



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERÍA

**SISTEMA DE ALARMA RESIDENCIAL EMPLEANDO
VOZ SINTÉTICA VÍA TELEFÓNICA**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE:
INGENIERO ELÉCTRICO Y ELECTRÓNICO**

P R E S E N T A :

LUIS ALBERTO ÁLVAREZ FERNÁNDEZ



**DIRECTOR DE TESIS:
M. I. LARRY HIPÓLITO ESCOBAR SALGUERO**

MÉXICO, D.F.

2014



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

AGRADECIMIENTOS

A mis padres por el apoyo a lo largo de los años y por alentarme siempre a ser una mejor persona y por enseñarme a esforzarme para cumplir las metas que me proponga. A mis hermanos por estar siempre conmigo animándome y alentándome a seguir adelante.

A mis amigos, con los que he aprendido acerca de la vida y con los que compartí gran parte de mis estudios.

Al M.I. Larry Escobar por el apoyo y la paciencia durante la elaboración de este trabajo, y por las enseñanzas que me ha brindado.

Resumen

En el presente trabajo se describe la implementación de un sistema de alarma residencial, el cual al detectar la apertura de puertas o ventanas realiza una llamada telefónica al teléfono celular del dueño de la vivienda para notificarle, mediante el uso de voz sintética, acerca del sensor que se ha activado, así mismo el sistema es capaz de detectar si se accionó el interfón y en este caso, el sistema realiza la llamada telefónica y permite que la persona que tocó pueda establecer comunicación con el dueño de la vivienda. El sistema se realizó empleando una tarjeta de desarrollo (DSK) la cual contiene el procesador digital de señales (DSP) TMS320C5402 de *Texas Instruments (TI)*, la síntesis de la señal de voz se realiza empleando el DSP y se emplearon los periféricos de la tarjeta de evaluación para realizar la conexión de los sensores así como de la línea telefónica. El sistema debe ser capaz de funcionar en tiempo real para dar aviso de manera inmediata acerca de la intrusión, y que de esta manera el usuario pueda realizar las acciones necesarias para evitar daños o pérdidas en su vivienda.

Índice general

1. INTRODUCCIÓN	1
2. SÍNTESIS DE VOZ	5
2.1 La señal de voz	5
2.1.1 Producción de voz	5
2.1.2 Clasificación de los sonidos de la voz	6
2.1.3 Representación digital de la señal de voz	7
2.2 Vocoder LPC	9
2.3 Análisis de la señal de voz	10
2.3.1 Proceso de ventaneo	10
2.3.2 Umbral de silencio	12
2.3.3 Filtro de preénfasis	13
2.3.4 Estimación de la naturaleza de la señal	14
2.3.5 Obtención de los coeficientes LPC empleando autocorrelación	18
2.3.6 Ganancia del segmento	21
2.4 Síntesis de voz	21
2.4.1 Filtro todo polo	22
2.4.2 Filtro de deénfasis	22
2.5 Resumen	24
3. COMUNICACIÓN TELEFÓNICA	25
3.1 Sistema telefónico	25
3.2 Circuito telefónico	26
3.3 Medios de transmisión	27
3.3.1 Transmisión de voz	29
3.4 Conmutadores	30
3.5 Señales de control	32
3.6 Tonos duales de multifrecuencia (DTMF)	33
3.6.1 Generación de tonos	34
3.7 Resumen	38
4. ANÁLISIS Y DISEÑO DEL SISTEMA	39
4.1 Descripción general del sistema	39
4.2 DSK TMS430C5402	41
4.2.1 Características generales DSP TMS320C5402	45

4.3	Hardware	48
4.4	Software	49
4.4.1	Análisis fuera de línea	49
4.4.2	Procesamiento en tiempo real	51
4.5	Resumen	55
5.	PRUEBAS Y RESULTADOS	57
5.1	Evaluación de la voz generada	57
5.2	Evaluación de los tonos DTMF generados	62
5.3	Evaluación del hardware	64
5.4	Resumen	65
6.	CONCLUSIONES	67
•	BIBLIOGRAFÍA	69
•	ANEXOS	
○	Anexo A: Código en Matlab® para el análisis fuera de línea	71
○	Anexo B: Código en C para el DSP	75

Índice de figuras

1.1.	Esquema general del sistema	2
1.2.	Esquema de síntesis de voz	3
2.1	Aparato productor de voz	6
2.2	Ejemplos de segmentos de voz a)voceados b) no voceados	7
2.3	Representación de las señales de voz	7
2.4	Modelo general de producción de voz	8
2.5	Ventana Hamming en el dominio temporal y frecuencial	11
2.6	Segmento de voz aplicándole una ventana Hamming con $N=256$	12
2.7	Envolvente típica de una señal de voz	13
2.8	Filtro de preénfasis	14
2.9	Función y(n) de recorte central	15
2.10	Ejemplo de recorte central de un segmento de voz	16
2.11	Autocorrelación de segmento voceado y tren de impulsos	17
2.12	Autocorrelación de segmento no voceado	17
2.13	Filtro de predicción	18
2.14	Espectro de la señal y salida del filtro todo polo $p= 5$	20
2.15	Espectro de la señal y salida del filtro todo polo $p= 10$	20
2.16	Espectro de la señal y salida del filtro todo polo $p= 20$	21
2.17	Filtro de deénfasis	23
2.18	Ejemplo de una señal de voz original y sintética	23
2.19	Espectro de una señal de voz original (negra) y sintética (azul).....	24
3.1	Circuito simplificado del aparato telefónico (analógico)	27
3.2	Cable coaxial	28
3.3	Transmisión empleando ondas de radio	28
3.4	Transmisión empleando satélites	29
3.5	Ancho de banda de la señal de voz.....	30
3.6	Conexión manual entre líneas	31
3.7	Sistema de conmutación telefónico	31
3.8	Timbre del teléfono	32
3.9	Matriz de frecuencias del teclado telefónico	33
3.10	Ejemplo de una señal DTMF en el dominio temporal y frecuencial	34
3.11	Diagrama de bloques de un generador DTMF	35
3.12	Filtro IIR para generar señales senoidales	36
3.13	Respuesta en frecuencia del filtro IIR	37
3.14	Ejemplo de una señal senoidal empleando un oscilador recursivo	38

4.1	Diagrama de bloques del sistema	39
4.2	Diagrama de flujo del funcionamiento.....	40
4.3	Diagrama de la vivienda	41
4.4	Sistema de desarrollo TMS320C5402 DSK	42
4.5	Interfaz DAA	43
4.6	Códec para interfaz DAA	43
4.7	Circuito de micrófono y bocina	44
4.8	Diagrama a bloques del TMS320C5402	47
4.9	Interfaz de expansión de memoria y periféricos	48
4.10	Diagrama de conexión de los interruptores	49
4.11	Diagrama de flujo del programa de MATLAB®	50
4.12	Diagrama de flujo de la generación de tonos DTMF	52
4.13	Diagrama de flujo de la recuperación de datos	53
4.14	Diagrama de flujo de la síntesis de voz	54
5.1	Señal a)Original, b)Sintética, c)Espectro original, d)Espectro sintética, N=512, p=10	58
5.2	Señal a)Original, b)Sintética, c)Espectro original, d)Espectro sintética, N=256, p=10	58
5.3	Señal a)Original, b)Sintética, c)Espectro original, d)Espectro sintética, N=128, p=10	59
5.4	Señal a)Original, b)Sintética, c)Espectro original, d)Espectro sintética, N=128, p=5	60
5.5	Señal a)Original, b)Sintética, c)Espectro original, d)Espectro sintética, N=128, p=10	60
5.6	Señal a)Original, b)Sintética, c)Espectro original, d)Espectro sintética, N=512, p=20	61
5.7	Espectrograma de la señal de voz “principal” a)Original, b)Sintética, N=128, p=10	62
5.8	Tono DTMF correspondiente al número 1	63
5.9	Espectro de tono DTMF correspondiente al número 1	63
5.10	Ejemplo de varios tonos DTMF, correspondiente a los dígitos 1, 2 y 3	64

Glosario y abreviaturas

A

ai: coeficientes del filtro todo polo.

ALU: unidad aritmética lógica (*Arithmetic Logic Unit*).

C

CCSU: unidad de comparación, selección y almacenamiento (*Compare, Select, and Store Unit*).

D

DAA: arreglo de acceso a datos (*Data Access Arrangement*).

DFT: transformada discreta de Fourier (*Discrete Fourier Transform*).

DMA: canal de acceso directo a memoria (*Direct Memmory Access*).

DSK: DSP starter kit.

DSP: procesador digital de señales.

DTMF: tonos duales de multifrecuencia (*Dual Tone Multi Frequency*).

F

FFT: transformada rápida de Fourier (*Fast Fourier Transform*).

H

HPI8: interfaz de puerto paralelo tipo huésped de 8 bits (*Enhanced 8 bit Parallel Host Port Interface*).

L

LPC: codificación por predicción lineal.

M

MAC: multiplicación acumulación.

MATLAB: laboratorio de matrices, herramienta de software matemático.

McBSP: puerto serial bufereado multicanal (*Multichannel Buffered Serial Port*).

MIPS: millones de instrucciones por segundo.

N

NC: interruptor normalmente cerrado (*Normally Close*).

P

Pipeline: técnica de procesamiento en paralelo empleada en los DSP, que consiste en dividir una instrucción en varias etapas de tal manera que otra instrucción pueda ser inicializada antes de ser terminada la primera.

Pitch: frecuencia a la que vibran las cuerdas vocales.

PLL: circuito de malla de fase amarrada (*Phase Locked Loop*).

R

DARAM: RAM de doble acceso (*Dual Access RAM*).

RJ11: interfaz física usada para conectar redes de teléfonos.

ROM: memoria de solo lectura (*Read Only Memory*).

S

SRAM: memoria RAM estática (*Static Random Access Memory*).

V

Vocoder: codificador de voz.

Capítulo 1

INTRODUCCIÓN

En la actualidad es necesario conocer el estado de nuestra casa o residencia de forma remota, por lo que se requiere de sistemas que nos proporcionen dicha información de manera práctica, una forma de realizar esto es transmitiendo los datos requeridos mediante la línea telefónica a un teléfono celular, lo cual permite que el sistema se comuniquen con el usuario sin importar el lugar donde este se encuentre.

Las alarmas están presentes cada día más en nuestras vidas, debido a la inseguridad actual y, por consiguiente, la necesidad de tener mayor protección ya sea en nuestros hogares, oficinas, etc. Hace algunos años los sistemas de seguridad solo se encontraban en lugares muy concretos, pero hoy en día se encuentran en hogares, fábricas, bancos, tiendas, etc. Las alarmas no son nuevas, estas se han utilizado desde hace más de 5000 años para advertir a tribus o individuos sobre algún peligro o sobre la llegada de individuos extraños, algunos utilizaban señales de humo, tambores, cuernos de animales, o cualquier instrumento que produjera sonido [1].

Se puede definir un sistema de seguridad como el "conjunto de elementos e instalaciones necesarios para proporcionar a las personas y bienes materiales existentes en un local determinado, protección frente a agresiones, tales como robo, atraco o sabotaje e incendio" [2].

Todos los sistemas de seguridad tienen tres componentes básicos en común [2]:

1. Detección/Sensores: son los elementos que detectan los eventos de interés en un lugar determinado y envían dicha información a la unidad de control.
2. Unidad de control: recibe las diferentes señales de los sensores y pone en marcha los sistemas de alarma y aviso.
3. Aviso/Alarma: son los dispositivos encargados de avisar las variaciones detectadas por los sensores dentro del sistema de seguridad.

En caso de un siniestro, en principio el sistema lo detectará, luego lo señalará, para posteriormente iniciar las acciones encaminadas a disminuir o extinguir los efectos [2]. Así mismo existen diferentes sistemas que dan aviso de la intrusión, en el caso de este proyecto se seleccionó un sistema de aviso a distancia mediante el teléfono y debido a que el sistema debe comunicarse con el usuario, es necesario que esta comunicación se realice de manera eficiente y cómoda para el usuario, para esto se propone el uso de voz sintética, la cual es una forma práctica de comunicar el mensaje respectivo debido a que estamos acostumbrados a comunicarnos por señales de voz.

En el presente trabajo se implementó un sistema de seguridad que nos permite conocer si en algún momento se ha accedido a la casa, mediante la apertura de puertas o ventanas, de esta manera podemos ser alertados en caso de algún intento de robo.

Por lo que el **objetivo** de este trabajo es:

Diseñar un sistema de alarma, para casa habitación el cual debe ser capaz de detectar la apertura de puertas y/o ventanas para informar al usuario mediante voz sintética a través de una llamada telefónica, así mismo en caso de que se active el timbre el sistema deberá establecer comunicación entre la persona que tocó y el usuario.

El sistema diseñado consiste de una tarjeta de desarrollo, la cual contiene un procesador digital de señales (DSP), que se encarga de realizar una llamada telefónica, a un número almacenado en memoria y genera los diferentes mensajes de voz para poder notificar el evento que se ha presentado, como se muestra en la figura 1.1.

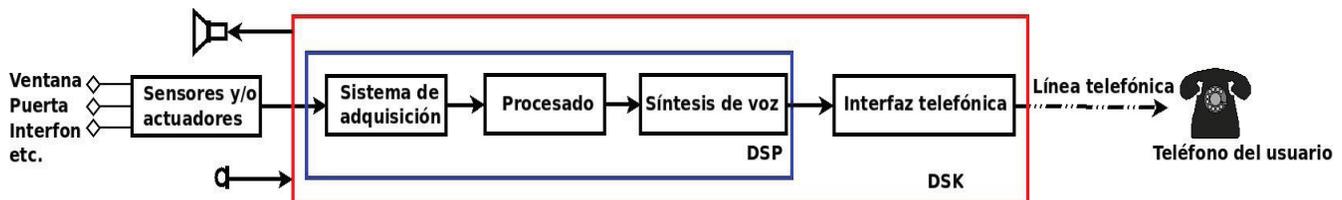


Figura 1.1: Esquema general del sistema.

Ya que el sistema establece comunicación con el usuario mediante la línea telefónica, es necesario considerar los mensajes que van a ser transmitidos, y debido a que las señales de voz se muestrean a 8 KHz, la cantidad de memoria necesaria para poder almacenar estas señales es bastante grande, por lo que es necesario reducir la cantidad de datos de esta señal, la técnica de síntesis empleada, permite reducir la información de la señal de voz hasta un 5% o incluso menos, gracias a esto es posible generar los diferentes mensajes sin la necesidad de requerir una memoria externa en el sistema, lo que ahorra gastos en los componentes y espacio.

El sistema se implementó usando una tarjeta de desarrollo DSK5402, la cual cuenta con una interfaz RJ-11 para conexión telefónica, en el DSP se encuentran las diferentes rutinas para detección de eventos, marcación telefónica y síntesis de voz.

Con esta tarjeta de desarrollo, la única conexión extra necesaria sería la de los sensores, los cuales están conectados a los pines de interrupción externa del DSP, hasta que se presente alguna señal proveniente de los sensores.

Para realizar la síntesis de voz se parte de un análisis fuera de línea en el cual se extraen parámetros de dicha señal, estos parámetros son almacenados en la memoria del DSP, una vez que se presenta algún evento, el procesador genera la señal de voz con los parámetros

previamente almacenados, así mismo realiza el proceso de descolgado y marcado en la línea telefónica para posteriormente enviar los datos de la señal de voz sintética.

La obtención de parámetros de la señal de voz se realizó mediante un script de MATLAB, en el cual se programó un vocoder por codificación de predicción lineal (LPC), dicha predicción se realizó con el método de autocorrelación. Con este procedimiento se pueden obtener los coeficientes "ai" de la señal de voz, así como la naturaleza de dicha señal (voceada, no voceada) y el periodo de pitch de la señal en caso de que esta sea una señal voceada, como se ve en la figura 2.

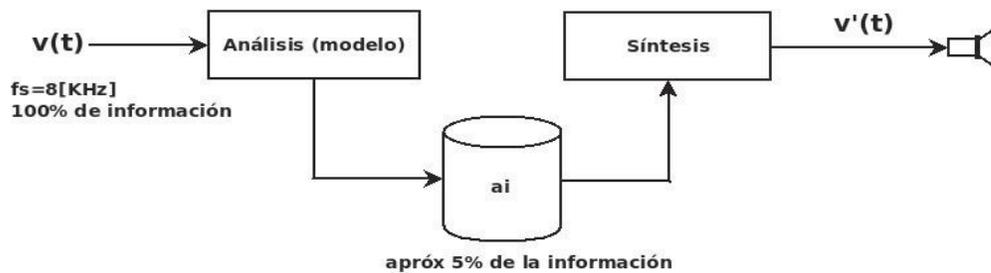


Figura 1.2: Esquema de síntesis de voz.

La síntesis de voz se realizó mediante un filtro todo polo cuyos coeficientes son los términos "ai" y se alimenta con un tren de pulsos separados por el periodo de pitch en caso de ser un segmento voceado; en caso de ser una segmento no voceado el filtro se alimenta con ruido blanco.

Para establecer la comunicación mediante la línea telefónica se deben realizar las siguientes acciones [3]:

1. Levantar el auricular.
2. Detección de un tono llamado "disca".
3. Envía al sistema el número telefónico a llamar.
4. Detección de tonos que indican el estado de la llamada (llamando, ocupado, etc.)
5. Notificar una llamada entrante al teléfono.
6. Convierte el lenguaje de una persona en señales eléctricas para su transmisión.
7. Convierte la señales eléctricas recibidas en audio.
8. Ajusta los cambios en la fuente de alimentación que recibe.
9. Notifica al sistema que una llamada ha terminado al colgar el auricular.

Este sistema es capaz de realizar la mayoría de las funciones antes mencionadas, de tal manera que logra establecer la comunicación con el usuario.

Este trabajo está organizado de la siguiente manera, en el *capítulo dos* se estudian las características de la señal de voz, así como el análisis de esta para poder obtener los parámetros que la caracterizan y poder realizar la síntesis de esta. En el *capítulo tres* se analizan el funcionamiento del sistema telefónico, así como de las señales que en éste se emplean. El diseño del sistema se explica en el *capítulo cuatro*, aquí se agregan los circuitos empleados así como diagramas de flujo del programa realizado. Los resultados se muestran en el *capítulo cinco* y en el *capítulo seis* se exponen las conclusiones.

Capítulo 2

Síntesis de voz

El habla ha sido utilizada desde hace miles de años como el método principal para transmitir información, por lo que resulta natural pensar que los sistemas utilicen voz para brindarnos información, esto ha sido posible desde que se desarrollaron los sistemas de grabación [4].

La síntesis de voz consiste en la producción del habla humana de manera artificial, es un área de estudio dentro del procesamiento digital de señales, la cual se encarga del estudio de las señales de voz, algunas de las aplicaciones del procesamiento digital de señales de voz son [5]:

1. Transmisión y almacenamiento de la señales.
2. Síntesis del habla.
3. Reconocimiento de voz.
4. Reconocimiento del hablante.
5. Ayuda para los discapacitados.
6. Aumentar la calidad de la señal de voz.
7. Reducir el volumen de información de la señal.

En este capítulo se presentan los conceptos básicos de la señal de voz, sus características, así como la manera de extraer los parámetros de esta señal para posteriormente poder realizar la síntesis y recuperar la información contenida en dicha señal.

2.1 LA SEÑAL DE VOZ

La voz está compuesta por una secuencia de sonidos, estos sonidos y la transición entre ellos sirven como una representación simbólica de la información [5].

El sonido no es más que la propagación de una onda longitudinal, generalmente, a través del aire; para que el sonido sea producido es necesaria la vibración de alguna fuente la cual comprima y expanda las moléculas del medio por el cual se va a propagar la onda.

2.1.1 Producción de voz

El sistema de producción de la voz está constituido por los pulmones, la traquea, las cuerdas vocales, la laringe, faringe y las cavidades bucal (oral) y nasal. Los diferentes sonidos producidos se dan por el movimiento de dichos órganos, así como del movimiento de una serie de elementos articulatorios como son los dientes, los labios, la lengua, etcétera, como se muestran en la figura 2.1.

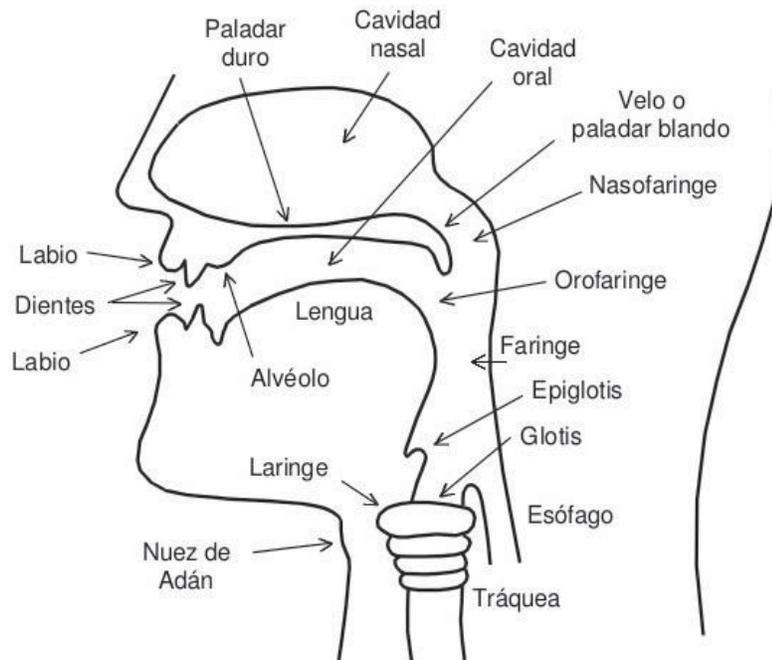


Figura 2.1: Aparato productor de voz [4].

Tanto la cavidad bucal y nasal, también conocidos como tracto vocal y tracto nasal, respectivamente, pueden ser vistos como tubos de sección transversal no uniforme con frecuencias de resonancia diferentes, a las frecuencias de resonancia del tracto vocal se le conocen como *formantes*. Estas frecuencias dependen del área del tracto vocal, pero su ancho de banda depende de las pérdidas en el tracto vocal (formantes a frecuencias bajas), de la fricción viscosa, las pérdidas térmicas del tracto vocal y de las pérdidas de radiación (formantes a frecuencias altas) [5].

2.1.2 Clasificación de los sonidos de la voz

Los sonidos de la voz pueden ser clasificados de acuerdo a su modo de excitación en *voceados* y *no voceados*, al combinar estos sonidos se pueden generar otros, como son: sonidos plosivos, mezclados, susurros, etc [6], a pesar de que la simplificación de los sonidos en voceados y no voceados parece una simplificación excesiva, esta es aceptada de forma generalizada [7].

Los sonidos *voceados* son producidos por la oscilación de las cuerdas vocales generando pulsos cuasi - periódicos los cuales excitan el tracto vocal, estos pulsos tienen una frecuencia fundamental llamada frecuencia de pitch.

Los sonidos *no voceados* son generados por la turbulencia del aire al pasar por el tracto vocal, el cual está constreñido en algunos puntos, por lo que este tipo de sonidos tienen una

naturaleza ruidosa.

En la figura 2.2 se muestran segmentos de voz voceados y no voceados, así como el espectro de cada uno, se puede apreciar que los sonidos voceados presentan cierta periodicidad la cual no existe en los sonidos no voceados.

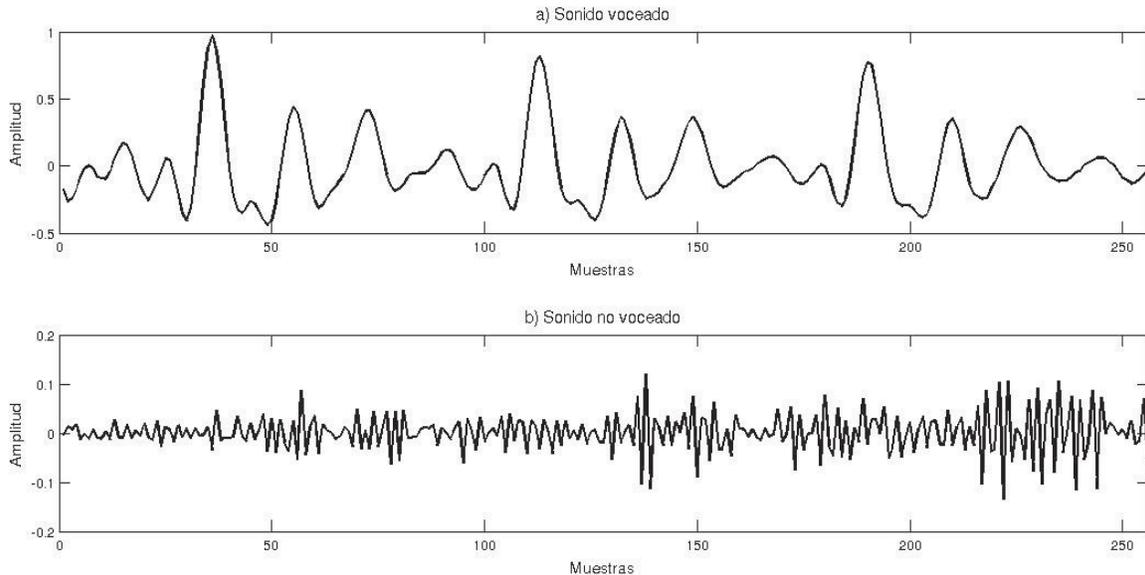


Figura 2.2: Ejemplos de segmentos de voz a) voceados b) no voceados

2.1.3 Representación digital de la señal de voz

Existen muchas maneras de clasificar las diferentes formas de representar las señales de voz, las cuales deben garantizar que la información contenida en la señal de voz se pueda extraer fácilmente por el oyente o por una máquina [5], estas clasificaciones se pueden agrupar como se muestra en la figura 2.3.

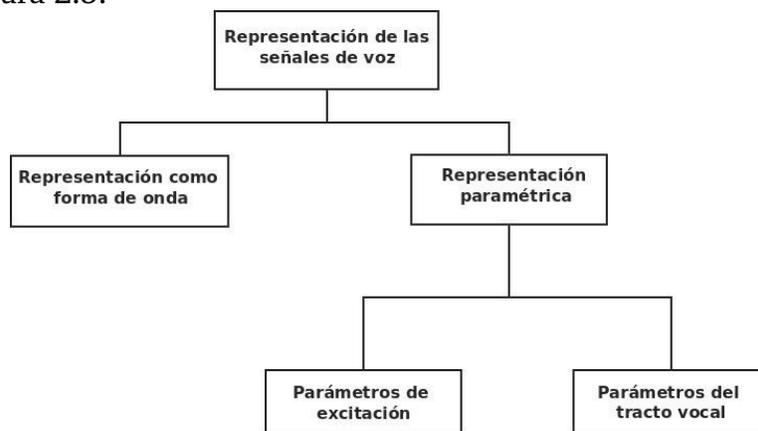


Figura 2.3: Representación de las señales de voz [5].

La representación como forma de onda se basa en preservar la forma de la señal analógica, mientras que las representaciones paramétricas se basan en representar la señal como la salida de un modelo de producción de voz; la base de estos métodos es el uso de la redundancia inherente a la señal de voz para reducir la cantidad de información, ya que una teoría acústica detallada debe considerar [5]:

- a) La variación en el tiempo del tracto vocal.
- b) Pérdidas debido a la conducción y a la fricción viscosa en las paredes del tracto vocal.
- c) La suavidad de las paredes del tracto vocal.
- d) La radiación del sonido debida a los labios.
- e) El acoplamiento nasal.
- f) La excitación del sonido en el tracto vocal.

En la figura 2.4 se muestra el modelo terminal – análogo, este modelo representa la producción de voz en función de la señal deseada [8].

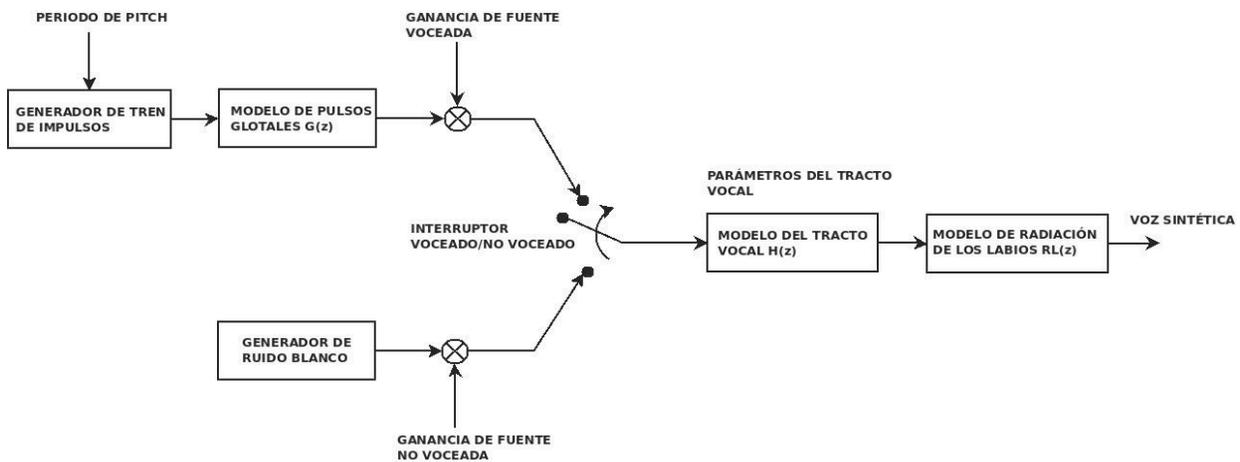


Figura 2.4. Modelo general de producción de voz.

Este modelo consta de un modelo del tracto vocal $H(z)$ y un modelo de radiación de los labios $RL(z)$, el cual puede ser excitado por un tren de impulsos separados por un periodo igual al periodo de pitch, en el caso de que se determine que la señal es voceada, o por ruido blanco en caso de se determine que es no voceada.

La función de transferencia del modelo del tracto vocal se muestra en la ecuación (2.1):

$$H(z) = \frac{s(z)}{u(z)} = \frac{G}{1 + \sum_{k=1}^N a_k z^{-k}} \quad (2.1)$$

donde G representa la ganancia global y a_k son el conjunto de coeficientes del filtro todo polo, los cuales producen una forma de onda similar a la envolvente de la señal.

El modelo de radiación de los labios $RL(z)$ se conoce como filtro de deénfasis que se analizará más adelante en el capítulo.

2.2 VOCODER LPC

Como se mencionó anteriormente las técnicas de representación paramétrica pretenden reducir el volumen de información necesario para reconstruir la señal de voz manteniendo la inteligibilidad del mensaje, para esto se hace uso de los vocoders, los cuales son codificadores de voz, existen muchos tipos de vocoders, entre los más conocidos se encuentran:

1. Vocoder Channel.
2. Vocoder de Fase.
3. Vocoder de Formantes.
4. Vocoder por codificación de predicción lineal (LPC).
5. Vocoder Cepstral.

En el presente trabajo se hizo uso de un vocoder LPC por ser el más utilizado debido a su simplicidad de implantación, el cual consiste en calcular una muestra de voz a partir de una combinación lineal de muestras pasadas; este vocoder se basa en el modelo terminal – análogo de la figura 2.4.

Este vocoder consta de un bloque codificador LPC y un bloque decodificador LPC. El codificador LPC se encarga de realizar el análisis de la señal de voz para obtener los parámetros LPC, el decodificador hace uso de los parámetros LPC para realizar la síntesis de la señal de voz.

Para poder realizar el análisis de la señal de voz es necesario segmentar dicha señal en tramas, las cuales serán analizadas de manera individual, el proceso de análisis de la señal de voz consta de los siguientes pasos [3]:

1. Determinar que la señal no sea una trama de silencio.
2. La señal se pasa por un filtro de preénfasis.
3. Se determina si la trama corresponde a una señal voceada o no voceada.
4. En caso de que sea una señal voceada se obtiene el periodo de pitch.
5. Se obtienen los coeficientes de predicción lineal.
6. Se calcula la ganancia del filtro todo polo.
7. Se almacenan estos parámetros.

En el proceso de síntesis de voz para cada trama se requiere [8]:

1. Obtención de los parámetros almacenados.
2. Se verifica si la señal es silencio.
3. En caso de no ser silencio se determina si es voceada o no voceada.
4. En caso de ser voceada se genera un tren de impulsos, con el periodo de pitch correspondiente, en caso de ser no voceada se genera ruido blanco.
5. Se utiliza la ganancia del segmento.
6. Se recuperan los coeficientes de predicción lineal que definen el filtro todo polo.
7. La señal se pasa por un filtro de deénfasis.

Tanto los pasos para realizar el análisis de la señal de voz como para la síntesis de voz serán desarrollados en la sección 2.3 y 2.4, respectivamente.

2.3 ANÁLISIS DE LA SEÑAL DE VOZ

Como se mencionó en la sección anterior el vocoder LPC consta de un codificador, el cual se encarga de obtener los parámetros que representan la señal de voz, y que se almacenaran para posteriormente reconstruir esta señal por medio del decodificador LPC y de esta manera obtener voz sintética capaz de transmitir el mensaje original.

Antes de poder obtener los parámetros de la señal es necesario segmentarla empleando ventanas, ya que la voz es una señal aleatoria por lo que los valores de esta señal no pueden ser determinados mediante una función matemática, para poder analizarla es necesario considerar que las propiedades de la señal de voz varían “lentamente”, esto se logra al tomar segmentos de 10 a 30 ms de la señal, con lo que se puede garantizar que la señal cumpla con las condiciones de estacionaridad y ergódicidad de tal manera que se puedan estimar los parámetros del modelo todo polo [8].

2.3.1 Proceso de ventaneo

Las ventanas son funciones que limitan el estudio de una señal a un número determinado de muestras segmentando la señal original.

Al multiplicar una señal por una ventana se obtiene el efecto de segmentación, en la cual el valor de la señal fuera de la ventana es de cero. Es muy importante elegir la forma de la ventana, ya que si la ventana tiene un corte brusco ésta puede generar componentes de alta frecuencia en la señal resultante. Existen muchos tipos de ventanas, las más comunes son [5] [8]:

- Rectangular.
- Bartlett.
- Hanning.
- Hamming.

- Blackman.
- Kaiser.

La ventana más utilizada dentro del procesamiento de voz, debido a sus propiedades espectrales, es la ventana Hamming [9], la cual se define en la ecuación (2.2).

$$\omega(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right) \quad (2.2)$$

Donde N es el tamaño de la ventana. En la figura 2.5 se muestra la ventana Hamming en el dominio del tiempo y en el dominio de la frecuencia.

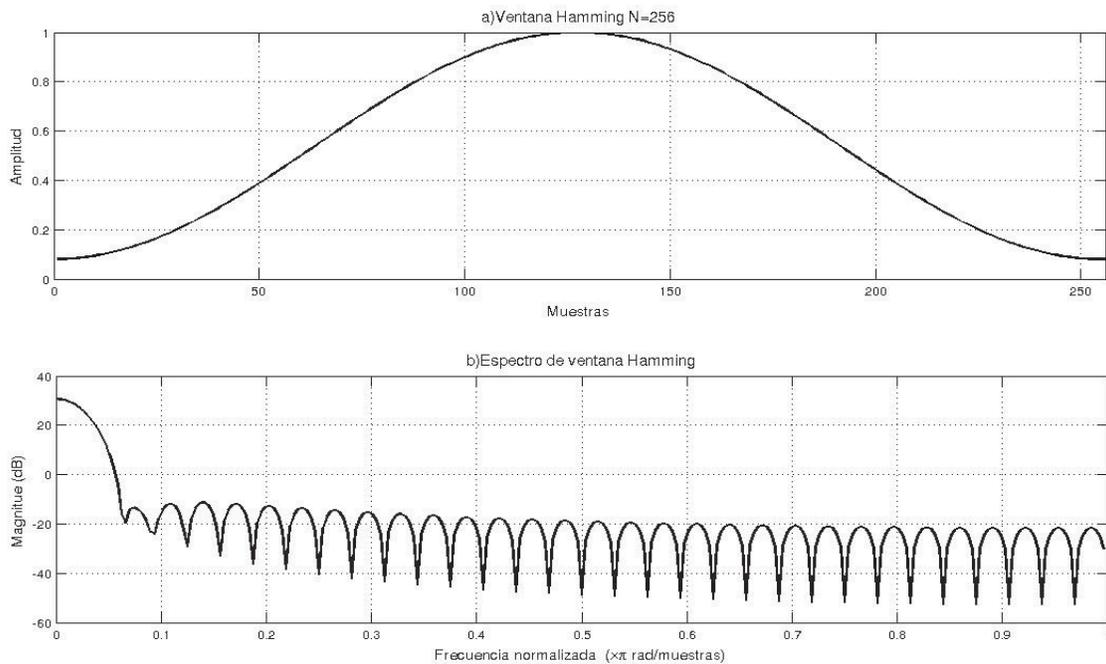


Figura 2.5: Ventana Hamming en el dominio temporal y frecuencial.

La diferencia entre la magnitud del primer y segundo lóbulo es de aproximadamente -42.6 dB. En el procesamiento de voz esta es la ventana más utilizada debido a que se mantiene una buena relación entre el ancho del lóbulo principal y la atenuación de los lóbulos secundarios.

Esta ventana permite recortar la señal sin alterar de manera significativa el contenido frecuencial de la señal suavizando las transiciones entre los límites de las tramas a analizar, esto se puede observar en la figura 2.6.

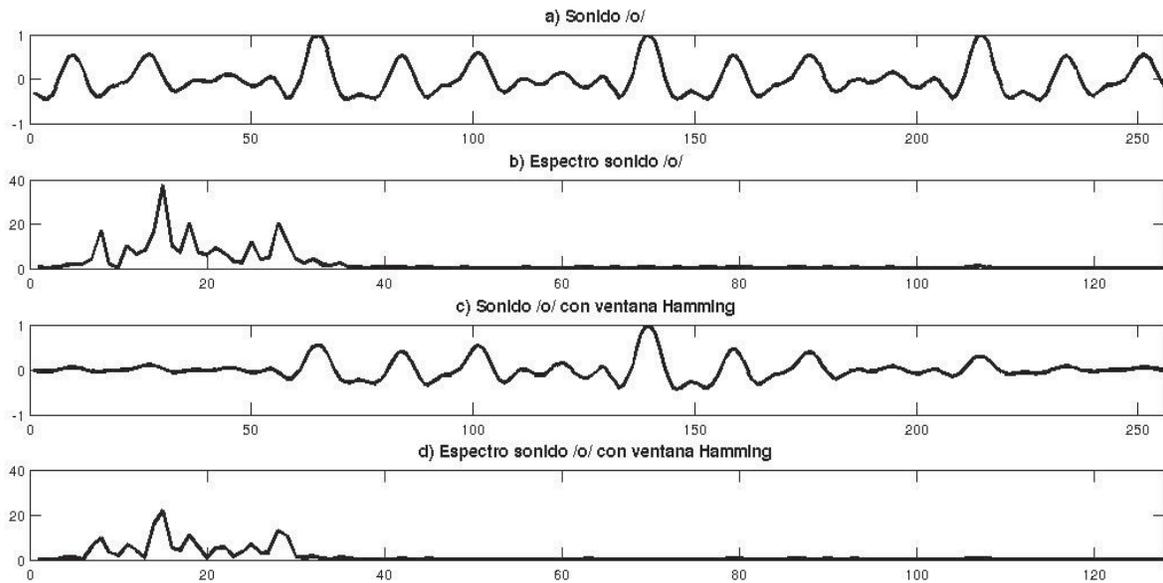


Figura 2.6: Segmento de voz aplicándole una ventana Hamming con $N=256$.

En las figura 2.6.a y 2.6.b se muestran una trama de voz y su espectro en frecuencia, respectivamente, mientras que en la figura 2.6.c y 2.6.d se muestran la misma trama de voz y su espectro una vez aplicada la ventana Hamming, por lo que se puede apreciar que esta ventana no afecta de manera significativa el espectro de la señal de voz. Una vez realizado este proceso se procede a estimar la energía de la señal para determinar que el segmento no sea silencio.

2.3.2 Umbral de silencio

El silencio es la ausencia de sonido y forma parte de las señales de voz, por lo que es necesario determinar si el segmento de voz analizado es un segmento de silencio, en caso de que así sea no será necesario realizar los demás pasos del proceso de análisis. Para poder determinar si el segmento analizado es silencio es necesario estimar su energía y compararla con la energía obtenida de un umbral de silencio experimental; el cálculo de la energía se hace por medio de la ecuación (2.3).

$$E = \sum_{i=0}^{N-1} |s(n)|^2 \quad (2.3)$$

Donde E es la energía del segmento, $s(n)$ son las muestras del segmento de voz y N es el número de muestras del segmento. El valor de umbral usado depende de las condiciones de ruido al momento de obtener la señal de voz. Si el segmento no se clasifica como silencio entonces se prosigue con el análisis de la señal para poder extraer los parámetros de ésta.

2.3.3 Filtro de preénfasis

En la figura 2.7 se muestra la envolvente típica de una señal de voz, se puede apreciar que las componentes de alta frecuencia están atenuadas, si esto se permitiera el modelo de predicción lineal aproximaría correctamente las frecuencias bajas de la señal, pero no las frecuencias altas.

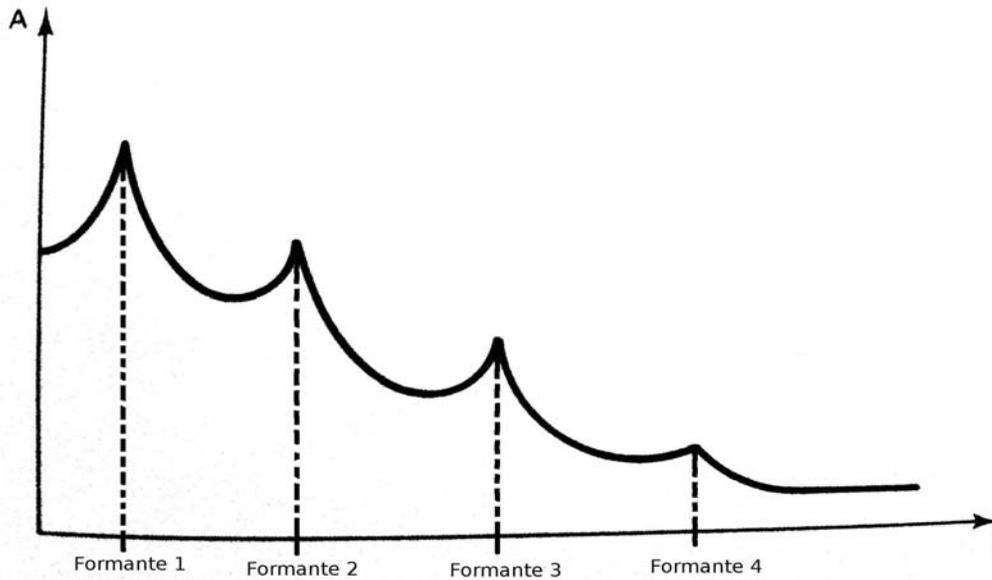


Figura 2.7: Envolvente típica de una señal de voz [9].

Por lo tanto es necesario compensar este comportamiento con un filtro de preénfasis, el cual tiene la forma de un filtro paso altas como en la ecuación (2.4) [8], en la figura 2.8 se muestra la respuesta en frecuencia de este filtro.

$$H(z) = 1 - \alpha z^{-1} \quad (2.4)$$

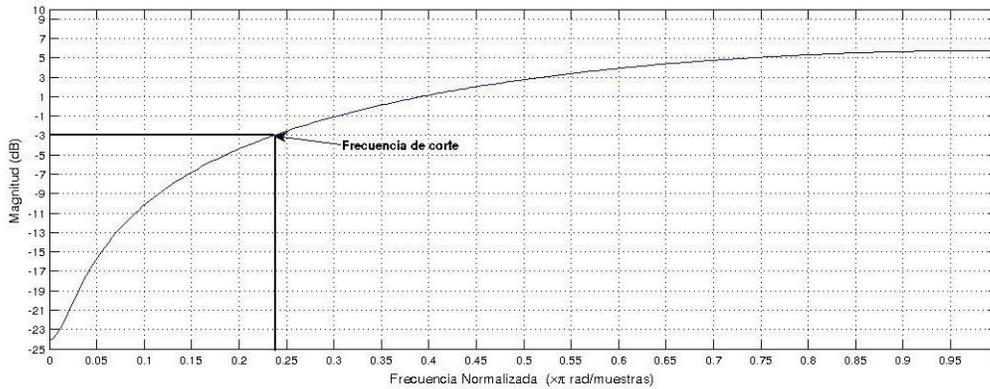


Figura 2.8: Filtro de preénfasis.

Donde α es un valor cercano a 0.9, el valor más común es el de 0.9375 [8], se puede observar que este es un filtro FIR con un cero en α . Con este valor de α la frecuencia de corte de este filtro esta en 600 Hz.

2.3.4 Estimación de la naturaleza de la señal

Como se mencionó en la sección 2.1.3 la señal de voz se puede clasificar en voceada y no voceada. En el caso de que una señal sea voceada, la entrada al filtro todo polo debe de ser un tren de impulsos los cuales están separados por un tiempo igual al periodo de pitch, por lo que es necesario determinar este periodo al momento de analizar cada trama de voz, una forma de hacer esto es mediante el método de recorte central, el cual explota las propiedades de la autocorrelación [4] y evitar de esta manera el uso de la transformada discreta de Fourier (DFT) o la transformada rápida de Fourier (FFT), lo que ahorra de manera significativa el costo computacional de estas operaciones.

Recorte central

Debido a que la autocorrelación contiene mucha información de la señal de voz, es necesario determinar si existe una frecuencia fundamental dentro de esta señal que pueda representarnos el pitch de la señal, por lo que es necesario resaltar esta periodicidad mientras que se minimiza el resto de la información de la señal que no se requiere; esto se logra recortando la amplitud de la señal a analizar a partir de niveles de umbral, de tal manera que la ecuación de recorte central queda definida como se muestra en la ecuación (2.5).

$$\begin{aligned}
 y(n) &= s(n) - C_L & \text{Si } s(n) \geq C_L \\
 y(n) &= s(n) + C_L & \text{Si } s(n) \leq -C_L \\
 y(n) &= 0 & \text{Otro caso}
 \end{aligned}
 \tag{2.5}$$

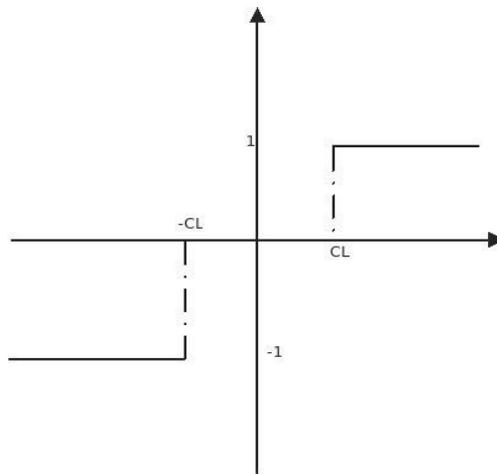
Otra forma de definir la función de recorte central es como se muestra en la ecuación (2.6)

$$\begin{aligned} y(n) &= 1 && \text{Si } s(n) \geq C_L \\ y(n) &= -1 && \text{Si } s(n) \leq -C_L \\ y(n) &= 0 && \text{Otro caso} \end{aligned} \quad (2.6)$$

Este recorte se realiza debido que sólo es necesario conservar la información de la periodicidad de la función, el valor de C_L es calculado para cada ventana y depende del valor máximo de ésta (A_{max}) [8].

$$C_L = 0.3A_{max} \quad (2.7)$$

En la figura 2.9 se muestra la función de recorte central, así mismo en la figura 2.10 se observa un ejemplo del recorte de una ventana por medio de la ecuación (2.6).



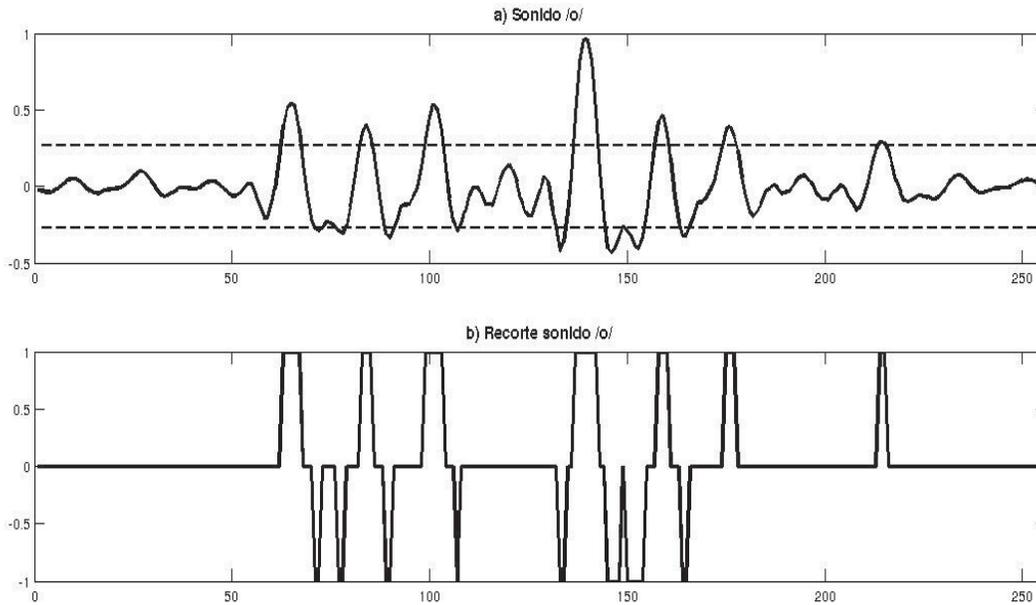


Figura 2.10: Ejemplo de recorte central de un segmento de voz.

Al emplear la ecuación (2.6) se pueden cometer errores debido a que existen muchos cambios de amplitud en el segmento (esto se puede apreciar en la figura 2.10), por lo que una manera de evitar esto es dividiendo la ventana a analizar en tres subsegmentos, se toman las amplitudes máximas del primer y tercer subsegmento (A_1, A_3) para posteriormente calcular el valor del umbral C_L como se muestra en la ecuación (2.8).

$$C_L = K \min(A_1, A_3) \quad (2.8)$$

Donde K es un parámetro de calibración que se encuentra entre 0.7 y 0.8 [8] y $\min()$ calcula el valor mínimo entre A_1 y A_3 .

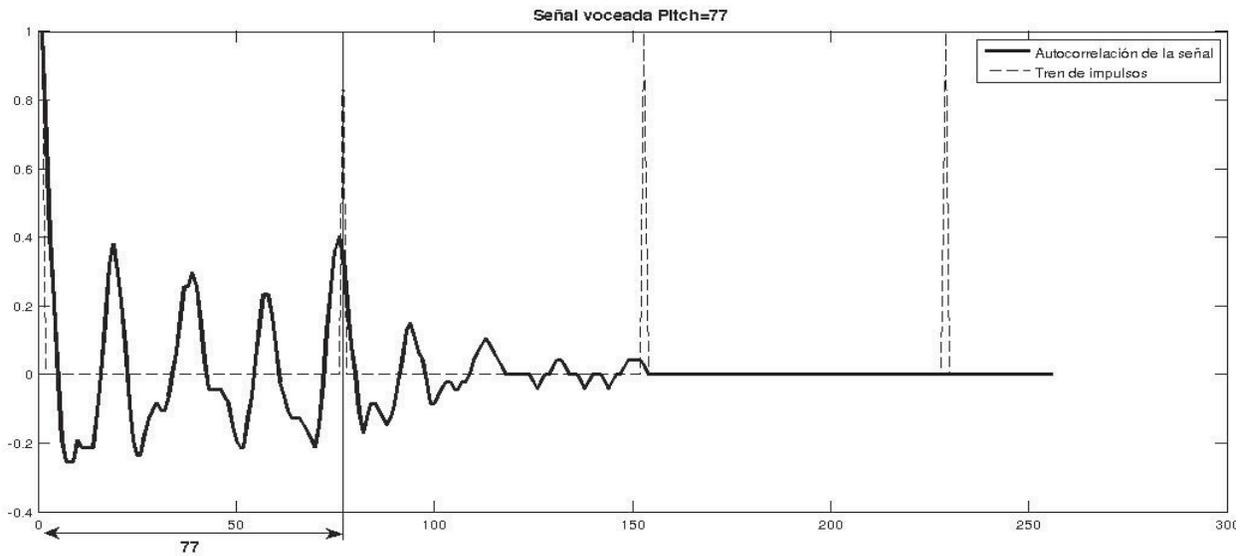
Estimación del periodo de pitch

Una vez determinado el valor C_L y recortada la señal, a esta se aplica la función de autocorrelación, $r_c(n)$, de la cual se obtiene el valor máximo entre las muestras 20 y 200, y se compara con un valor umbral $0.3r_c(0)$ si el valor máximo de la función entre dichas muestras es mayor que el valor umbral entonces el segmento se considera voceado.

Para determinar el periodo de pitch se obtiene el valor máximo de la función de autocorrelación entre las muestras 20 y 200, a este valor se le suma una constante de 20 [8].

En la figura 2.11 se muestra la función de autocorrelación de un segmento de voz el cual fue clasificado como voceado y obtuvo su periodo de pitch, así mismo se muestra un tren de

impulsos espaciados por un periodo igual al del pitch de la señal, dicho periodo tiene un valor de 77.



En la figura 2.12 se muestra la función de correlación correspondiente a un segmento de voz que fue clasificado como no voceado, como se puede apreciar no existe periodicidad en esta señal a diferencia del mostrado en la figura 2.11.

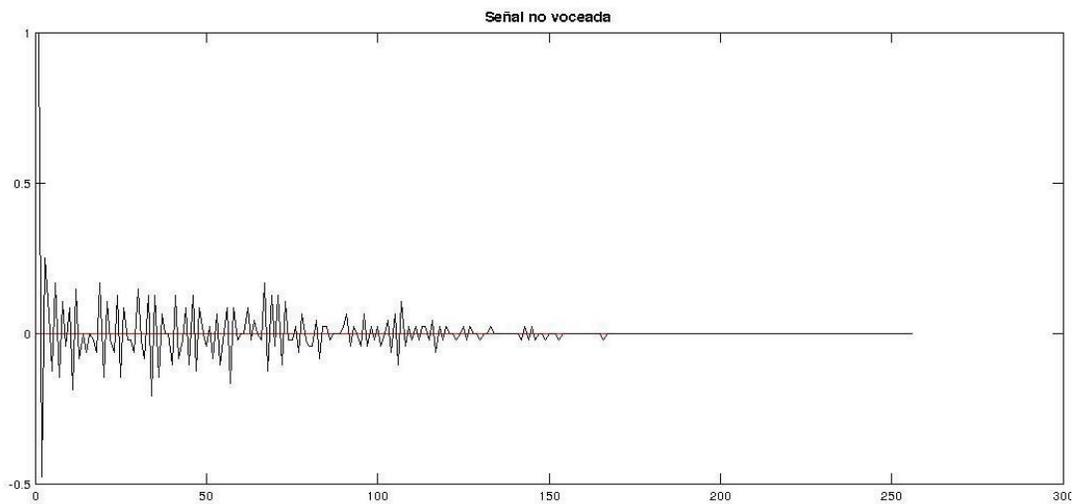


Figura 2.12: Autocorrelación de segmento no voceado.

2.3.5 Obtención de los coeficientes LPC empleando autocorrelación

Obteniendo la transformada Z inversa de la ecuación (2.1) se obtiene la ecuación en diferencias mostrada en la ecuación (2.9).

$$s(n) = Gu(n) - \sum_{k=1}^p a_k s(n-k) \quad (2.9)$$

donde $u(n)$ es la excitación, la cual depende de la naturaleza del segmento (voceado o no voceado) y $s(n)$ es la señal de voz.

La ecuación (2.10) muestra la señal estimada $\hat{s}(n)$ a partir de muestras pasadas con retraso k .

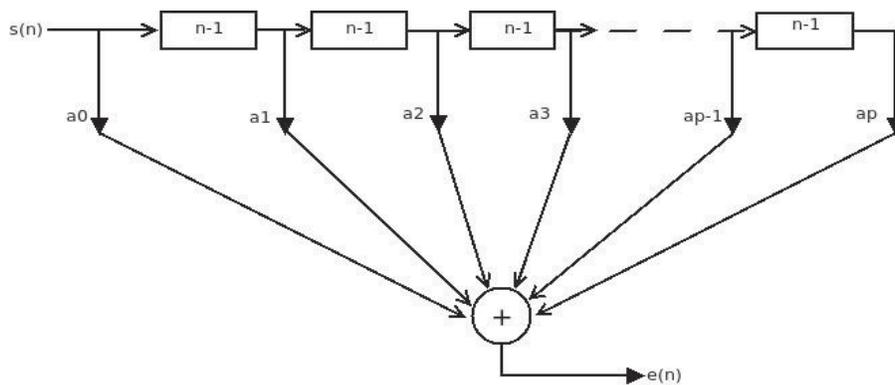
$$\hat{s}(n) = - \sum_{k=1}^p a_k s(n-k) \quad (2.10)$$

El error de predicción $e(n)$ se define como se muestra en la ecuación (2.11).

$$e(n) = s(n) - \hat{s}(n)$$

$$e(n) = s(n) + \sum_{k=1}^p a_k s(n-k) \quad (2.11)$$

La ecuación (2.11) puede ser representada como un filtro de predicción lineal como el mostrado en la figura 2.13.



La ecuación (2.11) puede ser representada usando notación vectorial ecuación (2.12).

$$e(n) = s(n) + A_p^T s_{pi} \quad (2.12)$$

$$A_p^T = [a_1 a_2 a_3 \dots a_p]$$

$$s_{pi} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_p \end{bmatrix}$$

Aplicando el criterio del error cuadrático a la ecuación (2.12) se obtiene la ecuación (2.13).

$$\begin{aligned} e^2(n) &= (s(n) + A_p^T s_{pi})(s(n) + A_p^T s_{pi}) \\ e^2(n) &= s^2(n) + 2s(n) A_p^T s_{pi} + A_p^T s_{pi} s_{pi}^T A_p \end{aligned} \quad (2.13)$$

Obteniendo la esperanza de la ecuación (2.13) se obtiene la ecuación (2.14).

$$E\{e^2(n)\} = r_p(0) + 2A_p^T r_p + A_p^T R_p A_p \quad (2.14)$$

Donde r_p es el vector de autocorrelación de la señal $s(n)$ y R_p es la matriz de autocorrelación, siendo esta una matriz Toeplitz simétrica. A_p Son los coeficientes que a_k del filtro todo polo empleado para la predicción lineal.

Aplicando el criterio de optimización sobre $E\{e^2(n)\}$, es decir, derivando la esperanza matemática del error cuadrático respecto de A_p .

$$\frac{\partial E\{e^2(n)\}}{\partial A_p} = 0 + 2r_p + 2R_p A_p = 0$$

$$R_p A_p = -r_p$$

Se obtiene la ecuación (2.15), la cual se denomina *ecuación de Wiener-Hopf* [8].

$$A_p = -R_p^{-1} r_p \quad (2.15)$$

La ecuación (2.15) puede ser expresada de forma matricial como se muestra en la ecuación (2.16).

$$\begin{bmatrix} r(0) & r(1) & r(2) & \dots & r(p-1) \\ r(1) & r(0) & r(1) & \dots & r(p-2) \\ r(2) & r(1) & r(0) & \dots & r(p-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r(p-1) & r(p-2) & r(p-3) & \dots & r(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} r(1) \\ r(2) \\ r(3) \\ \vdots \\ r(p) \end{bmatrix} \quad (2.16)$$

Resolviendo la ecuación (2.15) se obtienen los parámetros del filtro todo polo, el orden de predicción lineal p está asociado con el número de formantes que se quieran modelar. En la figura 2.14, 2.15 y 2.16 se muestran la respuesta del filtro todo polo para diferentes ordenes de predicción ($p= 5, 10$ y 20 respectivamente, $N=256$), así como la FFT de una señal de voz, se observa que la respuesta del filtro todo polo para un orden de predicción de 10 se aproxima a la envolvente del espectro de la señal.

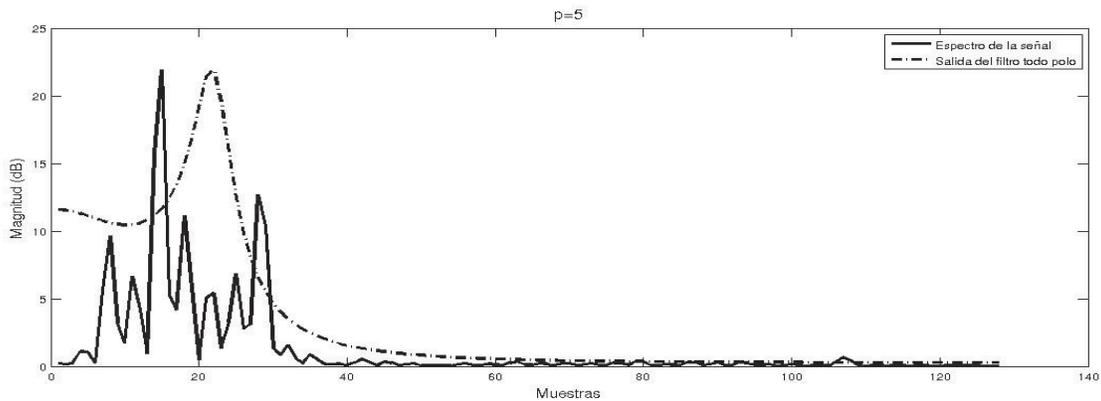


Figura 2.14: Espectro de la señal y salida del filtro todo polo $p=5$.

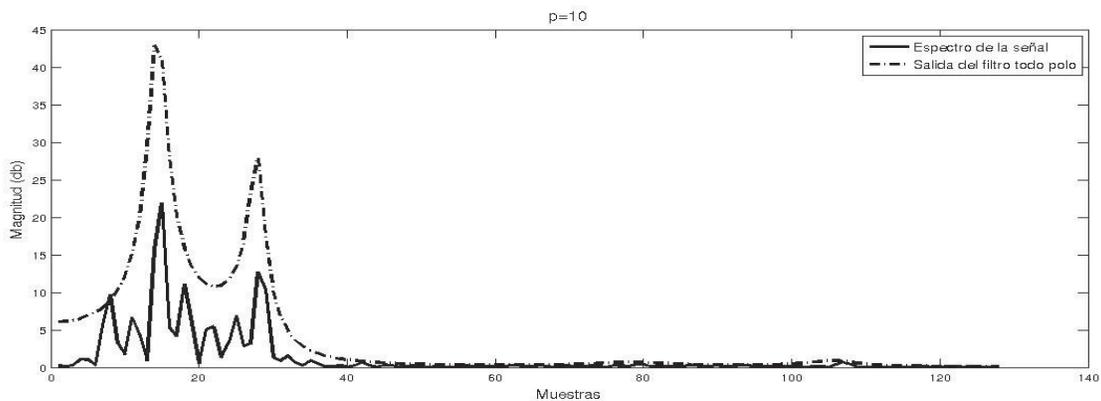


Figura 2.15: Espectro de la señal y salida del filtro todo polo $p=10$.

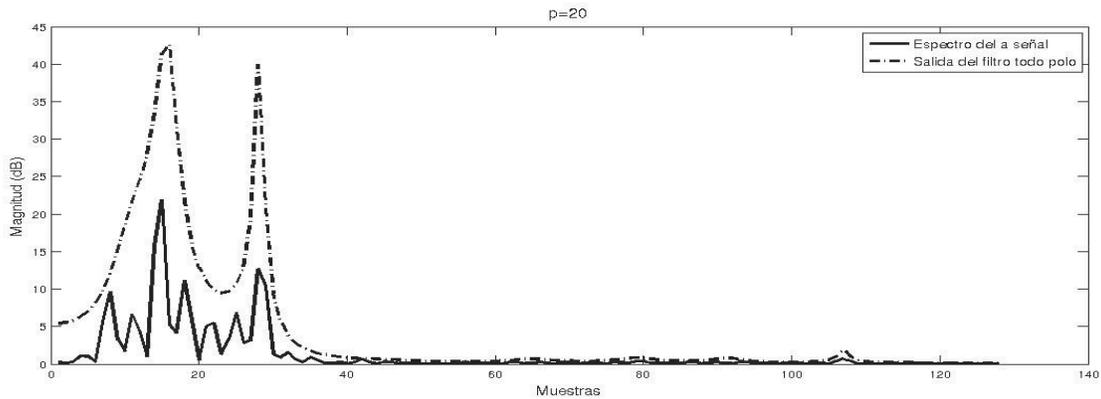


Figura 2.16: Espectro de la señal y salida del filtro todo polo $p=20$.

De las figuras 2.13, 2.14 y 2.15 se puede apreciar que el orden de predicción determina la aproximación de la envolvente a las frecuencias formantes, pero hay que considerar que al aumentar el orden de predicción también aumentan el número de operaciones que se deben realizar así como la memoria requerida para poder almacenar los coeficientes, el orden más usado para la estimación de los parámetros LPC y que presenta resultados óptimos es $p=10$.

2.3.6 Ganancia del segmento

Otro de los parámetros que se requiere calcular es la ganancia del segmento, G , de la ecuación (2.1), ésta se calcula mediante la ecuación (2.16) [8].

$$G = \sqrt{r(0) - \sum_{i=1}^p a_i r(i)} \quad (2.16)$$

Donde $r(i)$ es el vector de autocorrelación, a_i son los coeficientes LPC y p es el orden de predicción.

2.4 SINTESIS DE VOZ

Una vez que se tienen los parámetros de la voz se procede a realizar la síntesis, la cual consiste en reconstruir la señal de voz lo más parecida a la señal original, para esto se requiere identificar si el segmento de voz a sintetizar es un segmento de silencio, en caso contrario es necesario el uso de un filtro todo polo para realizar la síntesis de la señal así como el uso de un filtro de deénfasis el cual compensa los efectos de radiación de los labios.

2.4.1 Filtro todo polo

El filtro todo polo es un filtro IIR como en la ecuación (2.18), cuya entrada $u(n)$ es un tren de impulsos, en caso de que el segmento de voz a sintetizar sea voceado, o ruido blanco en caso de un segmento no voceado.

$$H(z) = \frac{G}{1 + \sum_{i=1}^p a_i z^{-i}} \quad (2.18)$$

Donde los coeficientes a_i son los coeficientes LPC calculados mediante la ecuación (2.16), si expresamos la ecuación (2.18) en el dominio temporal se obtiene la ecuación en diferencias (2.19).

$$\hat{s}(n) = Gu(n) - a_1 \hat{s}(n-1) - a_2 \hat{s}(n-2) - \dots - a_p \hat{s}(n-p) \quad (2.19)$$

donde G es la ganancia calculada mediante la ecuación (2.9), $u(n)$ es la excitación y $\hat{y}(n)$ es la voz sintética y p es el orden de predicción.

Como se muestra en la figura 2.4 aún falta modelar el efecto de radiación de los labios, esto se hace con un filtro de deénfasis.

2.4.2 Filtro de deénfasis

Este filtro consiste en un filtro IIR paso bajas como en la ecuación (2.20), donde α toma un valor cercano a 0.9, este filtro resalta las primeras frecuencias formantes, en las cuales se concentra la mayor energía de la señal lo que emula los efectos de radiación de los labios [8].

$$H(z) = \frac{1}{a - \alpha z^{-1}} \quad (2.20)$$

En la figura 2.17 se muestra la respuesta en frecuencia de este filtro, se puede apreciar que este filtro tiene un efecto inverso al del filtro de preénfasis, por esa razón es usual utilizar el mismo valor de α del filtro de preénfasis, la frecuencia de corte de este filtro se encuentra en 1300 Hz.

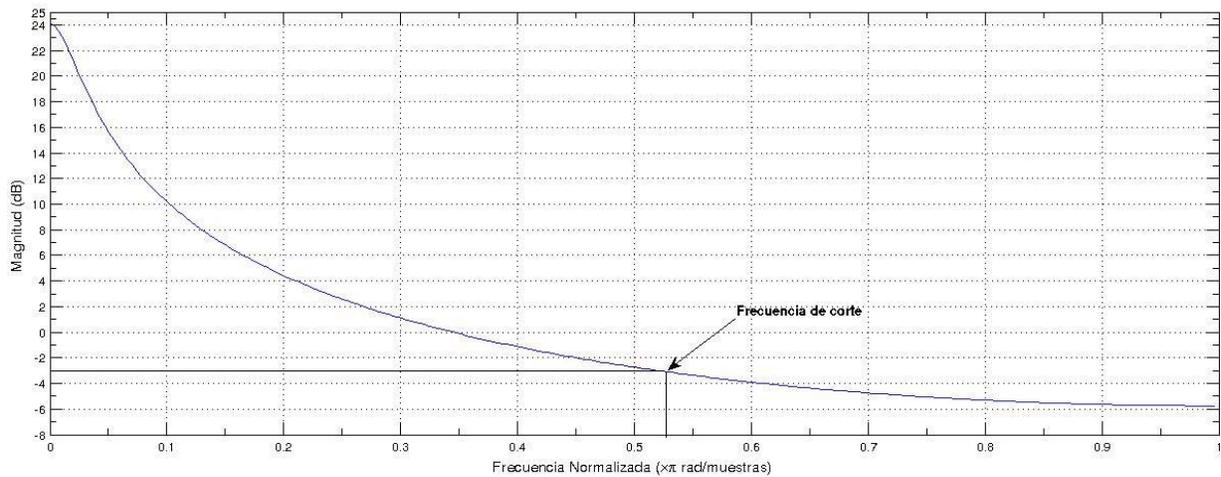


Figura 2.17: Filtro de deénfasis.

En la figura 2.18 se muestra una señal de voz junto con la reconstrucción de esta mediante un vocoder LPC.

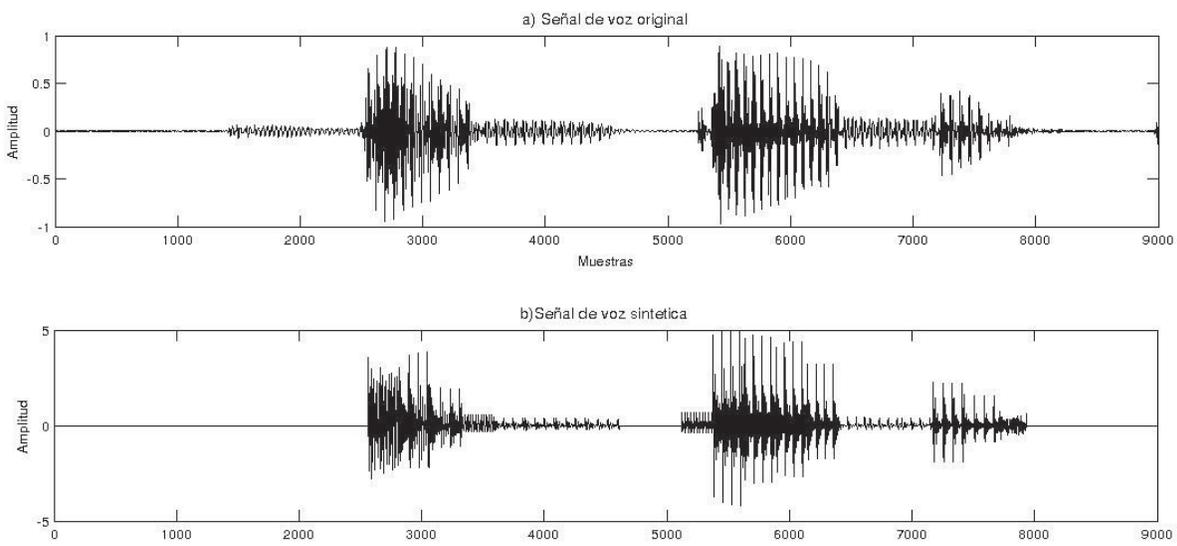


Figura 2.18: Ejemplo de una señal de voz original y sintética.

Se puede apreciar que la señal original y la señal reconstruida no guardan mucha relación en el espacio temporal, por lo que en la figura 2.19 se muestran los espectros de las mismas señales

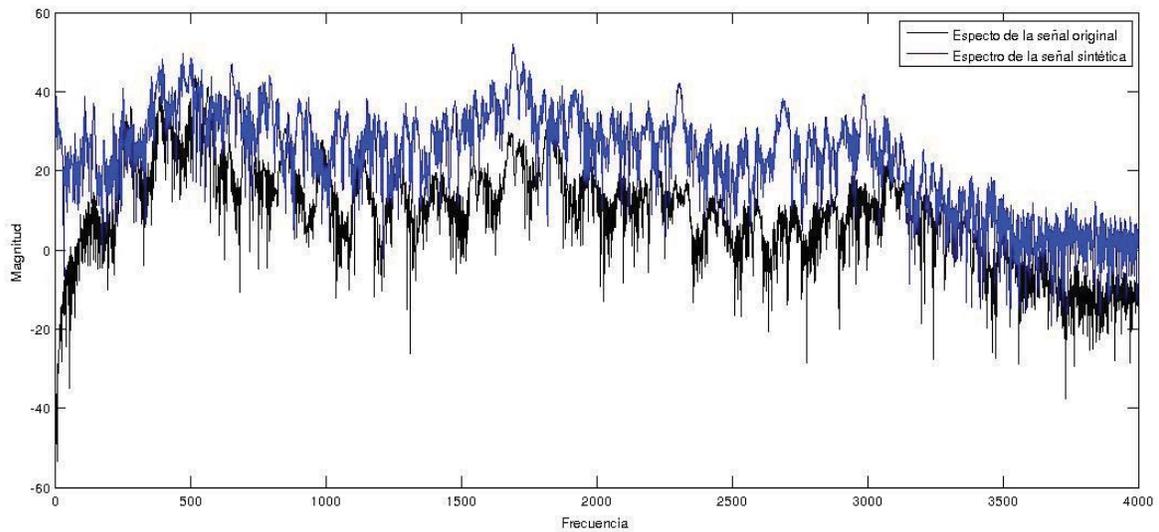


Figura 2.19: Espectro de una señal de voz original (negra) y sintética (azul).

En la figura 2.18 se observa similitud entre los espectros de las señales, por lo que la información de la señal original en el dominio frecuencial se conserva y la señal de voz es perceptible de manera aceptable.

2.5 RESUMEN

En este capítulo se dio una breve introducción a la producción de la señal de voz. Se mostró el modelo terminal - análogo para producción de voz sintética, así como la forma de obtener los parámetros para este modelo tales como los coeficientes del filtro todo polo, el periodo de pitch, la ganancia, etc, empleando el método de predicción lineal. Así mismo se mostraron los pasos necesarios para realizar el análisis y la síntesis de la señal de voz. El sistema implementado generará localmente voz sintética de acuerdo al evento que se detectó para posteriormente transmitírsela al usuario vía telefónica y de esta manera dar aviso sobre las condiciones del sistema.

Capítulo 3

Comunicación telefónica

El teléfono es un aparato cuyo fin es establecer comunicaciones habladas entre dos interlocutores situados en lugares diferentes. El 10 de marzo de 1876 se realizó la primer llamada telefónica, por Alexander Graham Bell, empleando un sistema telefónico funcional [10]. Desde entonces el teléfono ha pasado a formar parte de nuestra vida diaria, esto es debido a la necesidad de poder comunicarnos de forma remota con otras personas lo cual ha llevado al desarrollo de diversas tecnologías, entre ellas el telégrafo, fax, internet, etc.

En la actualidad debido al auge de la telefonía celular es posible establecer comunicación con otra persona prácticamente en cualquier parte, por lo que el uso de este tipo de comunicación proporciona una opción viable para dar aviso del estado de un sistema, como es el caso del sistema desarrollado en este trabajo,

En este capítulo se da una breve introducción al funcionamiento del sistema telefónico, las características de este sistema, sus subsistemas y los tonos de señalización que se requieren para que el sistema funcione.

3.1 SISTEMA TELEFÓNICO

Los teléfonos se utilizan para enviar y recibir señales acústicas por medio de señales eléctricas a distancia. El sistema telefónico es una red que consta de cuatro subsistemas principales los cuales son: el teléfono, los medios de transmisión, los conmutadores y las señales de control, todos estos sistemas se analizaran de manera breve durante este capítulo [11].

En la actualidad para poder realizar una llamada telefónica es necesario que el aparato sea capaz de realizar las siguientes acciones [3]:

1. Descolgar el auricular.
2. Detección de un tono llamado discar enviado por la central telefónica.
3. Teclear el número telefónico a llamar.
4. Detección de tonos que indican el estado de la llamada (llamando, ocupado, etc.).
5. Notificar cuando exista una llamada entrante al teléfono.
6. Convierte el lenguaje de una persona en señales eléctricas para su transmisión.
7. Convierte la señales eléctricas recibidas en audio.
8. Ajusta los cambios en la fuente de alimentación que recibe.
9. Notifica al sistema que una llamada ha terminado al colgar el auricular.

Para poder realizar todas estas acciones el teléfono requiere de una serie de circuitos que se conectan con la central telefónica los cuales sirven para poder conectar el teléfono de la persona que marco con el teléfono marcado.

3.2 CIRCUITO TELEFÓNICO

Todos los teléfonos están conectados a una central, la cual tiene equipos de conmutación, de señalización y baterías, encargadas de suministrar corriente para hacer funcionar el teléfono; los teléfonos se encuentran conectados a esta central por dos conductores llamados T o punta (*tip*) y R o anillo (*ring*), los equipos de conmutación de la central responden a las señales de marcado del teléfono y conectan este con el teléfono marcado.

En la figura 3.1 se muestra un diagrama simplificado del teléfono así como de la central telefónica; cuando el teléfono se encuentra colgado se aprietan los botones de la horquilla, lo que abre los contactos SH, se puede apreciar que en esta posición el teléfono queda desconectado de la central telefónica, el capacitor C bloquea el paso de la corriente continua de la batería, pero deja pasar la señal de la campanilla, por lo que esta permanece conectada. Cuando el auricular se retira los contactos SH se cierran, lo que permite el paso de la corriente en el circuito, a esta condición se le llama *descolgado*; en este estado se genera una señal que informa a la central que el usuario desea realizar una llamada.

Mientras el teléfono se encuentra en la posición de descolgado la central está dispuesta a aceptar el número telefónico al cual se desea marcar, a este número también se le puede llamar dirección [3].

La parte del teléfono con la cual habla la persona se llama transmisor, y se encarga de convertir las señales acústicas en señales eléctricas, esto se realiza mediante el uso de micrófonos, los cuales se pueden definir como transductores electroacústicos [12]. El receptor tiene la función inversa del transmisor, este se encarga de convertir las señales eléctricas recibidas en señales acústicas.

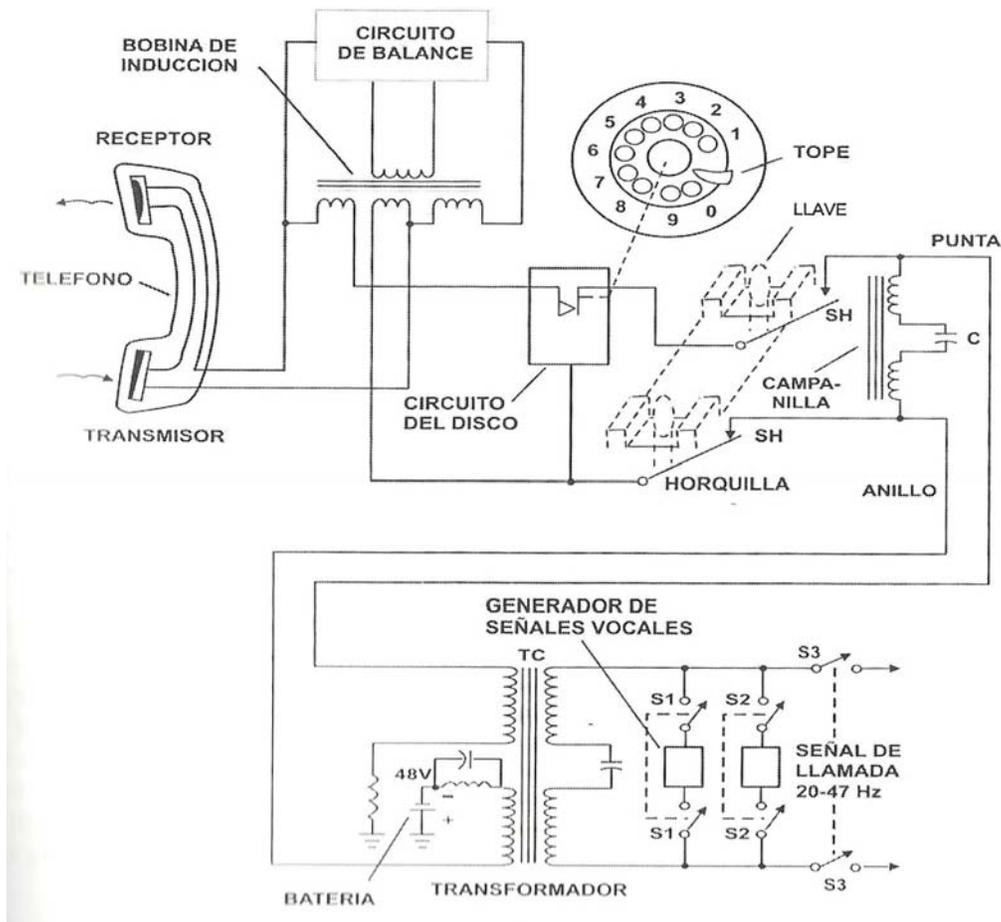


Figura 3.1: Circuito simplificado del aparato telefónico (analógico) [3].

Para poder establecer la llamada es necesario que la central conecte los dos teléfonos, esto se hace por medio de varios conmutadores y relevadores que conectan de forma automática los teléfonos al ser marcado el número, en la actualidad esto se logra mediante la decodificación de los tonos duales de multifrecuencia (DTMF), de las que se hablará más adelante en este capítulo.

3.3 MEDIOS DE TRANSMISIÓN

Para poder conectar los teléfonos se requiere de un medio por el cual las señales puedan ser transmitidas, por lo que surgen las redes de telecomunicaciones. Estas redes requieren de medios físicos para los sistemas de transmisión, ya sean analógicos o digitales, que permitan el intercambio de información entre usuarios sin importar la situación geográfica de estos. Algunos de los medios de transmisión son los siguientes [12]:

- Cables de pares simétricos: son un conjunto de conductores aislados entre sí y protegidos del exterior por una cubierta común.
- Cables coaxiales: están formados por dos conductores concéntricos, separados por aislante, el cable exterior es una malla de forma tubular como se ve en la figura 3.2.

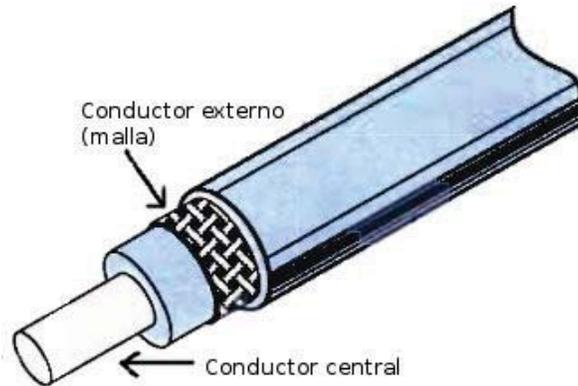


Figura 3.2: Cable coaxial [12].

- Cables de fibra óptica: conduce un rayo luminoso en su interior que puede ser utilizado como señal portadora.
- Transmisiones de radio: usan la atmósfera como soporte para la transmisión y usan una onda electromagnética como portadora, como se observa en la figura 3.3.

Dentro de las transmisiones por radio se encuentran los satélites, los cuales son repetidores de un enlace terrestre, estos se emplean para retransmitir las señales de una antena a otra, como se muestra en la figura 3.4.

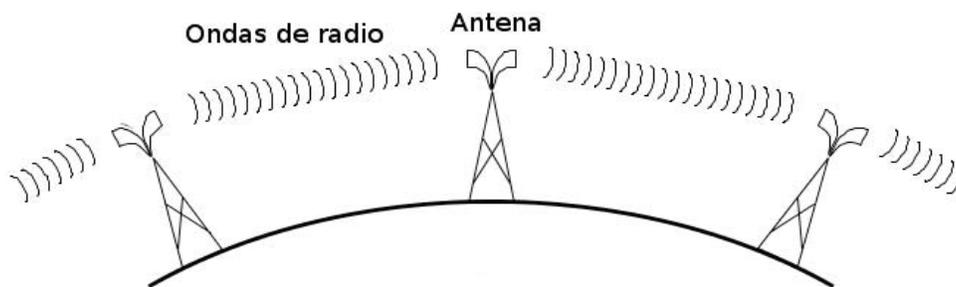


Figura 3.3: Transmisión empleando ondas de radio [11].

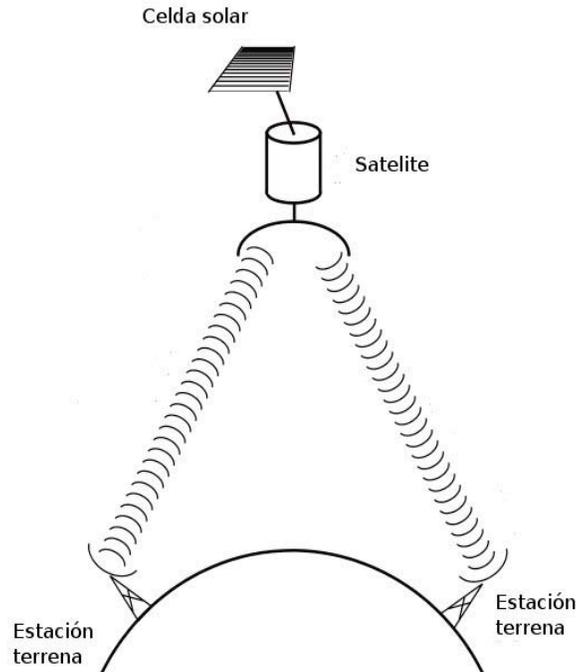


Figura 3.4: Transmisión empleando satélites [11].

La red telefónica usa de manera extensiva líneas cableadas para las transmisión de las señales de voz, pero en la actualidad, gracias al auge de la telefonía celular, muchas de las transmisiones son realizadas usando transmisiones de radio, lo que permite la portabilidad del equipo telefónico.

Es necesario destacar que todos los medios de transmisión pueden distorsionar las señales transmitidas, tanto en amplitud como en fase, además de que dichas señales pueden llegar a ser afectadas por el ruido presente en el ambiente.

Todos estos medios de transmisión deben de cumplir ciertos requisitos para poder transmitir de manera eficiente la señales de voz, una de las características más importantes a tomar en cuenta es el ancho de banda.

3.3.1 Transmisión de voz

La voz contiene un espectro de frecuencias que van desde los 100 Hz hasta los 6000Hz, pero se ha encontrado que la mayoría de la energía de esta señal se localiza entre los 200 Hz y 4000Hz como se muestra en la figura 3.5.

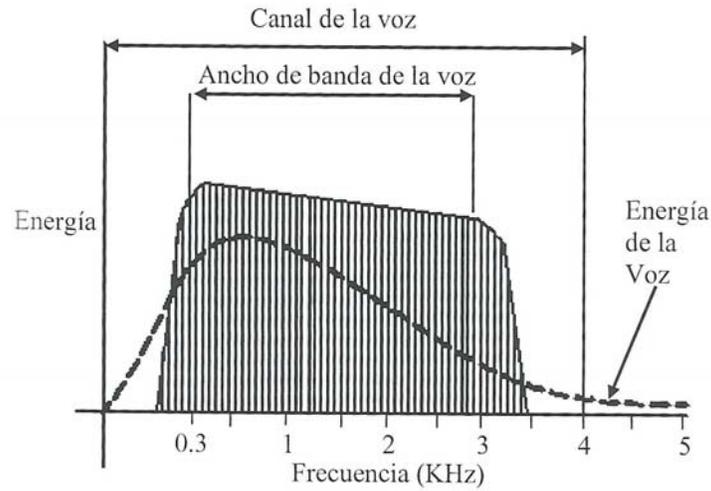


Figura 3.5: Ancho de banda de la señal de voz [12].

Para evitar el ruido, que puede causar problemas en la comunicación, los circuitos telefónicos son diseñados de tal manera que solo permiten el paso de ciertas frecuencias, las cuales se encuentran entre 0 y 4000 Hz, a este intervalo de frecuencias se le conoce como banda base, sin embargo, el ancho de banda de la línea no se utiliza por completo para la transmisión de las señales de voz, por lo que éstas se restringen a un intervalo de 300 a 3000 Hz [12].

También es necesario considerar que junto con las señales de voz se envían señales de control del sistema así como las señales necesarias para la marcación, algunas de estas señales se encuentran dentro del ancho de banda permitido para la señal de voz.

3.4 CONMUTADORES

En las primeras redes telefónicas los teléfonos estaban conectados uno al otro, sin conmutadores, pero a medida que la cantidad de teléfonos aumento este método resulto poco práctico, por lo que se establecieron centrales de conmutación, las cuales se encargaban de realizar la conexión entre los diferentes teléfonos.

En las primeras décadas de la telefonía la conexión de los teléfonos era de forma manual, y era realizada por los *operadores*, los cuales conectaban las líneas empleando cables, como se muestra en la figura 3.6, de este sistema provienen los nombres de punta y anillo que se emplean actualmente para conectar el teléfono con la central.

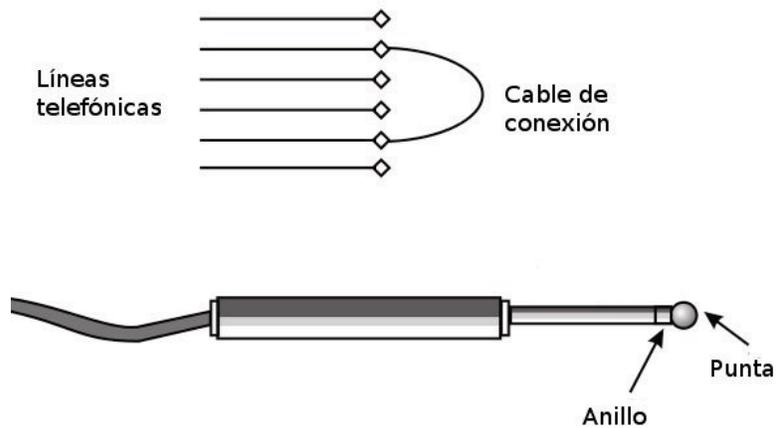


Figura 3.6: Conexión manual entre líneas [11].

Este tipo de conexiones eran lentas y costosas, por esta razón se buscaron formas de realizar esta tarea de manera automática. Actualmente estas conexiones se realizan de forma automática mediante circuitos electrónicos los cuales conectan los teléfonos de manera más rápida.

Para realizar las conexiones se requieren de dos sistemas, como se observa en la figura 3.7, la red de conmutación, que se encarga de conectar las líneas telefónicas y el sistema de control, el cual es responsable de emitir los comandos necesarios para operar los interruptores en el momento adecuado y realizar las conexiones específicas.

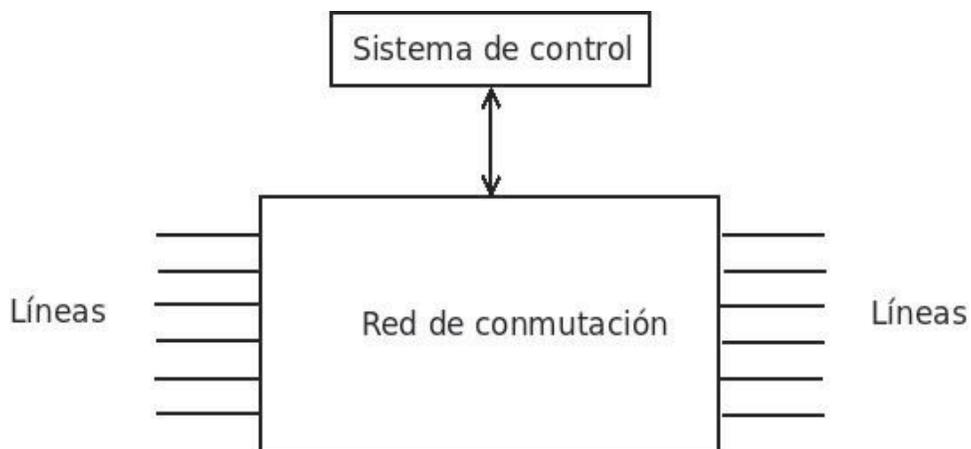


Figura 3.7: Sistema de conmutación telefónico [11].

En un principio tanto el sistema de control como la red de conmutación eran completamente electromecánicos, posteriormente se utilizaron computadoras para realizar las funciones del

sistema de control, pero seguían utilizando elementos electromecánicos en la red de conmutación, los sistemas actuales son totalmente electrónicos, tanto el sistema de control como la red de conmutación [11].

3.5 SEÑALES DE CONTROL

Como se mencionó anteriormente los sistemas telefónicos requieren de señales de control que indican a la central cuando se quiere realizar una llamada o cuando es necesario desconectar las líneas para que estas puedan ser usadas nuevamente, así mismo es necesario indicarle al usuario cuando la llamada se está procesando o si hay algún problema en su línea.

Al levantar el auricular la corriente continua fluye a través del teléfono, esta corriente es detectada por la central y se interpreta como una solicitud de servicio, esta corriente se mantiene mientras el teléfono se encuentre en uso y cuando esta corriente deja de fluir (el teléfono se encuentra en la posición de colgado) la central interpreta que la llamada ha terminado.

Además de las señales utilizadas por la central, también existen una serie de tonos que indican el estado de la línea tanto a la persona que marcó como a la persona que está recibiendo la llamada, estos tonos son [11]:

- Tono de ocupado: indica a la persona que marcó que el teléfono se encuentra descolgado, este tono consta de dos señales senoidales de 480 y 620 Hz, además de que se encuentra de manera alternada cada 0.5 s.
- Tono de llamada: este tono se presenta cuando la llamada está en proceso y consta de dos señales senoidales de 440 y 480 Hz, al mismo tiempo se envía una señal al teléfono marcado para indicarle que hay una llamada esperando, esta señal tiene un voltaje RMS de 75 V y una frecuencia de 20 Hz; ambas señales están presentes durante 2 s y ausentes durante 4 s, como se muestra en la figura 3.8.
- Tono de marcación: este tono indica al usuario que el teléfono está en funcionamiento y que la central reconoce que se encuentra descolgado, consta de dos señales senoidales de 350 y 440 Hz.

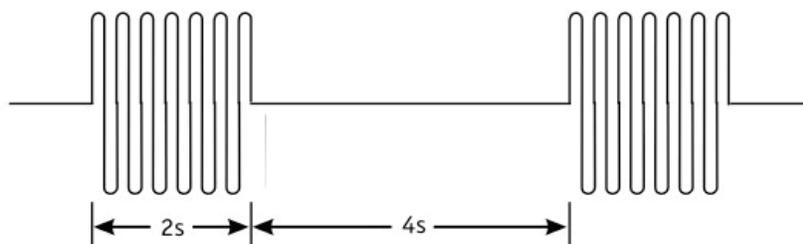


Figura 3.8: Timbre del teléfono [10].

La llamada termina cuando cualquiera de los usuarios cuelga el teléfono, esto desconecta nuevamente los circuitos y hace que la central libere las líneas para ambos teléfonos.

Para poder realizar una llamada telefónica es necesario marcar el número del teléfono con el que se desea comunicar, esto se realiza mediante el uso de una serie de tonos llamados DTMF, los cuales por su importancia dentro del sistema serán tratados en en la siguiente sección.

3.6 TONOS DUALES DE MULTIFRECUENCIA (DTMF)

Cuando el usuario pulsa una tecla del teléfono se generan dos señales senoidales de frecuencias diferentes, las cuales son decodificadas por la central para poder detectar el dígito que se marcó y de esta manera poder conectar el teléfono que marcó con el teléfono marcado.

Como se mencionó, los tonos duales de multifrecuencia (DTMF) son la suma de dos señales senoidales cuyas frecuencias definen el dígito marcado, estas frecuencias se clasifican en tono alto y tono bajo como se muestra en la figura 3.9, estas fueron seleccionadas de tal manera que no se produzcan armónicos ni intermodulación, por lo que tienen las siguientes características [3]:

- Ninguna frecuencia es múltiplo de otra.
- La diferencia entre dos frecuencias nunca es igual a cualquiera de las otras frecuencias.
- La suma de dos frecuencias no es igual a cualquiera de las otras frecuencias.

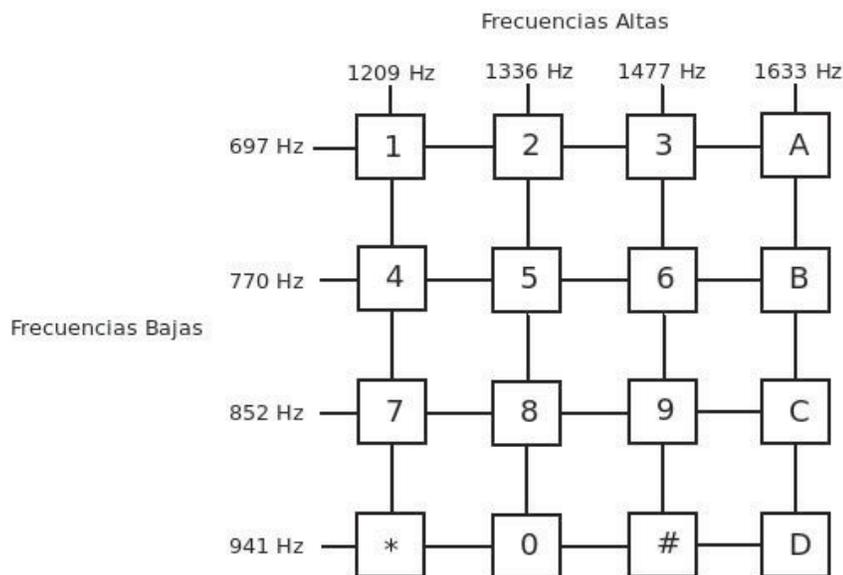


Figura 3.9: Matriz de frecuencias del teclado telefónico.

En la figura 3.10 se muestra la forma de onda de una señal DTMF correspondiente al número 1, así como su espectro en frecuencia (el espectro está escalado con 256 muestras y una frecuencia de muestreo de 8 KHz).

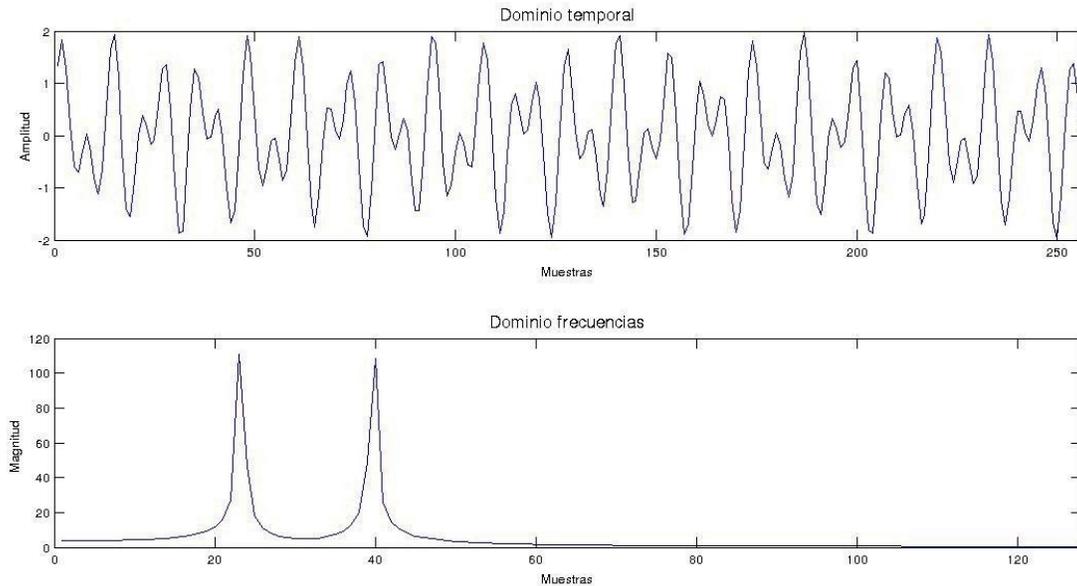


Figura 3.10: Ejemplo de una señal DTMF en el dominio temporal y frecuencial.

En el espectro en frecuencia se pueden apreciar los dos tonos que forman la señal DTMF. Para que la central pueda detectar los tonos es necesario que estos cumplan con ciertos requerimientos, los cuales son [3]:

- Se deben de transmitir menos de 10 dígitos por segundo.
- El tiempo mínimo entre dos tonos debe de ser de 50 ms.
- Los tonos deben presentarse durante al menos 40 ms.
- Las tolerancias aceptables en las frecuencias son de $\pm 1.5\%$ en Norte América.

Para que el sistema pueda realizar la llamada telefónica debe generar estos tonos tomando en cuenta todas estas consideraciones.

3.6.1 Generación de tonos

Los tonos DTMF pueden ser generados conectando dos generadores de señales senoidales en paralelo, donde se genera una señal de frecuencia alta y una de frecuencia baja, como se muestra en la figura 3.11, debido a que el sistema implementado es un sistema digital, dichos generadores deben de ser digitales.

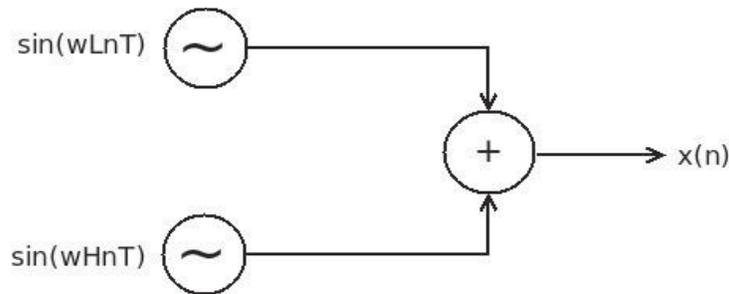


Figura 3.11: Diagrama de bloques de un generador DTMF

Donde $x(n)$ queda definida por la ecuación (3.1).

$$\begin{aligned}
 x(n) &= \sin(\omega_H nT) + \sin(\omega_L nT) \\
 \omega_H &= 2\pi f_H \\
 \omega_L &= 2\pi f_L
 \end{aligned}
 \tag{3.1}$$

Donde T es el periodo de muestreo de la señal, las frecuencias f_H y f_L son las frecuencias que identifican la tecla presionada y n es el número de muestra actual.

Existen diferentes formas de generar señales senoidales de forma digital, las más utilizadas son:

- a) Aproximación de polinomios: se basa en expresar las funciones seno y coseno como una expansión de la serie infinita de Maclaurin.
- b) Oscilador recursivo: se realiza utilizando la transformada Z de una señal senoidal o cosenoidal.
- c) Por búsqueda de tabla: esta técnica se basa en leer una serie de datos guardados, los cuales representan valores discretos de la forma de onda, la cual se genera al repetir esta lectura de manera cíclica.

En el presente trabajo se utilizó un oscilador recursivo, el cual se basa en el uso de la respuesta al impulso de la transformada Z de una función senoidal, ya que emplea operaciones básicas, multiplicaciones, sumas y retardos, las cuales pueden ser implementadas fácilmente en un DSP también tiene la ventaja de que se requiere una cantidad mínima de memoria para poder realizar señales senoidales.

En la ecuación 3.2 se muestra la transformada Z de una señal senoidal.

$$TZ\{\sin(\omega_0 n)\} = \frac{z^{-1} \sin(\omega_0)}{1 - 2z^{-1} \cos(\omega_0) + z^{-2}}
 \tag{3.2}$$

Si se considera que la salida del sistema debe de ser una señal senoidal, se tiene el sistema de la ecuación (3.3)

$$H(z) = \frac{Y(z)}{X(z)} = \frac{z^{-1} \sin(\omega_0)}{1 - 2z^{-1} \cos(\omega_0) + z^{-2}} \quad (3.3)$$

Desarrollando la ecuación (3.4):

$$\begin{aligned} Y(z)[1 - 2z^{-1} \cos(\omega_0) + z^{-2}] &= X(z)[z^{-1} \sin(\omega_0)] \\ Y(z) - 2z^{-1} Y(z) \cos(\omega_0) + z^{-2} Y(z) &= z^{-1} X(z) \sin(\omega_0) \\ Y(z) &= z^{-1} X(z) \sin(\omega_0) + 2z^{-1} Y(z) \cos(\omega_0) - z^{-2} Y(z) \end{aligned} \quad (3.4)$$

Obteniendo la transformada Z inversa de la ecuación (3.4):

$$y(n) = \sin(\omega_0)x(n-1) + 2\cos(\omega_0)y(n-1) - y(n-2) \quad (3.5)$$

Para que la respuesta del sistema sea una señal senoidal es necesario que la entrada sea una función impulso unitario, el cual se define en la ecuación (3.6).

$$x(n) = \delta(n) = \{1, 0, 0, 0, 0, \dots\} \quad (3.6)$$

La ecuación (3.5) puede ser vista como un filtro IIR como el mostrado en la figura 3.12.

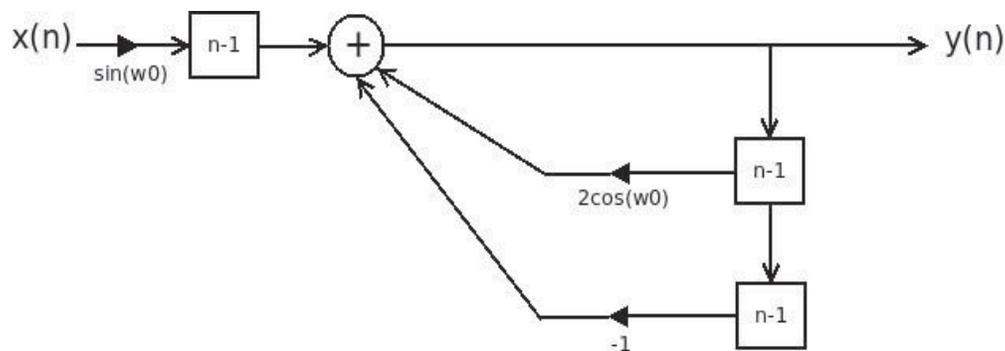


Figura 3.12: Filtro IIR para generar señales senoidales.

En la figura 3.13 se muestra la respuesta en frecuencia del filtro mostrado en la figura 3.12, se puede observar que tiene el comportamiento de un filtro pasa banda muy selectivo el cual al tener como entrada la función impulso unitario, cuyo espectro en frecuencia es una constante de valor 1, tendrá como salida una señal senoidal.

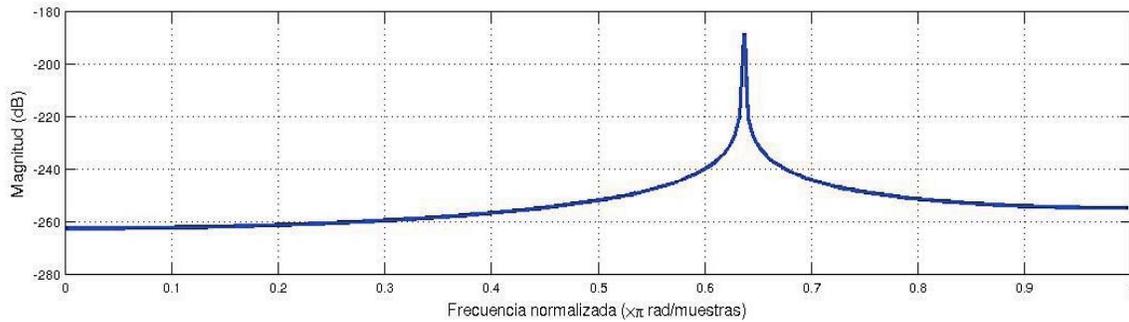


Figura 3.13: Respuesta en frecuencia del filtro IIR.

La ecuación (3.4) con condiciones iniciales se muestra en la ecuación (3.7).

$$\begin{aligned}
 y(0) &= 0 \\
 y(1) &= \sin(\omega_0) \\
 y(n) &= 2\cos(\omega_0)y(n-1) - y(n-2) \text{ para } n=2,3,\dots
 \end{aligned}
 \tag{3.7}$$

Una vez que se establece la ecuación del filtro y sus condiciones iniciales, el sistema permanecerá oscilando por siempre. Cabe destacar que empleando la ecuación (3.6) se pueden generar múltiples tonos si se cuentan con los valores correspondientes de $\sin(\omega_0)$ y $2\cos(\omega_0)$ empleando la misma estructura del filtro para cada tono, así mismo es posible generar los tonos DTMF al generar los tonos alto y bajo y sumarlos.

En la figura 3.14 se muestra una señal senoidal, con frecuencia de 697 Hz, 256 muestras y frecuencia de muestreo de 8000 Hz, así como su espectro en frecuencia generada empleando este método.

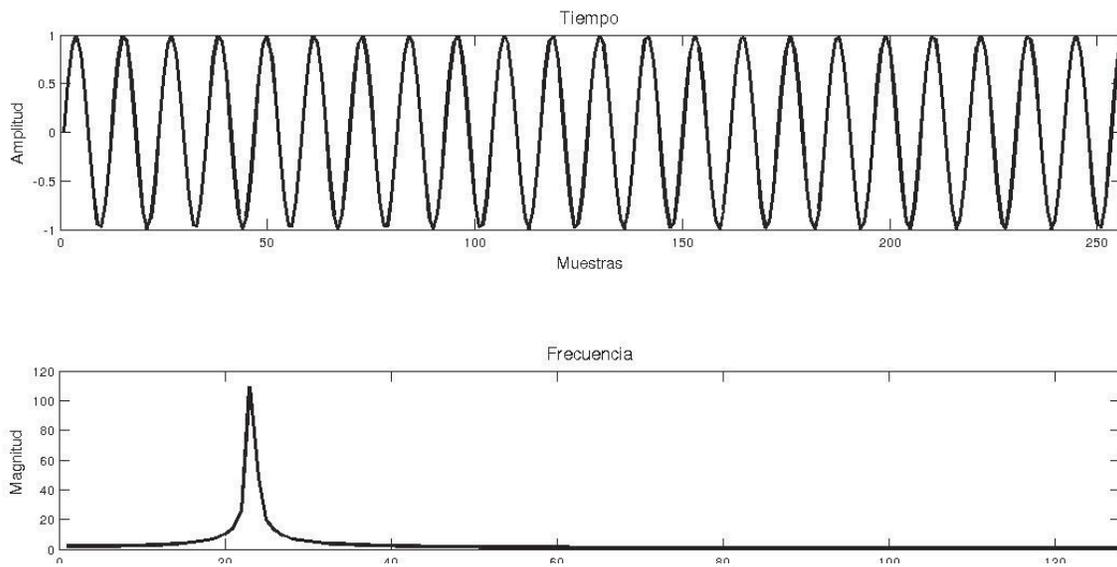


Figura 3.14: Ejemplo de una señal senoidal empleando un oscilador recursivo.

Como se puede observar de la figura 3.14 la señal generada empleando este método tiene un buen comportamiento en tiempo y en frecuencia.

Una vez generadas de manera digital las señales DTMF estas deben de ser convertidas en señales analógicas, para poder ser transmitidas mediante la línea telefónica y que la central pueda enlazar los teléfonos.

El sistema a diseñado contiene dos osciladores recursivos los cuales generan los tonos alto y bajo de todas las señales DTMF, necesarios para el marcado telefónico, de tal manera que se puede marcar cualquier número telefónico, previamente definido, cuando el sistema requiera notificar al propietario acerca de un evento.

3.7 RESUMEN

En este capítulo se dio una introducción a los elementos que componen el sistema telefónico, los cuales son: el teléfono, los medios de transmisión, los conmutadores y las señales de control, todos estos elementos son necesarios para poder realizar llamadas, así mismo se mostró la manera de generar las señales DTMF que se emplean al marcar un número telefónico, estas señales se generaron empleando un oscilador recursivo cuya respuesta al impulso es una señal senoidal.

Capítulo 4

Análisis y diseño del sistema

En este capítulo se desarrolló el análisis y diseño del sistema implementado en la tarjeta de desarrollo DSK5402, la cual tiene un procesador digital de señales (DSP) TMS320C5402 de la compañía *Texas Instruments*, esto se hace tomando en cuenta la teoría vista en los capítulos anteriores.

El sistema consiste en una tarjeta que detectara la apertura de puertas o ventanas para dar aviso al propietario de la vivienda, también permite conectar el interfón de la casa, cuando este sea activado, con el teléfono del usuario.

Cabe destacar que al emplear una tarjeta de evaluación que contaba con las interfaces necesarias para la conexión de la línea telefónica, así como de micrófono y altavoz, ayudó mucho al desarrollo de este sistema, de tal manera que solo fue necesario emplear las APIs (interfaces de programación de aplicaciones) para poder hacer uso de estas.

El hardware y software que se emplearon para poder realizar las acciones antes mencionadas se explica a lo largo de este capítulo.

4.1 DESCRIPCIÓN GENERAL DEL SISTEMA

El sistema consiste en una tarjeta de evaluación la cual contiene un procesador digital de señales (DSP) conectado a una interfaz RJ-11 y a un códec de audio, en la figura 4.1 muestra un diagrama a bloques del sistema.

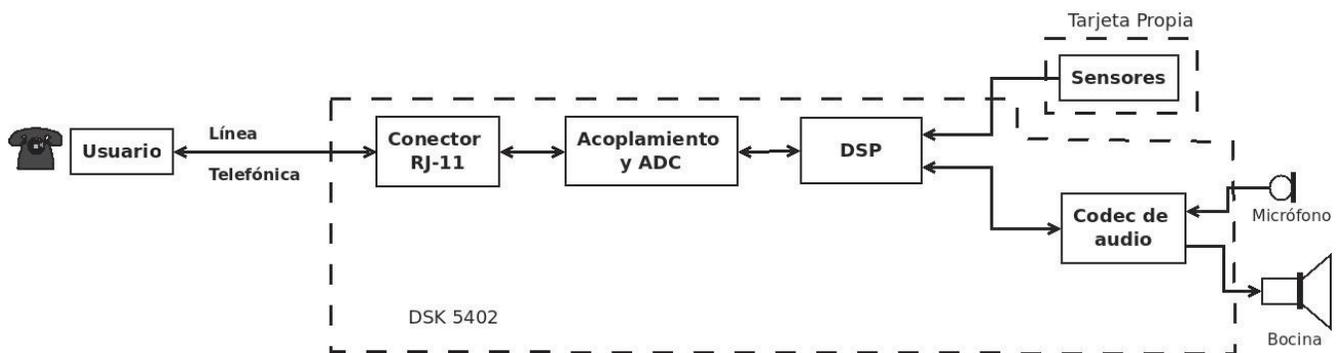


Figura 4.1: Diagrama de bloques del sistema.

En la figura 4.2 se muestra un diagrama de flujo del funcionamiento del sistema, al identificar la fuente de interrupción se determina la señal de voz a enviar.

Los parámetros necesarios para realizar la síntesis de voz se obtuvieron de muestras reales de voz las cuales fueron procesadas fuera de línea empleando Matlab®, con estos parámetros es posible realizar la síntesis de voz en tiempo real, en línea.

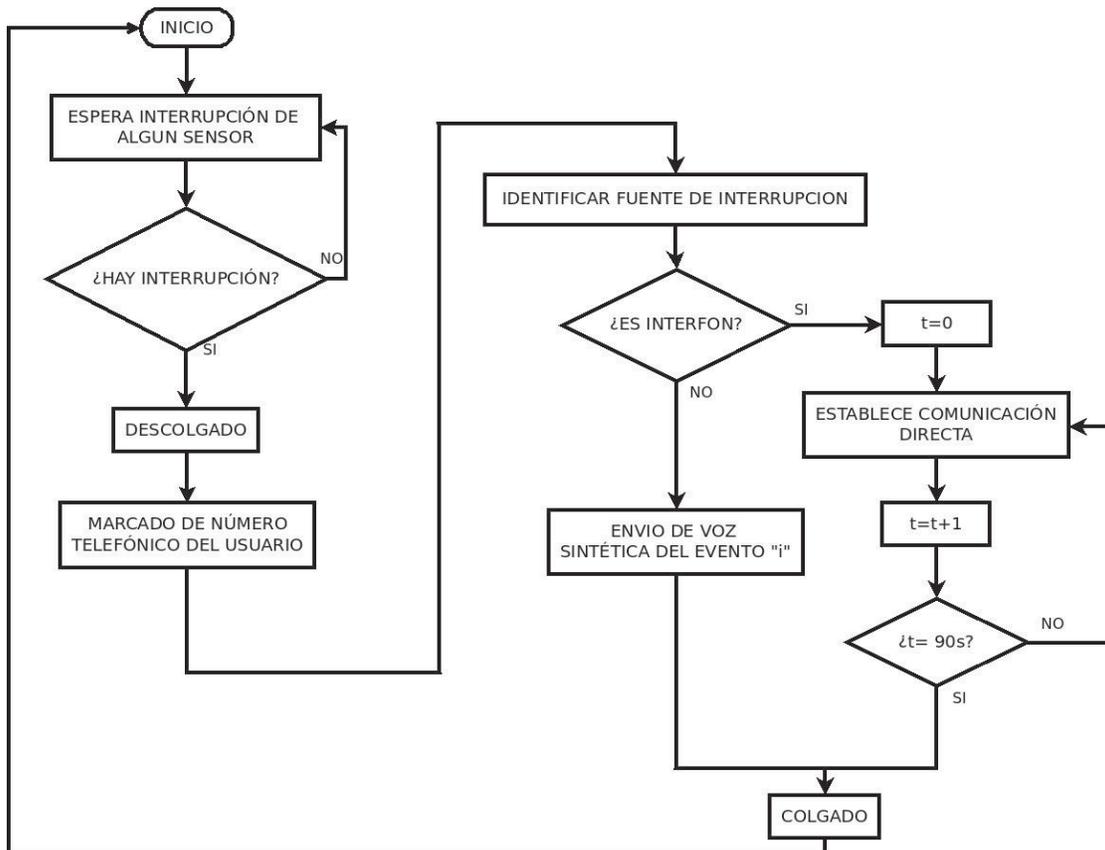


Figura 4.2: Diagrama de flujo del funcionamiento.

La voz sintética es enviada de forma intermitente durante un minuto de tal manera que, independientemente del momento en que el usuario conteste la llamada, el mensaje pueda ser recibido.

Para este trabajo se consideró una vivienda como la mostrada en la figura 4.3, determinándose que existen ocho posibles señales de alerta (estas se encuentran remarcadas en la figura), incluyendo la señal del interfón, las frases que se emplearon fueron las siguientes:

- Puerta principal abierta.
- Ventana de la sala abierta.
- Ventana de la cocina abierta.

- Ventana de la habitación uno abierta.
- Ventana de la habitación dos abierta.
- Ventana de la habitación tres abierta.
- Puerta del patio abierta.

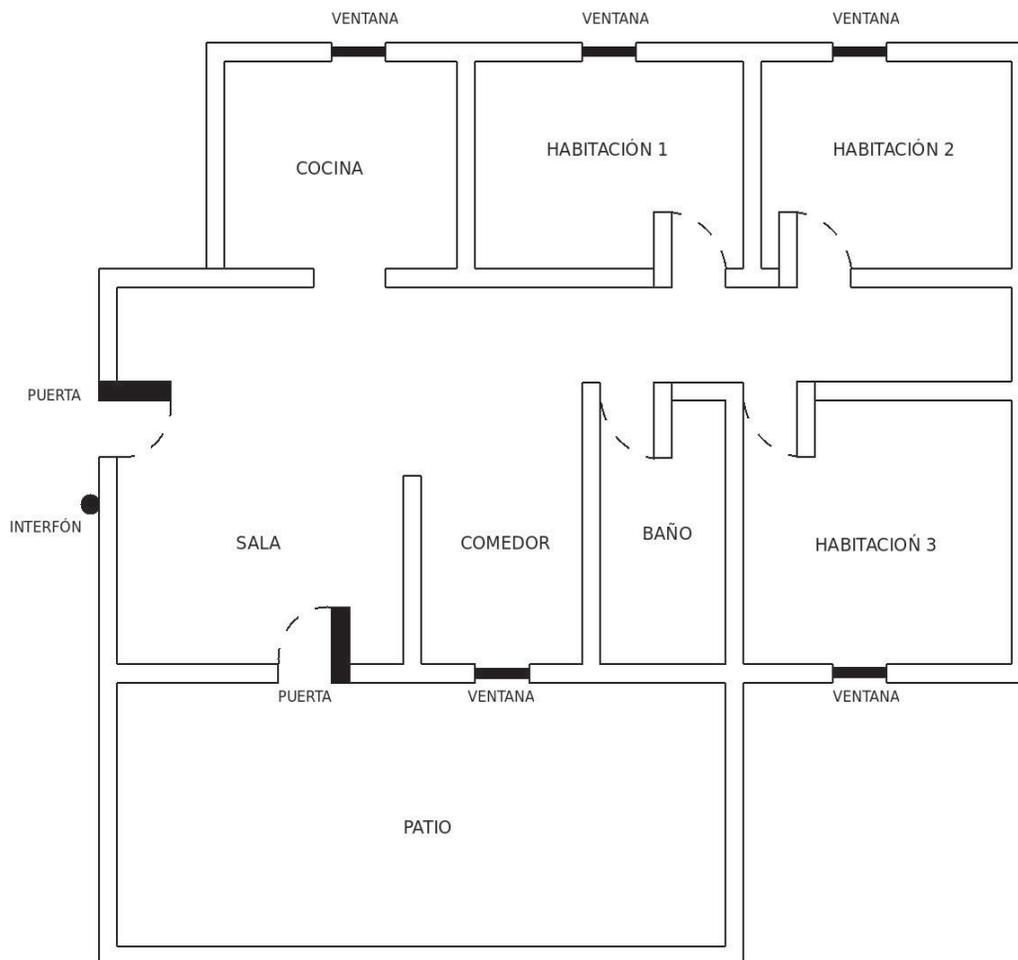


Figura 4.3: Diagrama de la vivienda.

Debido a que la función del interfón es comunicar a la persona que accionó el timbre con el dueño de la vivienda, esta señal de alerta no genera ninguna señal de voz sintética, en su lugar conecta la interfaz de audio con la línea telefónica de tal manera que el sistema funciona como un teléfono el cual está activado hasta que el dueño de la vivienda termine la llamada o hasta que pase un tiempo de 90s.

4.2 DSK TMS320C5402

La tarjeta que se empleó es la TMS320C5402 *DSP Starter Kit (DSK)* de Texas Instruments, ésta es una tarjeta de desarrollo de bajo costo que permite realizar y evaluar aplicaciones para el

DSP, en la figura 4.4 se muestra un diagrama a bloques de este sistema de desarrollo, esta tarjeta incluye:

- Un DSP TMS320C5402.
- Memoria externa SRAM de 6K x 16 bits.
- Memoria FLASH externa de 256K x 16 bits.
- Interfaz JTAG de prueba para la emulación del controlador bus.
- Controlador de puerto paralelo.
- Interfaz de audio para micrófono y bocina.
- Interfaz telefónica que emplea un arreglo de acceso a datos (DAA).
- Conector para expansión de tarjetas externas.
- Puerto de entrada y salida RS232.

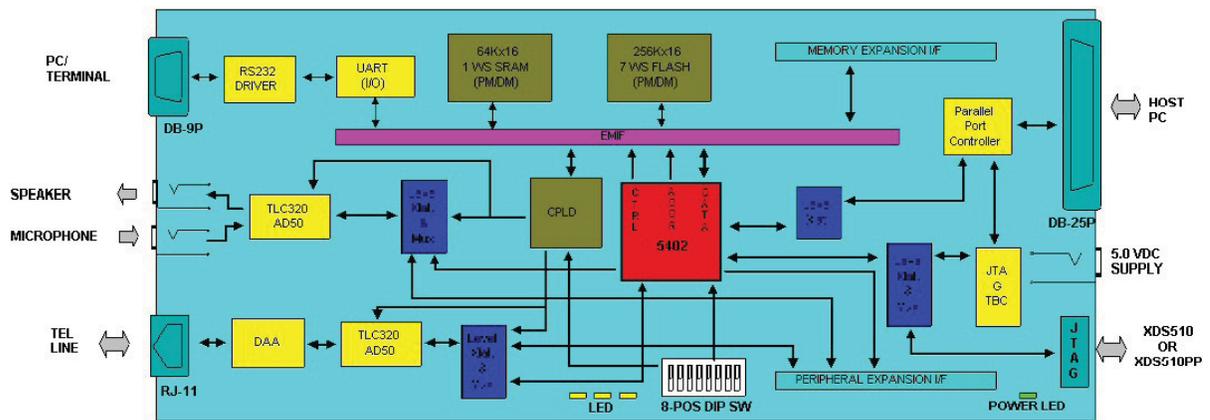


Figura 4.4: Sistema de desarrollo TMS320C5402 DSK [8].

Esta tarjeta cuenta con dos conectores de audio de 3.5 mm para poder conectar el micrófono y la bocina, así como un conector RJ-11 para la interfaz telefónica.

En la figura 4.5 se muestra la interfaz DAA que emplea el circuito integrado CPC5604, este circuito se encarga de manejar los diferentes voltajes presentes en la línea telefónica aislando de manera óptica la línea telefónica del resto de los componentes de la tarjeta.

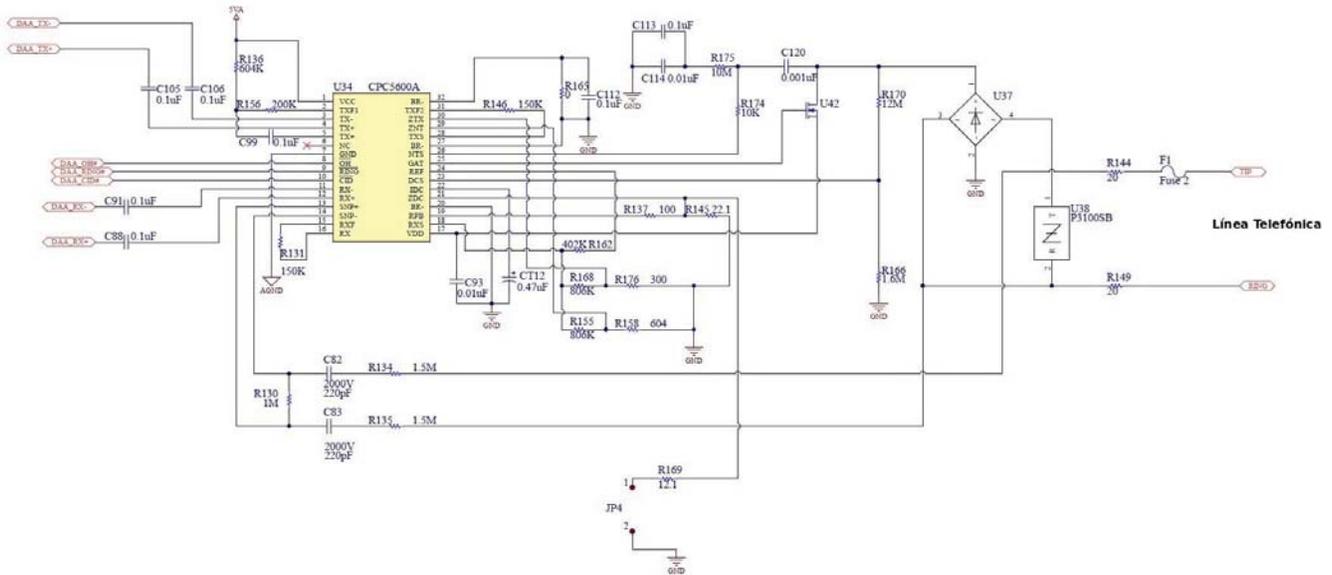


Figura 4.5: Interfaz DAA.

La interfaz DAA se conecta al códec TLC320AD50 (figura 4.6) el cual se encarga de convertir las señales analógicas provenientes de la interfaz DAA en señales digitales, que serán interpretadas por el DSP, y en convertir las señales digitales del DSP en señales analógicas que serán enviadas a la línea telefónica a través de la interfaz DAA. Este códec está conectado al puerto serial bufereado multicanal (McBSP) cero del DSP a través de cuatro líneas.

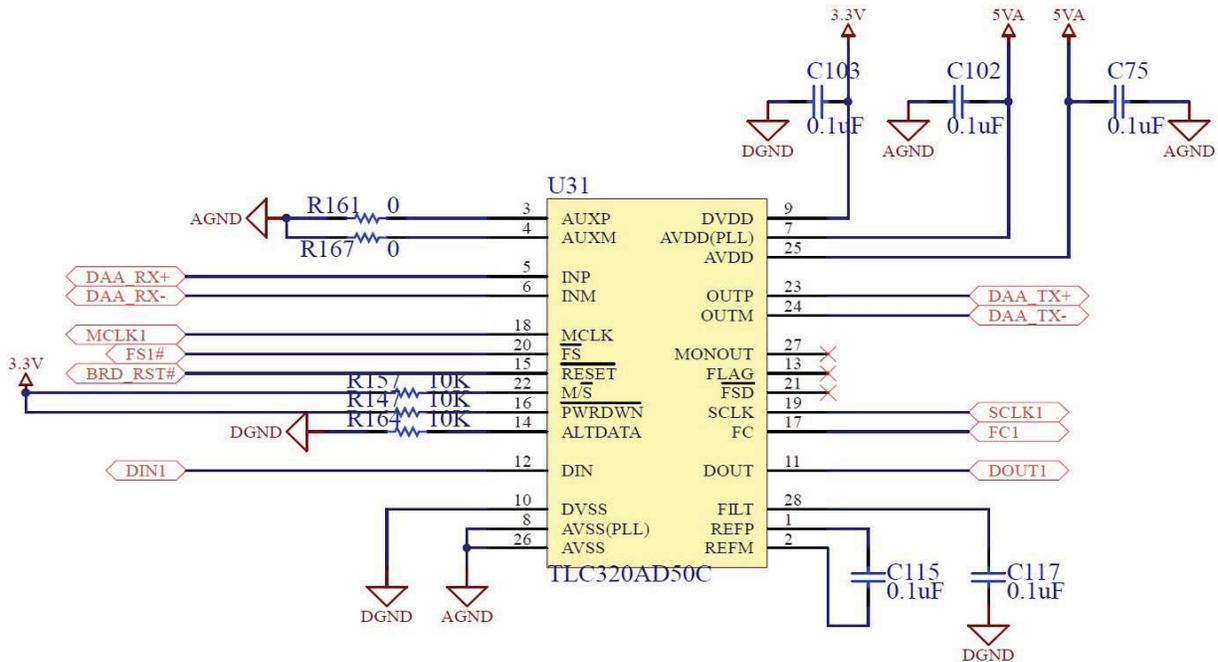


Figura 4.6: Códec para la interfaz DAA.

Como se mencionó, la tarjeta de desarrollo tiene una interfaz de audio por la cual es posible conectar un micrófono y una bocina a la tarjeta mediante dos conectores de 3.5 mm, el conector de entrada tiene una etapa de amplificación con la cual se obtiene un voltaje diferencial, como se muestra en la figura 4.7.b, esta señal es enviada al otro códec TLC320AD50, el cual está conectado al puerto McBSP uno, la entrada de micrófono esta diseñada para micrófonos electret, los cuales requieren un voltaje de *bias* para poder funcionar, este voltaje es filtrado empleando capacitores.

El conector de salida se encuentra conectado a un amplificador de audio LM386 cuya entrada proviene del códec, figura 4.7.c.

Las ganancias de las señales de entrada y salida de audio pueden ser controladas mediante software desde -12 dB a 0 dB en intervalos de 6 db, todo esto con una reducción de señal a ruido de 85 dB [8], esta interfaz también cuenta con un filtrado pasivo entre los conectores y el códec.

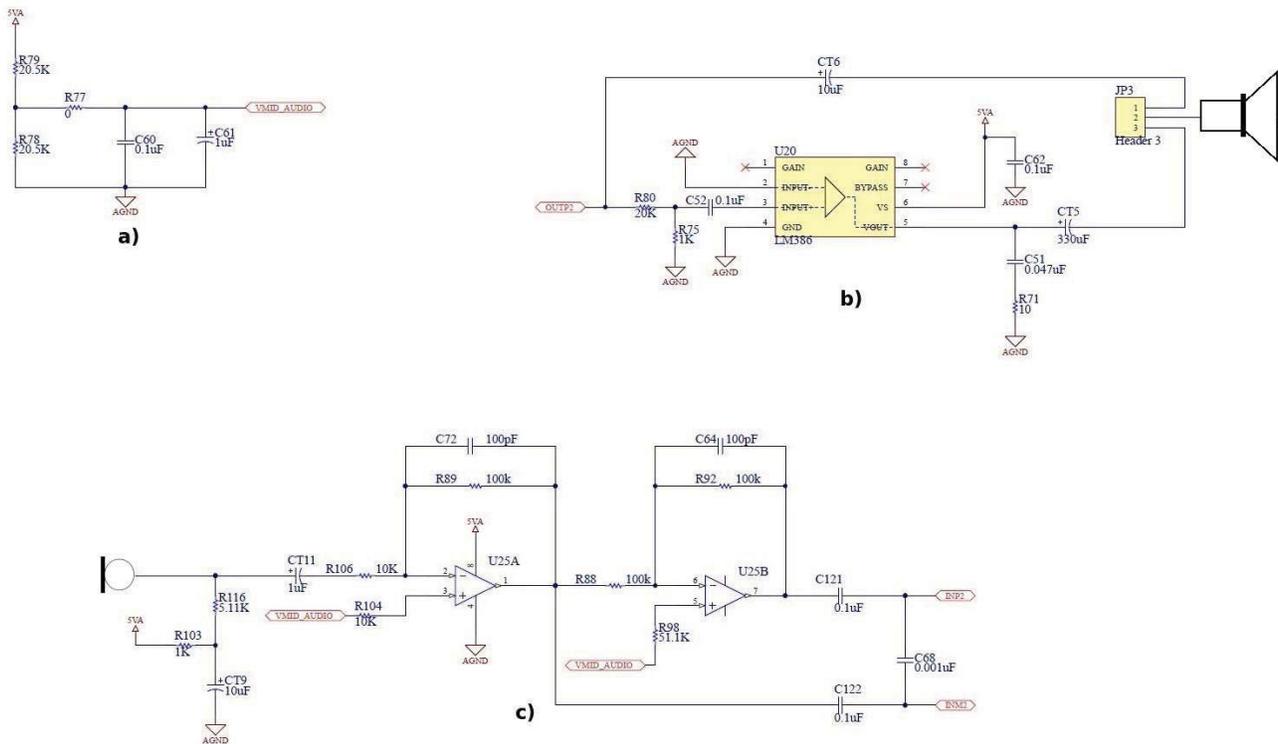


Figura 4.7: Circuito de micrófono y bocina.

La parte fundamental de esta tarjeta de desarrollo es el DSP, éste es el dispositivo al cual se conectan, de forma directa o indirecta, todos los periféricos de la tarjeta y es parte central de esta tesis, por lo que sus características se verán más a fondo en la siguiente sección.

4.2.1 Características generales del DSP TMS320C5402

Un DSP es un microcontrolador diseñado específicamente para el procesamiento digital de señales en tiempo real, esto implica el realizar el proceso deseado en un tiempo menor al tiempo de muestreo, para alcanzar el tiempo real es necesario realizar operaciones numéricas a muy alta velocidad, esto se logra trabajando en paralelo mediante *pipeline* para poder obtener datos y ejecutar las instrucciones de forma simultánea, además cuentan con sumadores y multiplicadores.

La técnica de procesamiento en paralelo *pipeline*, consiste en dividir una tarea en diferentes etapas sucesivas, de tal manera que se pueda comenzar otra tarea antes de que la tarea anterior sea completada [13].

Otras de las características que deben de tener los DSPs, además de poder manejar y procesar altos volúmenes de información en tiempo real, es el tener poderosos modos de direccionamiento, poder transferir de forma rápida información desde y hacia los periféricos.

La operación más importante que debe de realizar un DSP es la multiplicación y acumulación (MAC) en un ciclo de reloj, esta es la operación básica del procesamiento digital de señales para poder realizar convoluciones, correlaciones, filtros digitales y transformaciones matemáticas [14].

Una diferencia entre arquitecturas de DSP es si manejan aritmética de punto fijo o de punto flotante, las arquitecturas de punto fijo almacena los números en un formato entero, estos dispositivos son, usualmente, más baratos, más rápidos y de menor consumo de potencia que los de punto flotante, ya que tienen menos bloques internamente [3]; las arquitecturas de punto flotante, por otro lado, permiten manejar un intervalo dinámico muy grande, estos son especialmente útiles cuando las señales tienen un intervalo dinámico muy amplio o donde se requieren estructuras muy grandes de memoria, como en el procesamiento de imágenes, estos dispositivos son eficientes al ser utilizados con compiladores “C” de alto nivel.

El TMS320C5402 es un DSP de punto fijo perteneciente a la familia C54x de Texas Instruments, esta familia puede realizar entre 100 y 800 MIPS (millones de instrucciones por segundo) con un consumo de energía muy bajo, se emplean en aplicaciones como reproductores de música, VoIP, sistemas de localización satelital y equipos médicos portátiles [15].

Algunas de las características más importantes de este DSP son [16]:

- Arquitectura multibus avanzada con tres buses separados para memoria de datos de 16 bits y un bus para memoria programa.
- Unidad aritmética lógica (ALU) de 40 bits, que incluye un bloque de corrimiento y dos acumuladores de 40 bits.
- Multiplicador en paralelo de 17x17 bits acoplado a un sumador de 40 bits para poder

- realizar, en un solo ciclo, la multiplicación acumulación (MAC).
- Unidad de comparación, selección y almacenamiento (CSSU) para la suma y comparación para el algoritmo Viterbi.
- Dos generadores de dirección con ocho registros auxiliares y dos unidades aritméticas de registros auxiliares.
- Buses de datos con características de retención.
- Modo de direccionamiento extendido hasta 1M x 16 bits.
- 4K x 16 bits de ROM interna.
- 16K x 16 bits de DARAM.
- Instrucciones para el movimiento de bloques de memoria y manejo eficiente de datos y programa.
- Instrucciones con operandos de 32 bits.
- Regreso rápido de interrupciones.
- Modos de consumo de baja potencia con instrucciones IDLE1, IDLE2 e IDLE3.
- Velocidad de 100MHz (100 MIPS).
- Seis niveles de pipeline: pre-búsqueda, búsqueda, decodificación, acceso, lectura y ejecución.

Este DSP cuenta con los siguientes periféricos internos:

- Generador de estados de espera programados por software y un banco de interrupción programable y un banco de interrupción programable.
- Circuito de malla de fase amarrada (PLL) programado por software.
- Generador de relojes con oscilador interno o externo.
- Dos puertos seriales bufereados multicanal (McBSPs).
- Interfaz de puerto paralelo tipo huésped de 8 bits (HPI8).
- Dos temporizadores de 16 bits.
- Seis canales de acceso directo a memoria (DMA).

En la figura 4.8 se muestra la arquitectura del DSP TMS320C5402, éste usa una arquitectura tipo Harvard que optimiza el procesamiento por medio de sus tres buses independientes de memoria dato y un bus de memoria programa, al separar la memoria programa y la memoria dato se puede tener acceso simultáneo a las instrucciones del programa y datos teniendo un alto grado de paralelismo [8].

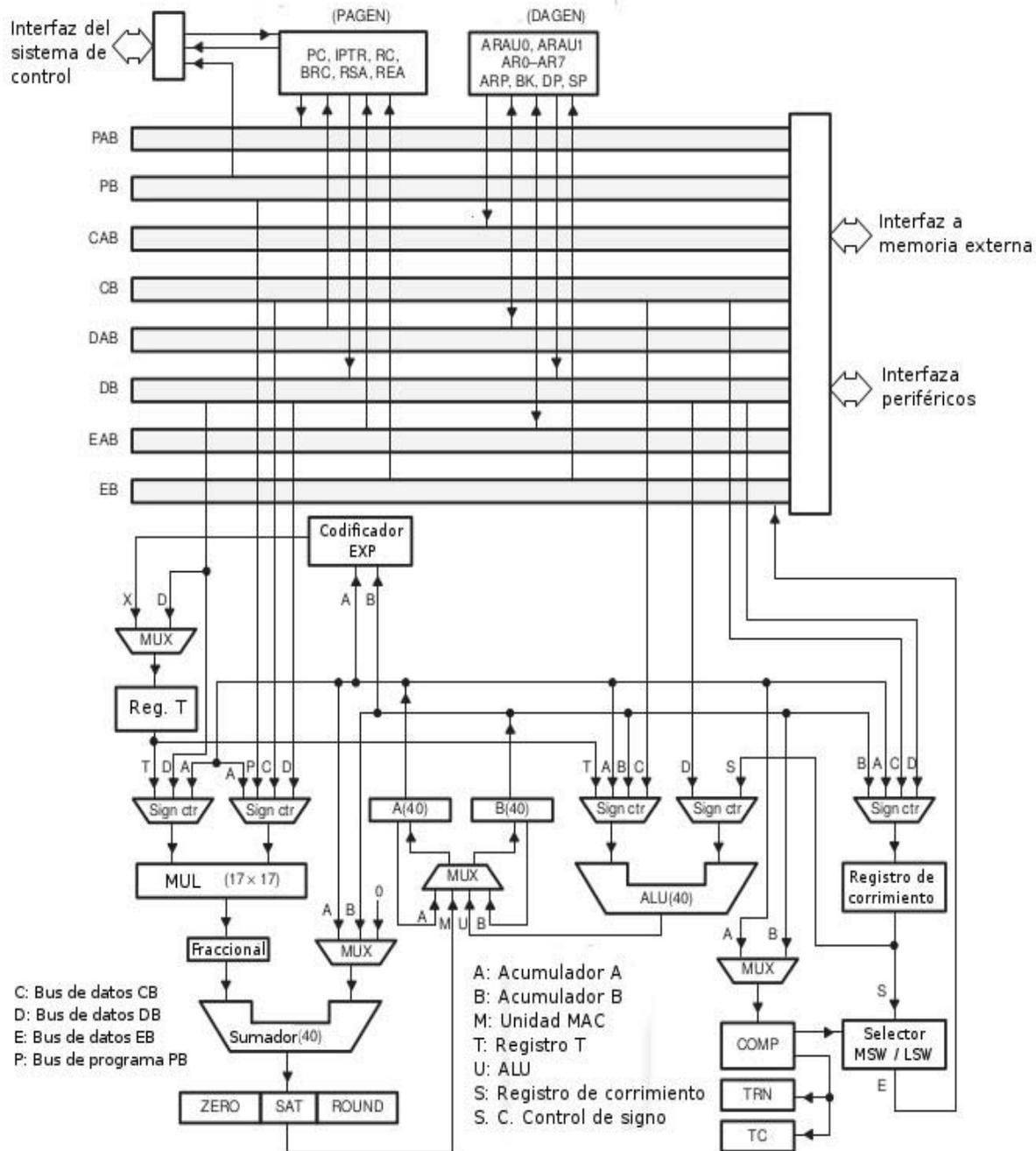


Figura 4.8: Diagrama a bloques del TMS320C5402 [17].

Esta arquitectura cuenta con ocho buses de 16 bits, cuatro para datos y programa (PB, CB, DB, EB), y cuatro para direcciones (PAB, CAD, DAB, EAB).

De acuerdo a sus características, se puede apreciar que el TMS320C5402 tiene las características necesarias para realizar la síntesis de voz en tiempo real, y la tarjeta de desarrollo cuenta con todos los elementos para la realización del sistema permitiendo reducir la cantidad de componentes adicionales.

4.3 HARDWARE

El hardware empleado en este trabajo consiste en una tarjeta de expansión que se montó sobre la tarjeta de desarrollo, en ella se agregaron los sensores de puertas y ventanas, así como el interruptor del interfón.

La comunicación entre la tarjeta de expansión y la evaluación se realizó a través de los conectores J3 y J9, de la tarjeta de evaluación, que corresponden a las interfaces de expansión de memoria y de periféricos, respectivamente, estos se muestran en la figura 4.9.

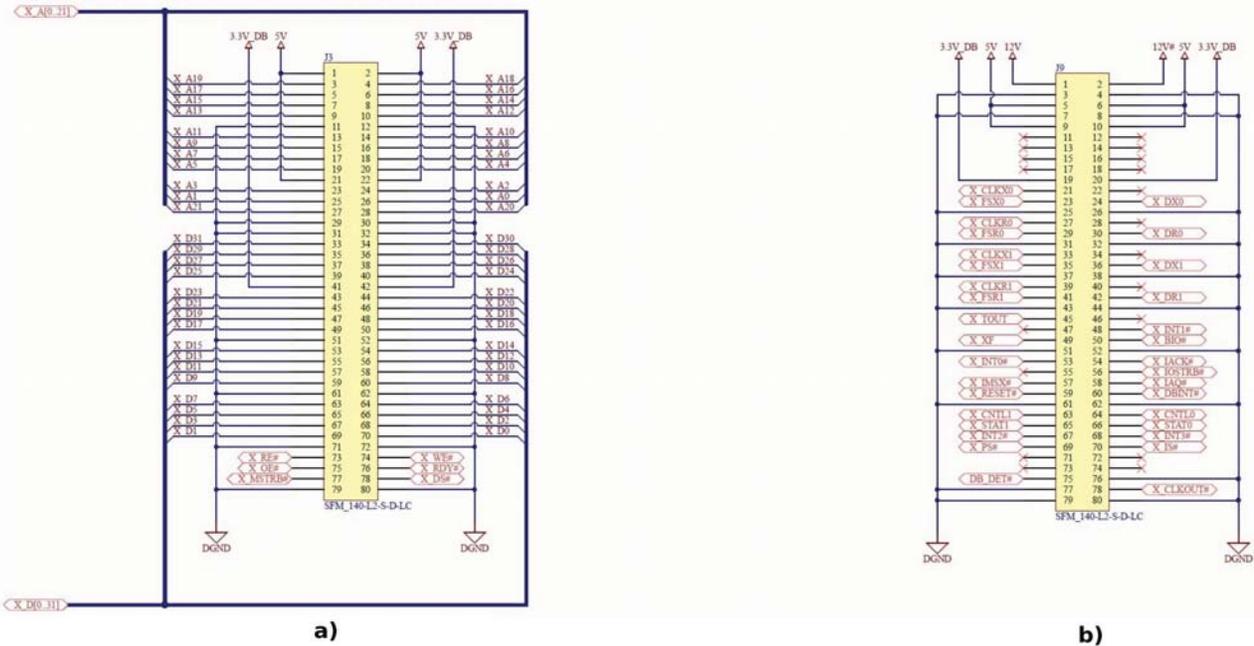


Figura 4.9: Interfaz de expansión de memoria y periféricos.

Estos son conectores de la marca Samtec, con número de parte SFM-140-Ñ2-S-D-LC, por esta razón se emplearon los conectores TFM-140-32-S-D-LC de la misma marca, como se puede apreciar de la figura anterior, los conectores cuentan con salidas de voltaje que se pueden emplear para polarizar la tarjeta de expansión.

En la figura 4.9 se muestran los pines de la interfaz de expansión de memoria a los cuales fueron conectados los sensores, estos están señalados en la figura 4.9.a como X_D0 a X_D7 y corresponden a los pines 70 al 63 y los pines 2 y 80 para polarizar la tarjeta de expansión, siendo estos pines 5V y DGND respectivamente. Se emplearon *push buttons* para emular los sensores ubicados en puertas y ventanas, estos fueron interruptores normalmente cerrados (NC), conectados como se muestra en la figura 4.10, se puede apreciar que mientras el interruptor se mantiene cerrado (no se detecta apertura de puerta o ventana) el voltaje que llega al pin del conector es de 0V, y una vez que el interruptor se abre el voltaje es de 5V, el cual representa un “1” lógico.

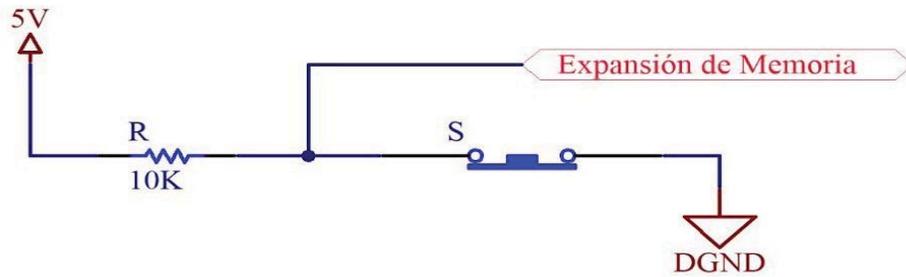


Figura 4.10: Diagrama de conexión de los interruptores.

Para el circuito del interfón se empleó un circuito similar pero se conectaron un micrófono y un altavoz, para que la persona que accionó el timbre pueda establecer comunicación, estos se conectaron a la tarjeta de evaluación mediante los conectores de audio de 3.5mm que se muestran en la figura 4.4.

4.4 SOFTWARE

El software desarrollado para este trabajo se dividió en dos partes ya que fue necesario hacer un análisis fuera de línea para poder obtener los parámetros necesarios para la generación de la señal de voz y una vez obtenidos estos, se realiza la síntesis en tiempo real de acuerdo con el interruptor que haya sido activado.

4.4.1 Análisis fuera de línea

Como se mostró en el capítulo 2, la síntesis de voz se logra mediante la extracción de parámetros de una señal de voz, en el caso de este trabajo esto se realizó mediante un programa en Matlab[®] que obtenía una señal de voz original a partir de un archivo .wav y generaba un archivo .dat con un formato especial para que dichos parámetros puedan ser guardados en la memoria interna del DSP, este procedimiento se muestra en la figura 4.11.

Lo primero que se requiere para esto es tener una señal de voz real, la cual se pasa por el filtro de preénfasis para después ser segmentada en M número de ventanas con un 128 de muestras, quedando cada ventana de 16 ms; una vez que se tiene el número de ventanas es necesario identificar si se trata de una trama de silencio o de voz, en caso de ser considerada como silencio, en caso contrario se procede a aplicar la ventana Hamming a cada ventana, posteriormente obtener el vector de autocorrelación de dicha ventana.

Una vez que se tiene el vector de autocorrelación se puede obtener la matriz Toeplitz para obtener los coeficientes LPC de la ventana actual, después es necesario obtener el periodo de pitch, esto se hace con el método de recorte central. Todo el proceso se repite para las M ventanas de la señal, una vez obtenidos todos los parámetros necesarios se guardan en un archivo .dat de tal manera que puedan ser guardados en la memoria del DSP, en la figura 4.11 se muestra el diagrama de flujo de este programa.

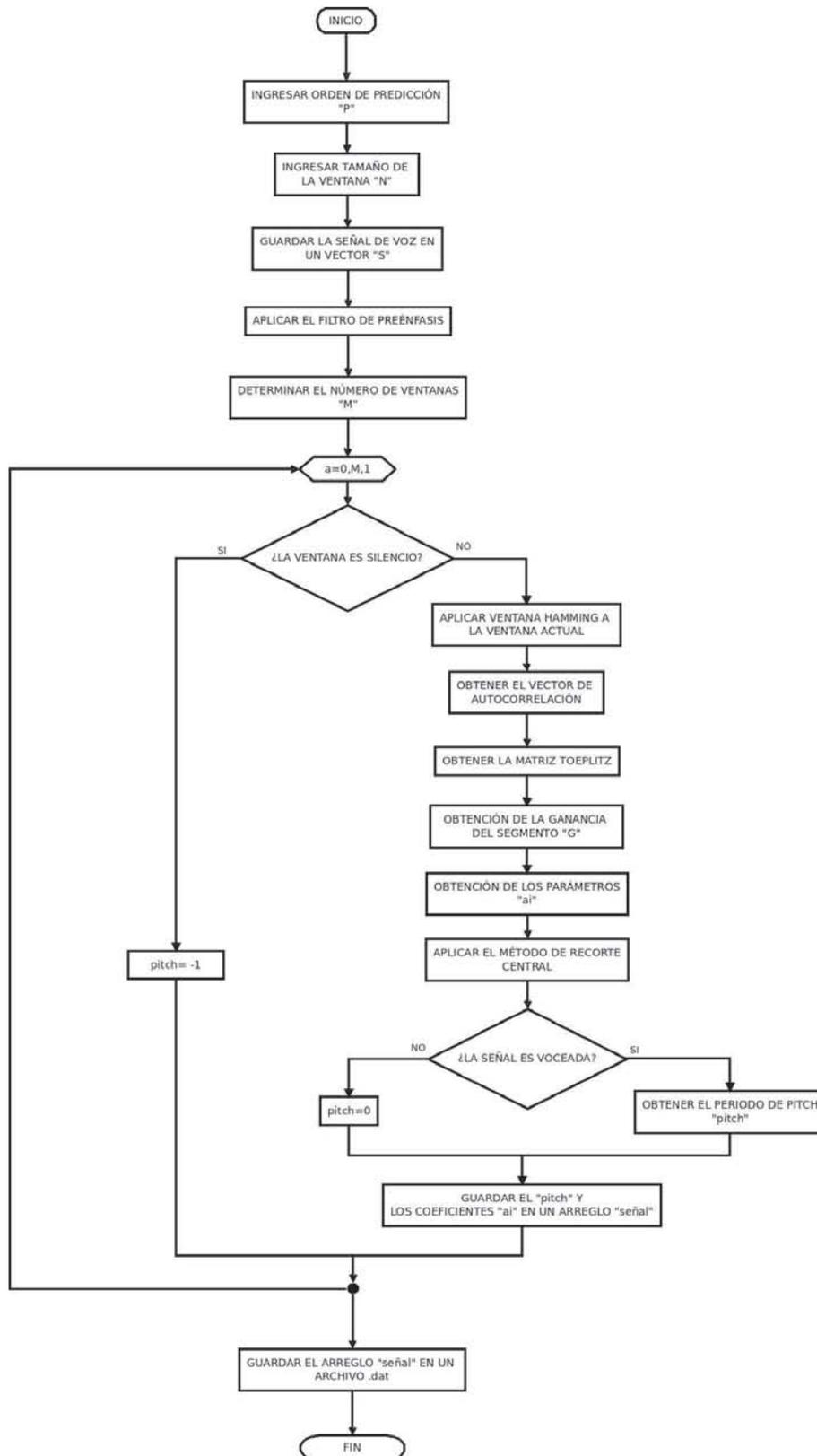


Figura 4.11: Diagrama de flujo del programa de MATLAB®.

Los datos se guardan en la memoria del DSP como un arreglo, de tal manera que en la primera posición del arreglo se guardan el número total de ventanas, M , las siguientes posiciones son el pitch, la ganancia del segmento y los coeficientes LPC, en caso de que la trama no sea silencio. En el caso de que la ventana sea clasificada como silencio se le asigna el valor de -1 al periodo de pitch y se procede a analizar la siguiente ventana, en caso contrario se determina si la señal es voceada o no voceada, si la trama se clasifica como voceada se guarda el periodo de pitch, si se clasifica como no voceada el pitch vale 0, esta estructura se muestra en la Tabla 4.1.

Tabla 4.1

Encabezado	1651 2 8000 1 ###
Número de ventanas	M
Valor del pitch	-1 = silencio, 0 = no voceada, >0 = voceada
Ganancia	G
Coefficientes	aj1, aj2, aj3, aj4, ..., ajn

*### número de datos en el archivo

Una vez obtenido dicho archivo es posible guardarlo en la memoria del DSP para posteriormente poder realizar la síntesis de la señal de voz.

4.4.2 Procesamiento en tiempo real

El DSP será el encargado de realizar las funciones de colgado y descolgado, así como generar los tonos DTMF necesarios para marcar el número telefónico y realizar la síntesis de voz, dependiendo del interruptor que se haya activado. Todas estas funciones deben de ser realizadas en tiempo real, además deben de cumplir con las características enunciadas en el capítulo 3, en la figura 4.12 se muestra el diagrama de flujo de la generación de tonos DTMF empleando dos osciladores recursivos, donde j es el número de dígitos a marcar y $aj1$, $aj2$, $bj1$ y $bj2$, son los parámetros de los tonos alto y bajo de las señales DTMF (ver sección 3.6.1), como se muestran en la ecuación 4.1.

$$\begin{aligned}
 a_{j1} &= \sin(\omega_L) \\
 a_{j2} &= \sin(\omega_H) \\
 b_{j1} &= 2 * \cos(\omega_L) \\
 b_{j2} &= 2 * \cos(\omega_H)
 \end{aligned}
 \tag{4.1}$$

Para poder realizar la llamada telefónica se emplearon las APIs que nos da el fabricante, dentro de éstas se encuentra funciones especiales para poder manejar el códec DAA de la tarjeta de desarrollo.

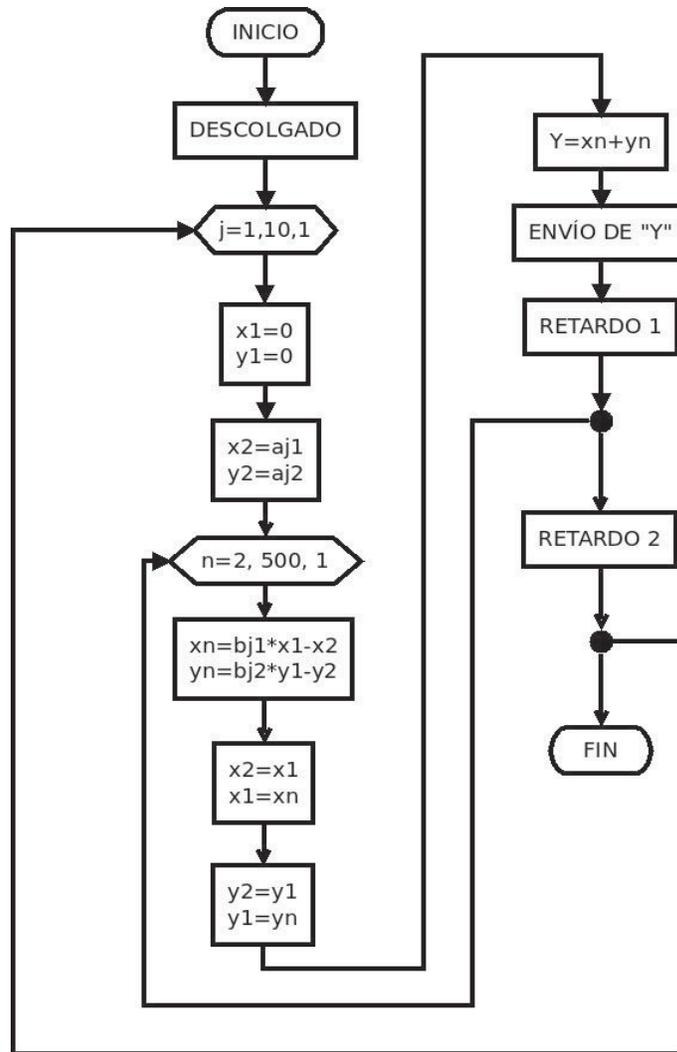


Figura 4.12: Diagrama de flujo de la generación de tonos DTMF.

Se generan 500 muestras (62.5 ms) por cada dígito a marcar, así como un retardo entre cada dígito, los cuales se emplean para que las señales tengan la duración necesaria para ser interpretadas por la central telefónica.

Una vez que se ha marcado el número telefónico es necesario identificar cual fue el interruptor que se activó, una vez que identificada la fuente es necesario recuperar la información guardada en la memoria del DSP correspondiente a la señal de alerta, el proceso para recuperar los datos se muestra en la figura 4.13.

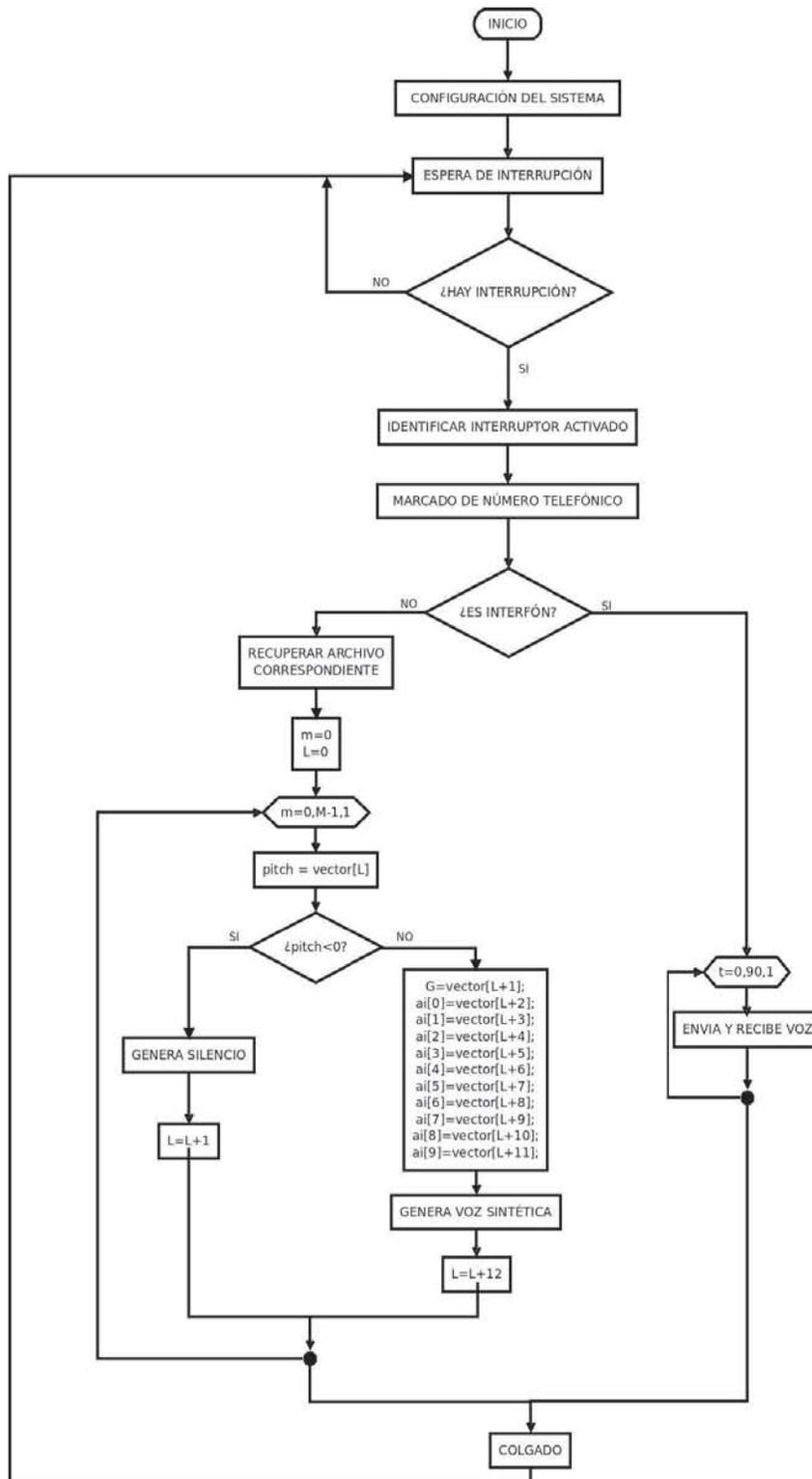


Figura 4.13: Diagrama de flujo de la recuperación de datos.

De la figura 4.13 se puede apreciar que en cuanto los datos del archivo han sido recuperados se procede a realizar la síntesis de voz, para esto se toma el valor del periodo de pitch para determinar la naturaleza del segmento, voceado o no voceado, y con esto poder generar la excitación necesaria para dicho segmento de voz, en la figura 4.14 se muestra el diagrama de flujo de la síntesis de voz, el cual corresponde a la ecuación (2.19).

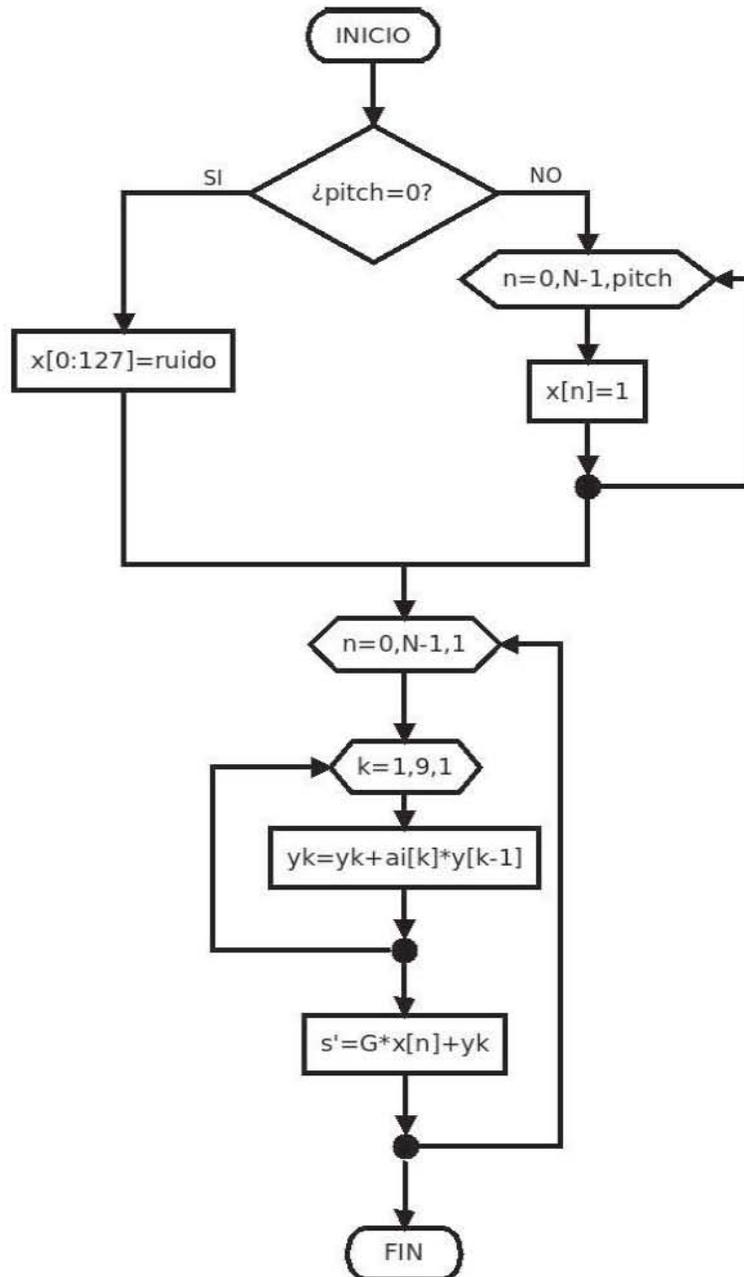


Figura 4.14: Diagrama de flujo de la síntesis de voz.

El diagrama mostrado en la figura 4.14 se repite para cada ventana de la señal de voz a generar, y el valor de “s” se envía a través de la línea telefónica con una frecuencia igual a la de muestreo, esta frecuencia se logra mediante el uso del timer del DSP.

Una vez que se finaliza la llamada telefónica el sistema vuelve al estado inicial para esperar que ocurra otro evento.

4.4 Resumen

En este capítulo se describió el funcionamiento del sistema, tanto en hardware como en software, el cual se implementó usando un procesador digital de señales (DSP), este es un microprocesador especialmente diseñado para realizar operaciones matemáticas intensivas de tal manera que pueda realizar el procesamiento digital de señales en tiempo real, una de las características más importantes de este procesador es la capacidad de realizar la operación MAC (multiplicación acumulación) en un ciclo de reloj, lo que permite que se puedan realizar filtros digitales, correlaciones, convoluciones, transformadas matemáticas, etc. También se mostraron las características de la tarjeta de desarrollo la cual cuenta con una interfaz telefónica y una interfaz de audio, las cuales fueron empleadas en este sistema.

Capítulo 5

Pruebas y resultados

En este capítulo se muestran los resultados obtenidos en el presente trabajo, dando especial énfasis en la síntesis de voz y en la generación de los tonos DTMF, así mismo se analizan la cantidad de recursos que se emplean de la tarjeta de desarrollo y del DSP.

En el caso de la voz sintética se compara la calidad de la señal al modificar algunos parámetros al realizar el análisis de la señal de voz, y de esta manera poder ver como éstos influyen en la calidad de la señal de voz,

5.1 EVALUACIÓN DE LA VOZ GENERADA

Este sistema se basa en el aviso al usuario de la activación de alguno de los sensores, esto se hace mediante el uso de voz sintética, por lo que es necesario que la señal de voz sea inteligible al ser transmitida por la línea telefónica, por eso es necesario evaluar la calidad de la voz generada.

Es difícil determinar la calidad de la señal de voz ya que en su mayoría se busca que sea inteligible para las personas, y este es un aspecto cualitativo, por lo que para poder realizar un análisis más cuantitativo de la calidad de la señal se comparó el espectro en frecuencia de la señal original con la señal sintética para verificar que tanto se aproximan.

La calidad de la voz generada depende básicamente del tamaño de la ventana a procesar, del número de coeficientes LPC empleados y de la calidad de la señal de voz original, en las figura 5.1 a 5.3 se muestran señales de voz sintética para diferente número de muestras, desde $N=512$, 256 y 128, respectivamente para la palabra “ventana” y con $p=10$.

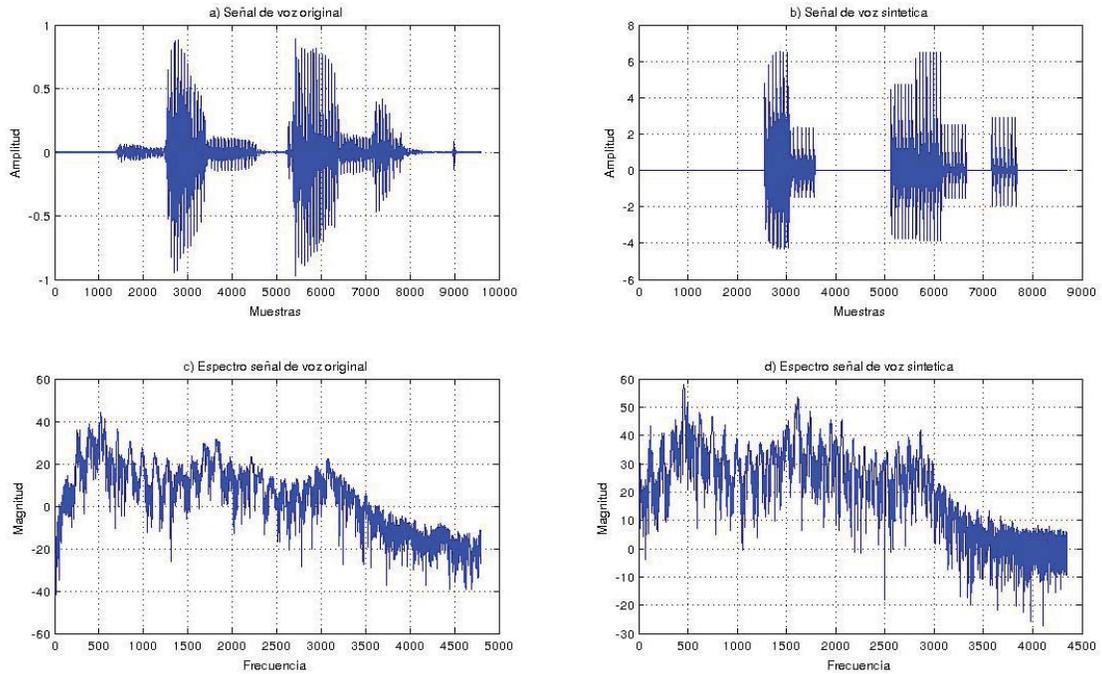


Figura 5.1: Señal a)Original, b)Sintética, c)Espectro original, d)Espectro sintética, $N=512$, $p=10$.

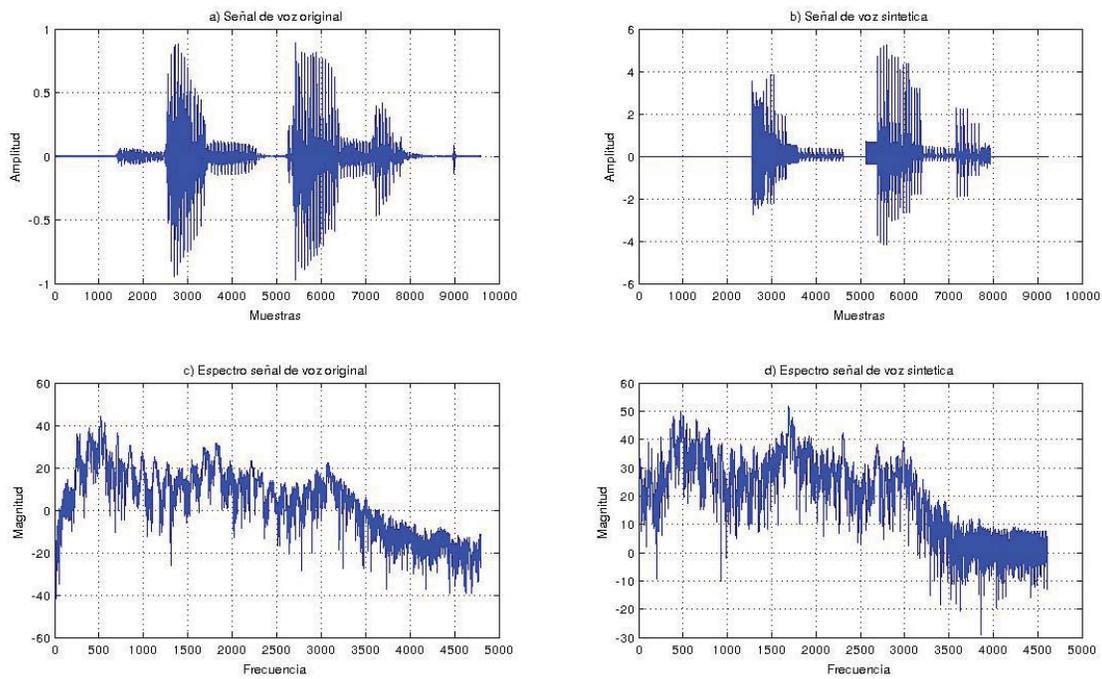


Figura 5.2: Señal a)Original, b)Sintética, c)Espectro original, d)Espectro sintética, $N=256$, $p=10$.

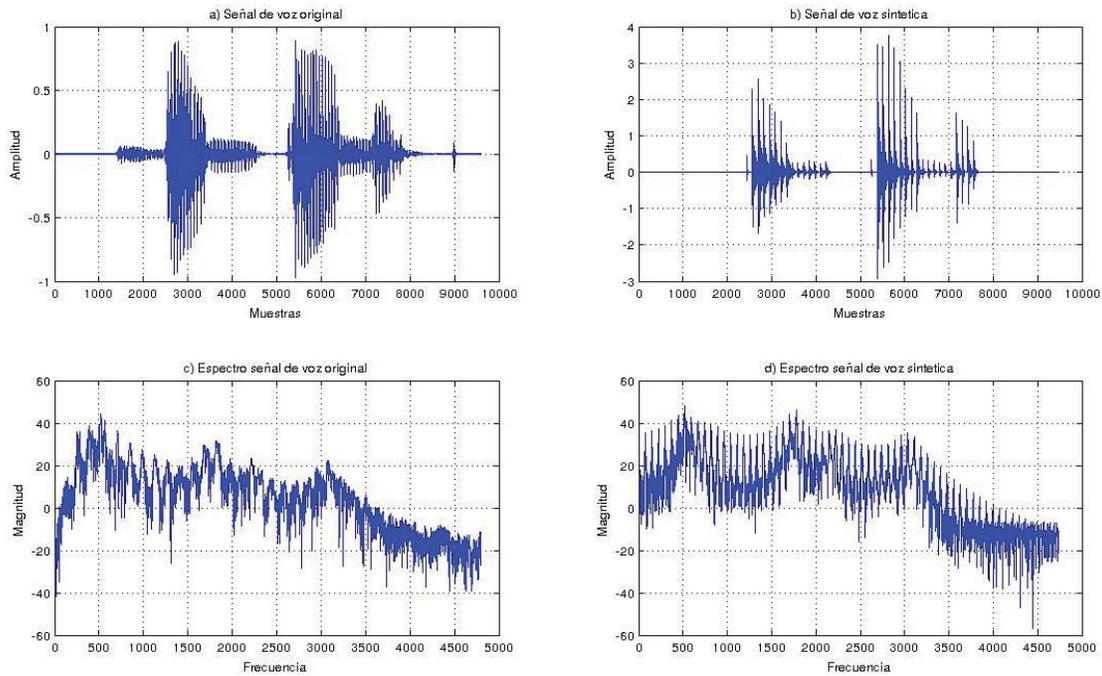


Figura 5.3: Señal a)Original, b)Sintética, c)Espectro original, d)Espectro sintética, $N=128$, $p=10$.

Como se puede observar el número de muestras interviene en la calidad de la voz generada, se debe de tener en cuenta que el tamaño de la ventana debe mantenerse dentro del intervalo de 10 a 30 ms [8], esto debido a que si la ventana es muy pequeña, menor al periodo de pitch, se perderá información en frecuencia, y por otro lado, si el tamaño de la ventana es muy grande, se perderá información en el tiempo [6], el número de muestras elegido fue de 128, lo que implica una ventana de 16ms, empleando este valor se obtuvieron señales de voz más inteligibles.

Otro de los factores a considerar a la hora de generar la voz sintética es el orden de predicción, por lo que se realizaron pruebas modificando este parámetro, los resultados se muestran en las figuras 5.4, 5.5 y 5.6, para ordenes de predicción de 5, 10 y 20, respectivamente y con un número de muestras de 128 para la palabra "ventana".

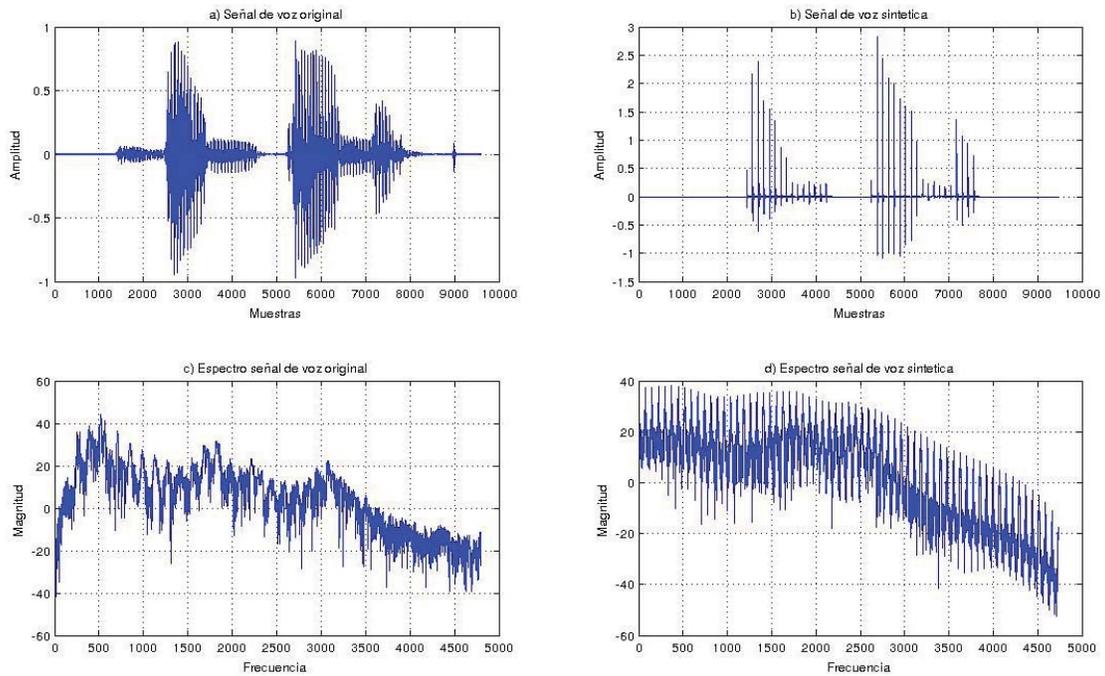


Figura 5.4: Señal a)Original, b)Sintética, c)Espectro original, d)Espectro sintética, $N=128$, $p=5$.

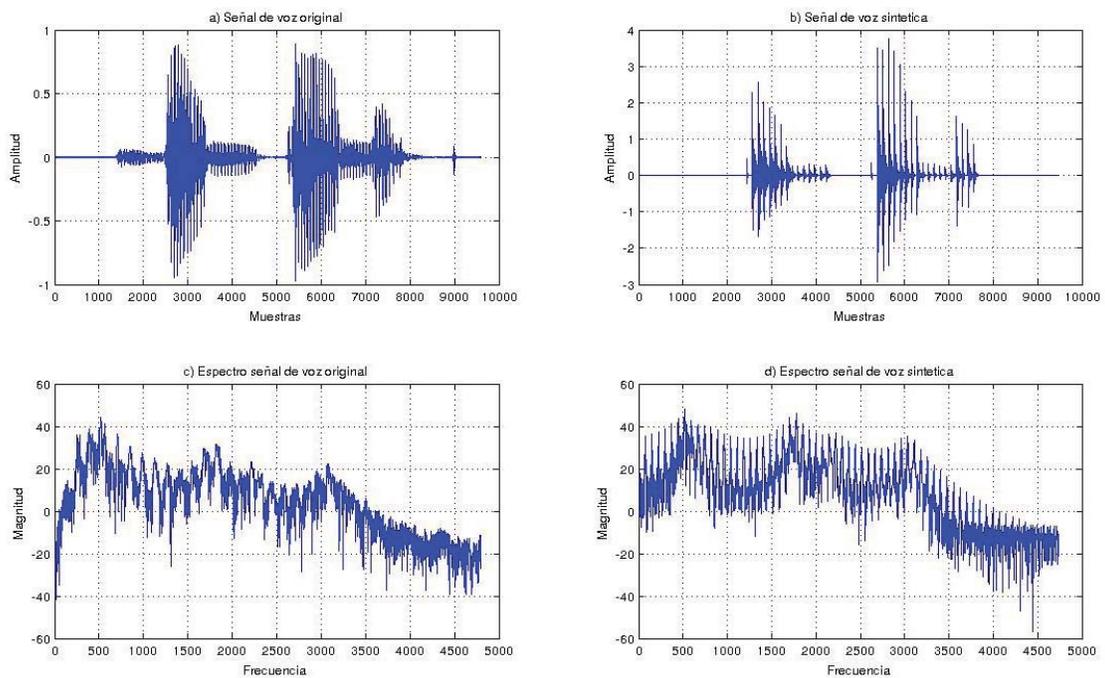


Figura 5.5: Señal a)Original, b)Sintética, c)Espectro original, d)Espectro sintética, $N=128$, $p=10$.

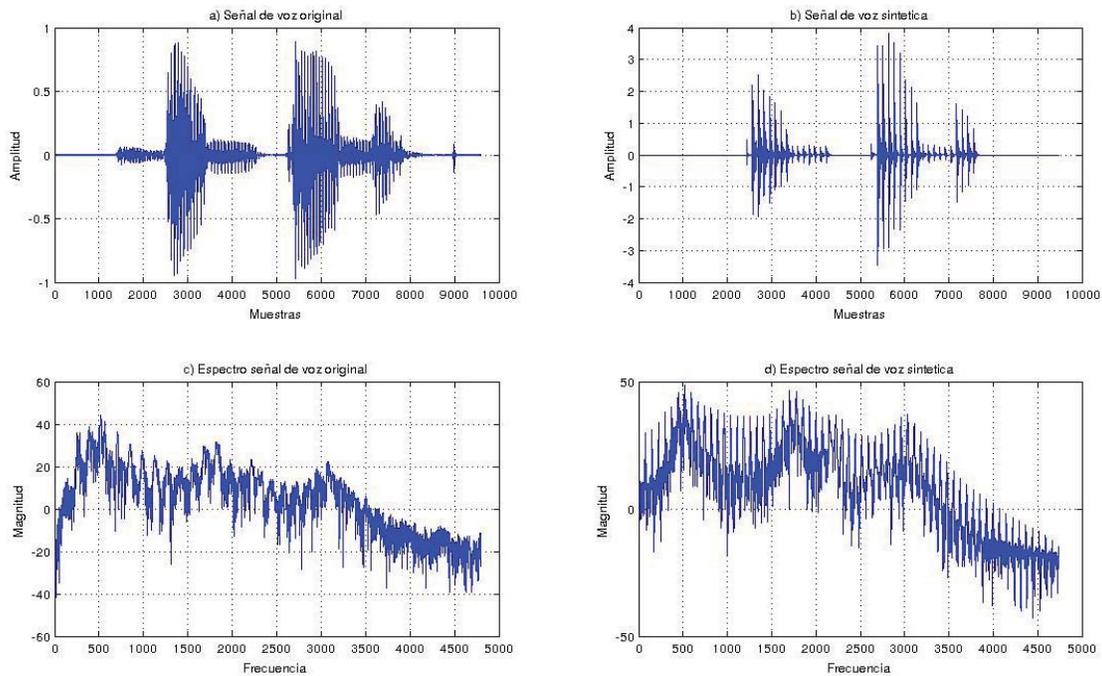


Figura 5.6: Señal a)Original, b)Sintética, c)Espectro original, d)Espectro sintética, $N=128$, $p=20$.

De las figuras 5.4, 5.5 y 5.6 se puede apreciar que un orden de predicción muy bajo genera una señal de calidad muy baja, mientras que para ordenes de predicción altos, mayores a 10, la calidad no varía de manera significativa, cabe destacar que el orden de predicción tiene mayor influencia sobre la calidad de la voz que el número de muestras tomados, debido a esto en este trabajo se empleó un orden de predicción de 10 y se trabajó con 128 muestras, ya que con estos valores se obtuvo una calidad de voz aceptable.

En la figura 5.7 se muestra el espectrograma de la señal de voz original comparada con la voz producida, se puede observar, que en la señal sintética el contenido de frecuencias altas es mayor que en la señal original, pero en ambas señales la mayor energía se presenta en bandas de frecuencias similares, por lo cual la señal es inteligible.

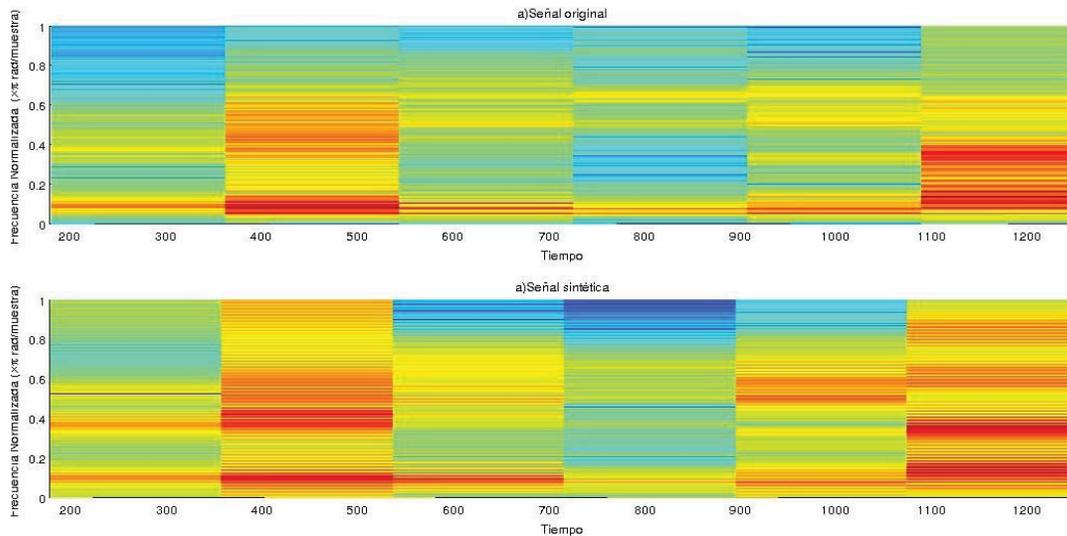


Figura 5.7: Espectrograma de la señal de voz "principal" a)Original, b)Sintética, $N=128$, $p=10$.

Cabe mencionar que la señal generada fue transmitida a través de la línea telefónica, y a pesar de esto se obtiene una señal de voz con calidad aceptable e inteligible para el usuario, de tal manera que el mensaje se puede entregar al usuario de manera correcta, lo mismo sucede con las señales de voz adquiridas, así mismo, la calidad de las señales de voz transmitidas desde el interfón al teléfono del usuario, y viceversa, fue similar a la de un teléfono fijo.

5.2 EVALUACIÓN DE LOS TONOS DTMF GENERADOS

Otro punto importante dentro del trabajo es poder establecer la llamada telefónica con la cual se comunica al usuario que se ha presentado un evento de interés, para esto es necesario que la generación de tonos DTMF cumpla con el estándar que se mencionó en el capítulo 3. Como se mencionó en dicho capítulo los tonos se generaron empleando dos generadores de señales senoidales, en la figura 5.8 se muestra un ejemplo de los tonos generados para el dígito "uno", estas frecuencias son 697 Hz para el tono bajo y 1209 Hz para el tono alto. En la figura 5.9 se muestra el espectro en frecuencia de dichos tonos.

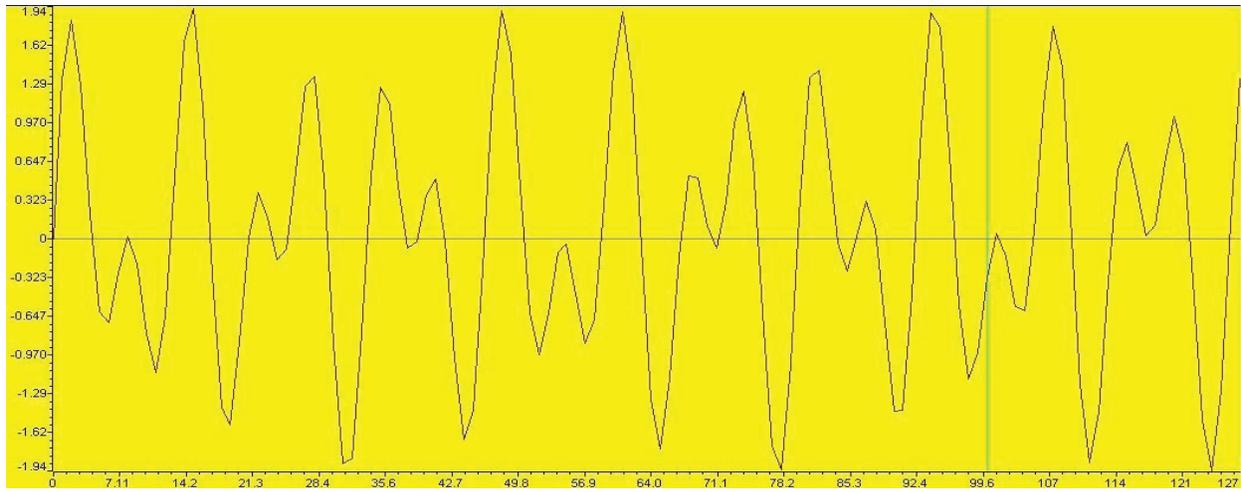


Figura 5.8: Tono DTMF correspondiente al número 1.

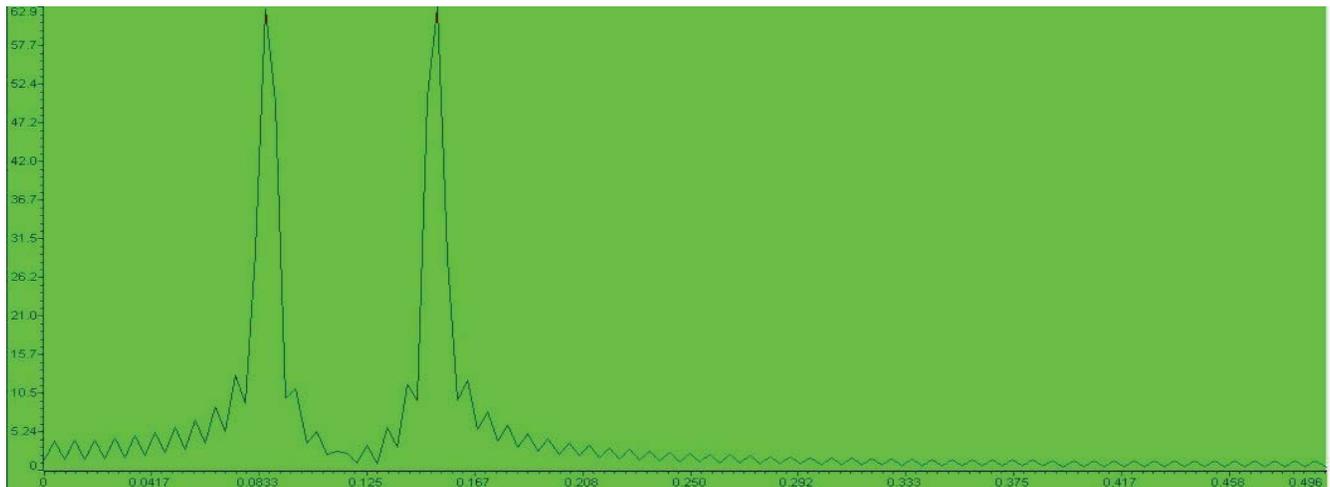


Figura 5.9: Espectro de tono DTMF correspondiente al número 1.

En la figura 5.10 se muestra un ejemplo de como se envían una secuencia de tonos DTMF a través de la línea telefónica al marcar un número telefónico, se buscó que las características de dichos tonos cumplieran con el estándar DTMF.

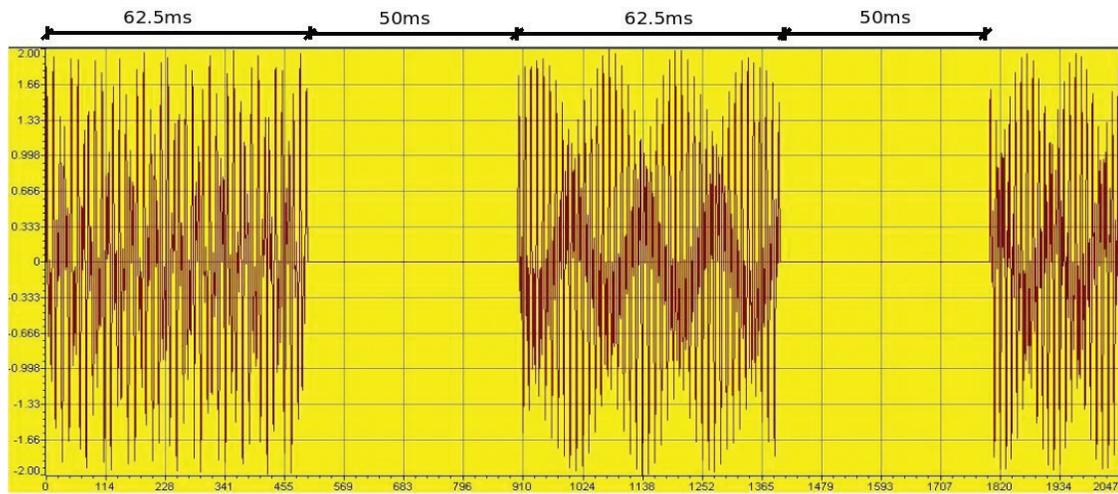


Figura 5.10: Ejemplo de varios tonos DTMF, correspondiente a los dígitos 1, 2 y 3.

Cabe mencionar que el sistema siempre pudo enlazar la llamada telefónica con el teléfono celular del usuario, de tal manera que se puede asegurar que los tonos DTMF siempre fueron generados de manera correcta y que sus características cumplieran con las establecidas en el protocolo DTMF.

5.3 EVALUACIÓN DEL HARDWARE

Al realizar un sistema de procesamiento digital de señales es importante analizar los recursos empleados para poder saber si los procesos se están realizando de manera correcta.

Una de las características de la síntesis de voz empleando predicción lineal es el nivel de compresión que se logra, esto es debido a que es necesario almacenar pocos parámetros para poder realizar la reconstrucción de la señal original. En la tabla 5.1 se muestran la cantidad de datos que se requieren guardar en la memoria del DSP, así mismo se muestra la cantidad de datos de las señales originales [18].

Tabla 5.1: Memoria empleada para los parámetros LPC.

Frase	Original [Kb]	Sintética [Kb]	% de compresión
Puerta principal abierta	26122	1009	96.13
Ventada cocina abierta	24454	1073	95.62
Ventana comedor abierta	24238	906	96.27
Habitación 1	30664	1077	96.49
Habitación 2	31187	1158	96.29
Habitación 3	29298	1077	96.33
Puerta del patio	21952	789	96.41

De la tabla 5.1 se puede observar que el algoritmo empleado en este trabajo permite recuperar una señal de voz empleando una cantidad de memoria menor al 5% de la que se requeriría si se guardaran las señales originales en la memoria del DSP.

Se midió el tiempo de ejecución del programa, estando dividido en:

- Tiempo de detección de interrupción.
- Tiempo requerido para generar una muestra de la señal DTMF.
- Tiempo necesario para generar una muestra de voz sintética.
- Tiempo requerido para realizar la marcación del número telefónico.

Estos tiempos se midieron en función de la cantidad de ciclos que requería el CPU para realizar las instrucciones necesarias en cada segmento del programa, los resultados se muestran en la tabla 5.2 (la frecuencia de reloj era de 10 MHz).

Tabla 5.2: Ciclos de CPU requeridos [18].

Señal	Ciclos empleados	Tiempo
Detección de interrupción	32	3.2 us
Muestra de DTMF	44	4.4 us
Muestra de voz sintética	901	90.1 us
Marcado del número telefónico	53875482	5.38 s

Debido a que se emplearon señales de voz se consideró una frecuencia de muestreo de 8 KHz, lo que implica que el tiempo entre cada muestra es de $125 \mu s$, en la tabla 5.2 se puede apreciar que el proceso que requiere mayor cantidad de procesos es la síntesis de voz, sin embargo requiere de muy poco tiempo para poder ser ejecutado en comparación con el periodo de muestreo, por lo que el sistema puede operar en tiempo real.

5.4 RESÚMEN

En este capítulo se analizaron los resultados obtenidos en el sistema, se revisaron los parámetros que más influyen en la calidad de la voz sintética, siendo esta una medida cualitativa, así mismo se analizaron los tonos DTMF los cuales cumplen con el estándar establecido ya que siempre que se presentaba un evento era posible enlazar la llamada telefónica. Se analizó el rendimiento del hardware empleado y se observó que los procesos que este lleva a cabo solo requieren de una fracción del total de recursos disponibles, por lo que se podrían agregar más funciones al sistema sin comprometer los procesos actuales.

Capítulo 6

Conclusiones

En este trabajo se diseñó e implementó un sistema de alarma en tiempo real, el cual da aviso al dueño de una vivienda acerca del estado de su propiedad, así mismo, en caso de que se active el interfón establece comunicación entre la persona que lo activó y el usuario, todo esto se hace mediante el uso de la línea telefónica de la casa y el teléfono celular del dueño.

Se hizo uso de voz sintética para dar aviso al usuario sobre el estado de los sensores, para esto se usó un Vocoder LPC, mostrándose su descripción teórica y su implementación dentro del DSP, este método permitió disminuir la cantidad de memoria empleada para los mensajes de alarma, reduciendo esta en más del 95% del tamaño original.

Para hacer uso de la línea telefónica se tuvo que desarrollar un generador de tonos DTMF, el cual se implementó en el DSP, este se hizo mediante el uso de un oscilador recursivo, estos tonos cumplieron con el estándar establecido ya que la llamada pudo ser establecida en todo momento.

El sistema se desempeña de manera correcta en tiempo real, ya que para generar cada muestra de la señal de voz se requiere un tiempo máximo de $90.1 \mu s$, con una frecuencia de reloj de 10 MHz; al emplear una frecuencia de muestreo de 8 KHz el tiempo máximo de procesamiento para que el sistema funcione en tiempo real es de $125 \mu s$, un tiempo por encima del proceso más lento del sistema.

Para enviar y recibir datos a través de la línea telefónica se hace uso de la interfaz RJ11 de la tarjeta de evaluación, con esta interfaz pudo ser posible realizar fácilmente la conexión del sistema con el teléfono del usuario.

Se empleó el DSP con una frecuencia de reloj de 10 MHz, pero este puede funcionar con una frecuencia de hasta 100 MHz, por lo que es posible mejorar el sistema actual para poder realizar un mayor número de tareas, entre ellas se podría implementar un sistema de detección de tonos DTMF el cual podría controlar el estado de las luces de la casa, así como controlar chapas electrónicas en puertas y ventanas.

Bibliografía

- [1] Traister J. E. Security/Fire-Alarm Systems. Mcraw-Hill. New York. 1996.
- [2] Muñoz J. Sistemas de seguridad. Paraninfo. Madrid. 2000.
- [3] Maya L. C. Sistema telefónico con síntesis de voz y detección de tonos. UNAM. FI. DF. 2010.
- [4] Chris R. Speech processing. McGraw-Hill. University of Essex. Londres. 1992.
- [5] Rabiner L. Digital processing of speech signals. Prentice Hall. Englewood cliffs. 1978.
- [6] Deller J. R. Discrete-time processing of speech signals. Prentice Hall. New York. 1993.
- [7] Shondi M. M. New methods of pitch extraction. IEEE Transactions on audio and electroacoustics, p262-266. Junio 1968.
- [8] Barajas P. G. y Molero A. M. Síntesis de voz en tiempo real empleando una arquitectura DSP. UNAM. FI. DF. 2004.
- [9] Papamichalis P. E. Practical approaches to speech encoding. Prentice Hall. New Jersey. 1987.
- [10] Harb M. Modern telephony. Prentice Hall. New Jersey. 1989.
- [11] Noll M. Introduction to telephones and telephone systems. Artech House. Boston. 1991.
- [12] Bigelow S. J. Understanding telephone electronics. SAMS. Indiana. 1991.
- [13] Smith S. The Scientist and engineer's guide to digital signal processing. 1997.
- [14] Kuo, Sen-Maw. Real-time digital signal processing :implementations, applications, and experiments with the TMS320C55X. Illinois, USA . 2001
- [15] Gutiérrez E. A. y González C. D. Realización de una tarjeta multisensada de voz sobre una plataforma DSP. UNAM. FI. DF. 2008.
- [16] Texas Instruments. TMS320VC5402 Fixed point digital signal processor. Literature number SPRS079E.

- [17] Texas Instruments. TMS320C54x Reference set, volume 1: CPU and peripherals. Literature number SPRU131G.
- [18] Escobar S. L. y Álvarez L. Sistema de alarma vía telefónica utilizando voz sintética. XI Congreso Internacional sobre Innovación y Desarrollo Tecnológico CIINDET, 2014, Cuernavaca, Morelos, México, Abril 2014.

Anexo A

Código de Matlab® para el análisis
fuera de línea

```

%ESTE PROGRAMA ANALIZA UNA SEÑAL DE VOZ, OBTIENE SUS PARÁMETROS, LOS
%CONVIERTE EN PUNTO ENTERO

clear;
p=input('INGRESE EL ORDEN DE PREDICCIÓN DEL SISTEMA: '); %orden de predicción
N=input('INGRESE EL NÚMERO DE MUESTRAS A TOMAR DE LA SEÑAL: '); %orden de predicción

ai(1:p+1)=0;
n=1:N;

n1=ceil(N/3); %primer subsegmento
n2=ceil(2*N/3); %segundo subsegmento
n3=N; %tercer subsegmento

Wsr(1:N)=0; %ventana recortada para analizar el pitch
w1=0.54-0.46*cos((2*pi*n)/(N-1)); %ventana Hamming

s=wavread('/home/luis/Documentos/Tesis/Audio/Yo/ventana'); %lectura de señal de audio
silencio=wavread('/home/luis/Documentos/Tesis/Audio/silencio.wav'); %silencio experimental

s=s';

L=length(s); %Numero de datos de la señal a analizar

M=floor(L/N); %Numero de ventanas

ruido=randn(1,N); %Señal de ruido para segmentos no voceados
ruido=ruido/(max(abs(ruido)));
pre_de=0.9375;
Qi=2 ^ 12;
ajuste=1/Qi;%convierte el último valor en flotante
ruido=floor(ruido*Qi);
alpha=floor(pre_de*Qi);

%-----
% OBTENCIÓN DE LA ENERGÍA DE UN UMBRAL DE SILENCIO EXPERIMENTAL
%-----
E=0;
for x=1:N
    E=E+(silencio(x)) ^ 2; %calcula la energia de un segmento de silencio
end
%-----

ventana=M;
for a=0:M-1

    Ws1=w1.*s(a*N+1:(a+1)*N); %recorte de la señal de voz al número de muestras deseado

%-----
% DETERMINA SI EL SEGMENTO ES SILENCIO
%-----
Es=0;
for x=1:N

```

```

    Es=Es+(Ws1(x))^2;          %calcula la energia de un segmento de voz
end

if(Es>0.1*E)
%-----
%-----
% FILTRO DE PREÉNFASIS
%-----
Ws(1)=Ws1(1);
for n=2:N
    Ws(n)=Ws1(n)-pre_de*Ws1(n-1);    %filtro de deénfasis alpha=0.9
end;
%-----

%-----
% ESTA SECCIÓN REALIZA EL ANALISIS DE LA SEÑAL DE VOZ
%-----
    rss=xcorr(Ws);          %autocorrelación de la señal Ws
    Rp=toeplitz(rss(N:N+p-2));    %genera la matriz toeplitz con 'p' desde R(0)...R(p-1)
    rp=rss(N+1:N+p-1);        %genera el vector de autocorrelación desde R(1)..R(p)

    aj=(-inv(Rp)*rp)';        %coeficientes LPC (ai)
    ai=[1 aj];
    G=sqrt( rss(N)-sum(ai.*rss(N+1:N+p)) );    %ganancia
    G=floor(G*Qi);          %ganancia Q12
    ai=floor(ai*Qi);        %coeficientes a1 en Q12
%-----

%-----
% ESTA SECCIÓN DETERMINA SI EL SEGMENTO ANALIZADO ES VOCEADO O NO VOCEADO Y
% ESTIMA EL PERIODO DE PITCH
%-----
    A1=max(Ws(1:n1));        %máximo del primer segmento
    A3=max(Ws(n2:n3));        %máximo del tercer segmento

    CL=0.6*min(A1,A3);        %valor del umbral

    for i=1:N
        if(Ws(i)>=CL)
            Wsr(i)=1;        %si Ws es mayor que el valor umbral positivo su valor es 1
        else
            if(Ws(i)<=-CL)
                Wsr(i)=-1;    %si Ws es menor que el umbral negativo su valor es -1
            else
                Wsr(i)=0;    %cero en otro caso
            end
        end
    end
end
end

```

```

Rc=xcorr(Wsr); %autocorrelación de la señal recortada
Rc(1:N)=Rc(N:2*N-1); %recorre la correlación al origen
Rc=Rc/Rc(1); %normaliza la señal

[m pit]=max(Rc(20:200)); %busca el máximo de la señal de autocorrelación entre las muestras 20 y 200

if(m>0.2*Rc(1))
    voiced=1; %se define como un segmento voceado y se estima el pitch
    pitch=pit+20;
else
    voiced=0; %se define como un segmento no vocead, no hay pitch
    pitch=0;
end

%-----
ventana=[ventana pitch G ai];

else

    ventana=[ventana -1];

end
end

longitud=dec2hex(length(ventana)+1);
%-----

%-----
% GUARDA EN UN ARCHIVO LOS COEFICIENTES EN UN ARCHIV
%-----
fid = fopen('/home/luis/Documentos/Tesis/Audio/Yo/Datos/ventana.dat','w'); %ABRIMOS UN ARCHIVO

fprintf(fid,'1651 2 8000 1 %s \n',longitud); %escribe la longitud

fprintf(fid,'%d \n',ventana);

fclose(fid);
%-----

%-----
% GUARDA EN UN ARCHIVO QUE GUARDA UNA SEÑAL DE RUIDO QUE SE PUEDE LEER EN C
%-----
fid = fopen('/home/luis/Documentos/Tesis/Audio/Yo/Datos/ruido.dat','w');

fprintf(fid,'1651 2 8000 1 %d \n',dec2hex(N)); %escribe la longitud

fprintf(fid,'%d \n',ruido);
fclose(fid);
%-----

```

Anexo B

Código en C para el DSP

```

#include "board.h"
#include "type.h"
#include "codec.h"
#include "daa.h"

#define GLOBAL_INT_ENABLE    asm( " rsbx intm ")
#define GLOBAL_INT_DISABLE  asm( " ssbx intm ")

#define N 128
#define p 10

volatile short *imr = (short *) 0x00;
volatile short *ifr = (short *) 0x01;
volatile short *pmst= (short *) 0x01d;

volatile short *drr = (short *) 0x41; //recibe del micrófono
volatile short *dxr = (short *) 0x43; //envía a la bocina

volatile short *daarr = (short *) 0x21; //recibe de la línea
volatile short *daaxr = (short *) 0x23; //envía a la línea

volatile short *prd = (short *) 0x25;
volatile short *tcr = (short *) 0x26;

HANDLE hDevice;

short yn;
int banderita=0;
int ruido[N];

ioport unsigned port8002;

interrupt void timer(void){                                //rutina de interrupción del timer

    banderita=1;
} //fin timer

void config(void);
void gen_dtmf(int a1, int b1, int a2, int b2);    //generacion de la señal dtm
void retardo(int t);
void genera_voz(int cosa[]);
void interfon();

void main(void){

int puerta[307],principal[434], abierta[432];
int del[380], patio[263],ventana[594], de[225] la[276], recamara[441];
int cocina[417], uno[354], dos[349], tres[265];

long n;
int sel0,sel1,sel2,sel3,sel4,sel5,sel6,sel7;
short data;

```

```
config();
```

```
inicio:
```

```
data = port8002;
sel0=data&0x001; //D0
sel1=data&0x002; //D1
sel2=data&0x004; //D2
sel3=data&0x008; //D3
sel4=data&0x010; //D4
sel5=data&0x020; //D5
sel6=data&0x040; //D6
sel7=data&0x080; //D7

if(data&0xFF!=0){           //algún boton se presiono
    goto descuelga;
}
else
    goto inicio;
}
```

```
descuelga:
```

```
daa_offhook();

gen_dtmf(11036,24218,14206,16324);//#0
retardo(50);

gen_dtmf(9315,26956,13322,19072);//#4
retardo(50);

gen_dtmf(9315,26956,13322,19072);//#4
retardo(50);

gen_dtmf(9315,26956,14206,16324);//#5
retardo(50);

gen_dtmf(9315,26956,14206,16324);//#5
retardo(50);

gen_dtmf(8527,27979,13322,19072);//#1
retardo(50);

gen_dtmf(8527,27979,14206,16324);//#2
retardo(50);

gen_dtmf(9315,26956,15021,13084);//#6
retardo(50);

gen_dtmf(10163,25701,14206,16324);//#8
retardo(50);
```

```

gen_dtmf(9315,26956,15021,13084);##6
retardo(50);

gen_dtmf(9315,26956,13322,19072);##4
retardo(50);

gen_dtmf(9315,26956,15021,13084);##6
retardo(50);

gen_dtmf(8527,27979,13322,19072);##1
retardo(4000); //retardo de 4 s

if(sel0!=0){          //interfon
    interfon();
}
else if(sel1!=0){    //puerta principal abierta
    for(n=30;n>0;n--){
        genera_voz(puerta);
        genera_voz(principal);
        genera_voz(abierta);
        retardo(1000);
    }
}
else if(sel2!=0){    //puerta del patio abierta
    for(n=30;n>0;n--){
        genera_voz(puerta);
        genera_voz(del);
        genera_voz(patio);
        genera_voz(abierta);
        retardo(1000);
    }
}
else if(sel3!=0){    //ventana de la cocina abierta
    for(n=30;n>0;n--){
        genera_voz(ventana);
        genera_voz(de);
        genera_voz(la);
        genera_voz(cocina);
        genera_voz(abierta);
        retardo(1000);
    }
}
else if(sel4!=0){    //ventana de la recamara uno abierta
    for(n=30;n>0;n--){
        genera_voz(ventana);
        genera_voz(de);
        genera_voz(la);
        genera_voz(recamara);
        genera_voz(uno);
        genera_voz(abierta);
        retardo(1000);
    }
}
}

```

```

else if(sel5!=0){          //ventana de la recamara dos abierta
    for(n=30;n>0;n--){
        genera_voz(ventana);
        genera_voz(de);
        genera_voz(la);
        genera_voz(recamara);
        genera_voz(dos);
        genera_voz(abierta);
        retardo(1000);
    }
}
else if(sel6!=0){          //ventana de la recamara tres abierta
    for(n=30;n>0;n--){
        genera_voz(ventana);
        genera_voz(de);
        genera_voz(la);
        genera_voz(recamara);
        genera_voz(tres);
        genera_voz(abierta);
        retardo(1000);
    }
}
else if(sel7!=0){          //ventana del comedor abierta
    for(n=30;n>0;n--){
        genera_voz(ventana);
        genera_voz(del);
        genera_voz(comedor);
        genera_voz(abierta);
        retardo(1000);
    }
}
else{                      //nada
    goto colgado;
}

```

colgado:

```

    daa_onhook();

```

```

    goto inicio;
    while(1);

```

```

} //fin main

```

```

//-----
// ESTA FUNCION CONFIGURA LOS PERIFERICOS
//-----
void config(void){

```

```

GLOBAL_INT_DISABLE;          //deshabilita interrupciones

```

```

    brd_init(10);

```

```

hDevice = codec_open(DAA_CODEC);
codec_sample_rate(hDevice,SR_8000);

hDevice = codec_open(HANDSET_CODEC);

codec_dac_mode(hDevice, CODEC_DAC_15BIT);
codec_aout_gain(hDevice, CODEC_AOUT_MINUS_0dB);
codec_sample_rate(hDevice,SR_8000);

*pmst = 0x00A0 ;           // remapea vec. interrup.

*imr |= 0x0008 ;

*tcr = 0x000E ;           // control del timer (ultimos 4 bits para TDDR)
*prd = 0x0052;           // periodo del timer para generar una frecuencia de 8000Hz

*ifr = *ifr;

daa_init();

GLOBAL_INT_ENABLE;

    daa_onhook(); //cuelga la linea en caso de que estas se encontrara abierta
} //fin config

//-----
// ESTA FUNCION GENERA UNA SEÑAL DTMF
//-----
void gen_dtmf(int a1, int b1, int a2, int b2){

short n,x1=0,x2=0,y1=0,y2=0,yn=0,xn=0;
long aux=0;

    banderita=0;

    x2=0;
    x1=a1;

    y2=0;
    y1=a2;

    *daaxr=0;
    *dxr=0;

    while(banderita!=1);
    *daaxr=y1+x1;

    for(n=2;n<500;n++){
        banderita=0;
        aux=b1*x1;
        xn=(int)(aux>>14)-x2;
        x2=x1;

```

```

        x1=xn;

        aux=b2*y1;
        yn=(int)(aux>>14)-y2;
        y2=y1;
        y1=yn;

        while(banderita!=1);
        *daaxr=yn+xn;
    }
} //fin gen_dtmf

```

```

//-----
// RETARDO [ms]
//-----
void retardo(int t){ //retardo de 1ms
int n,k;

port0=0;
for(n=0;n<t;n++){ //retardos
    for(k=0;k<8;k++){
        while (banderita!=1);
        banderita=0;
    }
}

} //fin retardo

```

```

//-----
// ESTA FUNCION GENERA VOZ SINTETICA
//-----
void genera_voz(int cosa[]){
int m=0,n,k=0,M;
int LI=1;
int y[p];
int G,pit,alpha=3276;
int ai[p];
int x[N];

long aux;

M=cosa[0];
pit=cosa[1];
banderita=0;
while(m<M){
n=0;

for(n=0;n<N;n++)
    x[n]=0;
for(n=0;n<p-1;n++){ //borra la señal y para la nueva ventana
    y[n]=0;
}
}

```

```

if(pit<0){
  for(n=0;n<N;n++){
    while (banderita!=1);
    banderita=0;
    *daaxr=0;
  }
  LI=LI+1;
}

else{
  G=cosa[LI+1];
  ai[0]=cosa[LI+2];
  ai[1]=cosa[LI+3];
  ai[2]=cosa[LI+4];
  ai[3]=cosa[LI+5];
  ai[4]=cosa[LI+6];
  ai[5]=cosa[LI+7];
  ai[6]=cosa[LI+8];
  ai[7]=cosa[LI+9];
  ai[8]=cosa[LI+10];
  ai[9]=cosa[LI+11];

  if(pit==0){ //generador de pitch o ruido
    for(n=0;n<N;n++){
      x[n]=ruido[n];
    }
  }
  else{
    for(n=0;n<N;n+=pit)
      x[n]=4096;//1(Q12)
  }

  //filtro todo polo
  for(n=0;n<N;n++){
    yn=0;

    for(k=1;k<p;k++){
      aux=ai[k]*y[k-1];
      yn=yn+(int)(aux>>12);//obtiene sum(ai[k]*y[n-k])
    }
    aux=G*x[n];
    yn=(int)(aux>>12)-yn;//obtiene G*x[n]-sum(ai[k]*y[n-k])

    for(k=p-1;k>0;k--)
      y[k]=y[k-1];//recorre los valores del vector y

    y[0]=yn;//yn->y[n-1]

    aux=alpha*y[1];
    yn=y[0]+(int)(aux>>12);//filtro de deenfasis alpha=0.9
  }
}

```

```

    while (banderita!=1);
    banderita=0;
        *daaxr=yn;

    }//fin for

    LI=LI+12;
    }//end else
    pit=cosa[LI];
    m+=1;//incrementa la ventana

}//fin while

}//fin genera_voz

```

```

//-----
//          ESTA FUNCION SIRVE COMO INTERFON
//-----
void interfon(){

    long n;

    for(n=720000;n>0;n--){
        while (banderita!=1);

        *daaxr=*drr;
        *dxr=*daarr;
        banderita=0;
    }//fin for
}//fin interfon

```

