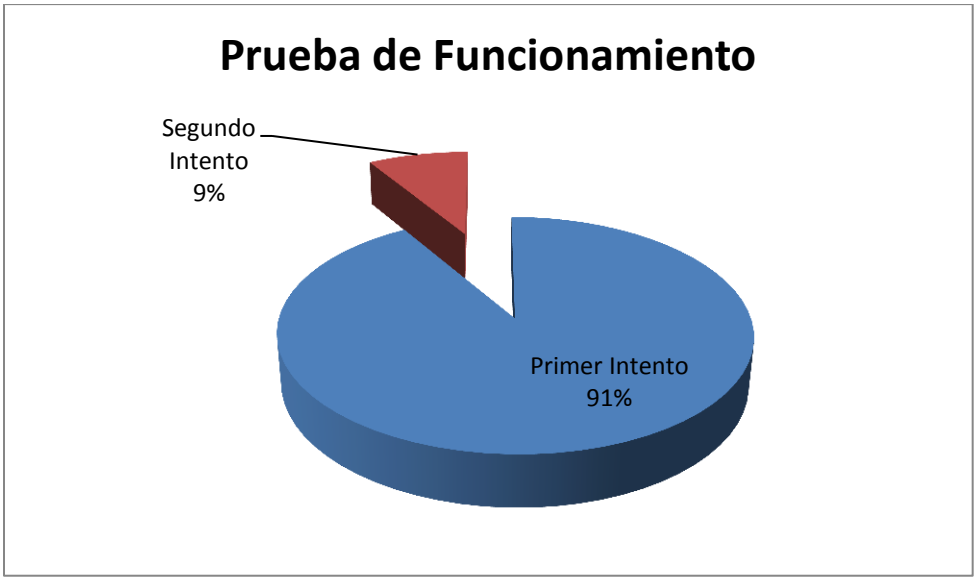


Figura 20. Resultados de la prueba de confiabilidad.

La prueba se realizó en un entorno controlado con bajo nivel de ruido, en el aula de clases en las instalaciones del Tecnológico Nacional de México, campus Instituto Tecnológico de Hermosillo.

El propósito de controlar por detección con comando de voz en una aplicación móvil sin internet, bajando el volumen del mismo, se logró con una tasa de falla del 8.57% en primer intento, y 0% en segundo intento. Lo que resulta en una confiabilidad de 91.48% en primer intento y 100% en segundo intento.

El intervalo de confianza del error de estimación se calculó de la siguiente forma

$\hat{p} \pm z_{\propto/2}\sqrt{\frac{\hat{p}}{n}(1-\hat{p})}$ = $(0.93) \pm 0.0987(\sqrt{\frac{0.93}{35}(1-0.93)})$ = 0.93 ± 0.0425

Intervalo $[0.9725, 0.8875]$

$\hat{p} = 0.93$

$n = 35$

$1-\propto = .95$

$\propto = 0.5$, $\frac{\alpha}{2} = 0.25$ lo que resulta de tabla $z_{\propto/2} = 0.0987$

La probabilidad de éxito en la estimación se representa con una confianza de $1-\propto$ que se estableció en 95% el nivel de confianza. Entonces $\propto$ resulta 0.5, y el error aleatorio o nivel de significación que representa las posibilidades de fallar en la estimación, que resulta en un intervalo de $[0.9725, 0.8875]$.

Resolviendo la pregunta planteada en la hipótesis de este proyecto, "¿Es posible desarrollar una aplicación que detecte fonemas específicos y controle el volumen de un reproductor de música en un dispositivo móvil inteligente? Se puede responder que si se presentó un prototipo de aplicación, la cual detecta un comando de voz y sí disminuye el volumen al nivel más bajo de un móvil inteligente, con una probabilidad de 5% de máximo error de la estimación o nivel de significancia, y un nivel de confianza de hasta 97% y mínimo de 89%.

Esta aplicación tiene que trabajar en conjunto con un servicio en segundo plano, para poder realizar el muestreo, la FFT y el posterior reconocimiento del comando, por lo que necesita ejecutarse en conjunto con el reproductor de música de la preferencia del usuario siempre y cuando pueda esta aplicación correr también en segundo plano, y no detenerse en cuanto empiece la ejecución de otra aplicación.

Esta condición se ha tornado difícil ya que la aplicación más popular, youtube, generó una aplicación “Music Key” para ello, y ya no es libre, esto es, que tiene un costo, y se vende a través de Google Play. Esta tendencia se ha expandido, y ya son pocas las aplicaciones que así lo hacen al día de hoy.[25] Cuando se inició el planteamiento de éste proyecto, dos años atrás, aún había este tipo de reproductores de música que corrían en segundo plano, pero hoy por hoy, esto está cambiando.

La prueba de confiabilidad se realizó con un reproductor que si puede ejecutarse en segundo plano, esto es, que no se detiene la música al iniciar otra aplicación. Se probó con el reproductor original del celular Samsung Galaxy 4, y en cuanto se presionaba el botón para dar el comando de voz, se paraba la música. No solo esta aplicación, sino también la denominada MusicAll, que podía estar trabajando otras aplicaciones al mismo tiempo, pero en cuanto se presionaba el botón para dictar comando, se detenía. Cabe mencionar que el pasado mes de mayo, todavía podía ejecutarse con el reproductor de música original del celular en mención y del celular LG g3.

Se afirma entonces, el objetivo general de éste trabajo, ya que se desarrolló una aplicación que procesa una señal de audio, que detecta fonemas específicos, y cuando identifica el comando “oye”, baja al más bajo nivel el volumen del móvil. Si detecta un comando diferente, sigue corriendo la música en el mismo nivel. Más aún, este comando es ejecutado solamente si los audífonos están conectados al dispositivo, de lo contrario, la música sigue en el mismo nivel de volumen.

Capítulo V: Conclusiones y trabajo futuro

El prototipo generado durante el desarrollo de esta tesis logra disminuir efectivamente el volumen del dispositivo móvil inteligente, que según la prueba de funcionabilidad aplicada, con un intervalo de error de $[0.9725, 0.8875]$, funciona en un 91.48%, en primer intento y 100% en segundo intento, utilizando un comando corto de voz "oye", independientemente del interlocutor.

Para la realización del comando coordinador del software fue indispensable y en general de todo el trabajo los cursos de electrónica digital, procesamiento digital de señales y matemáticas avanzadas.

La situación de riesgo, que por un lado se presenta con las personas que auditivamente se aíslan utilizando audífonos y música en alto volumen en sus dispositivos móviles, se puede resolver con el sistema propuesto en este trabajo de investigación ya que al proporcionar el comando corto de voz en consecuencia se disminuye en tiempo real el volumen del reproductor y sin la necesidad de acceder al uso de datos e internet.

Por otro lado, en cuanto al problema de accidentes en la vía pública se podría resolver de forma más contundente si en vez de reconocer un comando de voz, pudiera reconocer el claxon de un automóvil o del tren. Para ello también a través de la aplicación de la transformada rápida de Fourier, identificando la fundamental y los armónicos de dichos sonidos del entorno.

Una manera de mejorar el sistema actual sería, adaptar un módulo de configuración que de la opción para la selección de la palabra de comando de voz e inclusive en otros idiomas. Del mismo modo, otra idea de mejora futura sería modificar el comando que activa la disminución del volumen, para que en vez de reconocer voz, fuera el sonido de un claxon de tren o automóvil, vehículo de emergencia, entre otros, utilizando la misma transformada rápida de Fourier y con esto la posibilidad de disminuir el riesgo de accidentes viales que involucra el uso de audífonos.

Bibliografía

[1] J. F. James, *A Student's Guide to Fourier Transforms*, Third edit. New York: Cambridge University Press, 2011.

[2] J. Bobadilla, "La transformada de Fourier. Una visión pedagógica," *Estud. fonética Exp.*, vol. 10, pp. 41–74, 1999.

[3] J. Bernal, P. Gómez, and J. Bobadilla, "UNA VISIÓN PRÁCTICA EN EL USO DE LA TRANSFORMADA DE FOURIER COMO HERRAMIENTA PARA EL ANÁLISIS ESPECTRAL DE LA VOZ," *Estudios de fonética experimental X*, Barcelona, España, 1999.

[4] D. H. Mora, E. Huerta, and D. E. L. A. Fuente, "' Sintesis De Voz ' Y ' Filtrado Digital '.," Universidad Autónoma Metropolitana, 1997.

[5] H. H. A. Bonnet Alfonso Jorge, Gutierrez José Alfonso, "Reconocedor Automático de Comandos por medio del Habla para las Funciones de un Automovil," 2013. [Online]. Available: http://itzamna.bnct.ipn.mx/dspace/bitstream/123456789/12216/1/reconocedorautomatico.pdf. [Accessed: 29-Dec-2015].

[6] L. Gaceta, "Audífonos en los jóvenes: un problema de uso o de mal uso - La Gaceta," *La Gac.*, vol. 6727, 2016.

[7] M. P. (CNN), "Usar audífonos al caminar triplica el riesgo de lesiones en EU - Salud - CNNMéxico.com," *17 de Enero de 2012*. [Online]. Available:

http://mexico.cnn.com/salud/2012/01/17/usar-audifonos-al-caminar-triplica-el-riesgo-de-lesiones-en-eu. [Accessed: 29-Dec-2015].

[8] R. Lichenstein, D. C. Smith, J. L. Ambrose, and L. A. Moody, "Headphone use and pedestrian injury and death in the United States: 2004-2011.," *Inj. Prev.*, vol. 18, no. 5, pp. 287–90, Oct. 2012.

[9] I. M. Torres Hernandez, "Framework multiplataforma para reconocimiento de voz en aplicaciones open rich-client para dispositivos moviles"," Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, 2013.

[10] C. H. Basch, D. Ethan, S. Rajan, and C. E. Basch, "Technology-related distracted walking behaviours in Manhattan's most dangerous intersections.," *Inj. Prev.*, vol. 20, no. 5, pp. 343–6, Oct. 2014.

[11] D. H. Mora, E. Huerta, and D. E. L. A. Fuente, "Proyecto Terminal Electr ~ Nica Tema : ' Sintesis De Voz ' Y ' Filtrado Digital '.," 1997.

[12] L. Rabiner, "Automatic Speech Recognition – A Brief History of the Technology Development," 2004. [Online]. Available: http://www.ece.ucsb.edu/Faculty/Rabiner/ece259/Reprints/354_LALI-ASRHistory-final-10-8.pdf. [Accessed: 05-Jan-2016].

[13] "PhoneGap - EcuRed." [Online]. Available: http://www.ecured.cu/PhoneGap. [Accessed: 06-Jan-2016].

[14] I. M. Torres Hernández, "Framework multiplataforma para reconocimiento de voz en aplicaciones open rich-client para dispositivos mviles," Centro de Investigación y

de Estudios Avanzados del Instituto Politécnico Nacional, 2013.

[15] A. Larios, “Sistemas de reconocimiento y síntesis de voz,” Universidad de las Americas, 1999.

[16] V. J. A. Pérez Eyra Oxana, Poceros Fernando, “Sistema de Seguridad Por Reconocimiento de Voz,” Instituto Politécnico Nacional, 2013.

[17] B. Soediono, “RECONOCIMIENTO AUTOMÁTICO DEL HABLA UTILIZANDO LA TRANSFORMADA DE FOURIER Y REDES NEURONALES,” *J. Chem. Inf. Model.*, vol. 53, p. 160, 1989.

[18] L. Rabiner and B.-H. Juang, “Fundamentals of Speech Recognition,” *Prentice Hall*, vol. 103. p. 507, 1993.

[19] M. J. Roberts, *Señales y Sistemas*, Primera Ed. McGraw-Hill Interamericana, 2004.

[20] W. G. Van Tasell DJ1, Soli SD, Kirby VM, “Speech waveform envelope cues for consonant recognition,” *PubMed*, 1987. [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/3680774.

[21] K. S. Fowler Martin, *UML gota a gota*. Pearson Educacion, 1999.

[22] M. F. Triola, *Actualización Tecnologica Estadística*, Decimoprim. Pearson Educación de México, S. A. de C. V., 2014.

[23] G. T. en Mantenimiento, “Tasa de Falla y Tiempo Medio entre Fallas (MTBF),” 2015. [Online]. Available: https://www.gestiondeoperaciones.net/mantenimiento/tasa-de-falla-y-tiempo-medio-

entre-fallas-mtbf/. [Accessed: 15-Nov-2017].

[24] S. McClave, *Probabilidad y estadistica para ingeniería*. México, D. F.: Grupo Editorial Iberoamericana, S. A. de C. V., 1995.

[25] A. Laura, "El androide libre," 2015. [Online]. Available: https://elandroidelibre.elespanol.com/2015/01/las-mejores-aplicaciones-para-escuchar-youtube-en-segundo-plano.html. [Accessed: 29-Nov-2017].