



Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Electrónica

Curso: Procesamiento Digital de Señales

Proyecto 1

Realizado por:

Gladys Arias Abarca	2013018896
Alexis Gavriel Gómez	2016085662
Cristian Rivera López	2015073683
Andres Vargas Rivera	2015097412

Profesor:

Ing. Miguel Barboza, M.Sc

I Semestre, 2019

Fecha de Entrega: 18 de Junio

Investigación	3
Análisis del problema	4
Propuestas de Solución	4
Generación de Tonos	4
Detección de Tonos	5
Interfaz de Usuario Final	6
Metodología de diseño	7
Modelo de Cascada	7
Requisitos de Software	8
Diagramas de Diseño	10
Descripción de métodos	12
Generador de Tonos	12
Detector de Tonos	12
Interfaz de Usuario	12
Bibliotecas	13
Herramientas de ingeniería	14
Matlab	14
Micromodeler DSP	14
Ubuntu 18.04.02	14
Qjack	14
QT	14
Octave-Forge	14
Resultados iniciales	15
Generación de Tonos:	15
Detector de Tonos	15
Conclusiones	17
Referencias	17

1. Investigación

El auge de los sistemas embebidos a la medida, diseñados para espacios inteligentes, así como sistemas de propósito específico para un mejor rendimiento genera que el aprendizaje en áreas como el procesamiento digital de señales sea importante en la formación de los profesionales emergentes en el área de la tecnología. El procesamiento de señales digitales (DSP) es el proceso de analizar y modificar una señal para optimizar o mejorar su eficiencia o rendimiento. Implica aplicar varios algoritmos matemáticos y computacionales a señales analógicas y digitales para producir una señal de mayor calidad que la señal original.

El uso de sistemas digitales es de alta demanda laboral y práctica en constante en la investigación y en el desarrollo de nuevas tecnologías siendo el internet de las cosas primordial las herramientas digitales y analógicas que son necesarias para poder realizar y establecer las conexiones. Dentro de sus aplicaciones se encuentran el procesamiento de imágenes digitales, el control electrónico y las telecomunicaciones entre otros.

En el proyecto en cuestión se diseña e implementa un sistema de filtros digitales IIR para el análisis de señales específicas de entrada para de esta forma realizar una detección precisa de información. Esta información corresponde a dígitos basados en la tabla normalizada de frecuencias para Dual Tone Multi Frequency DTMF. El mapeo correspondiente identifica que se esta realizando una llamada al número del sistema. El filtro digital es un sistema que realiza operaciones matemáticas en una señal de tiempo discreto muestreada para reducir o mejorar ciertos aspectos de esa señal. Esto contrasta con el otro tipo importante de filtro electrónico, el filtro analógico, que es un circuito electrónico que funciona con señales analógicas de tiempo continuo

2. Análisis del problema

El problema planteado consiste en la generación y detección de tonos con frecuencias específicas con el fin de poder realizar llamadas así como tener la capacidad de identificar una llamada entrante y aceptarla. Con respecto al problema el equipo de trabajo mediante la técnica de “divide y

vencerás” procede dividir el problema en 3 secciones específicas las cuales son la generación de los tonos, la detección de los tonos y la interfaz que comunica al sistema con el usuario final. Cada una de estas secciones puede ser solucionada de manera independiente sin tener conflicto con las demás lo que le proporciona al equipo una gran capacidad de paralelización de las labores para una mejor división de tareas. Para el manejo del código fuente de la solución el equipo plantea primordialmente el uso de Github, herramienta de suma utilidad para la integración continua de versiones funcionales.

3. Propuestas de Solución

Generación de Tonos

- Generación de senoidales:

Osciladores:

El oscilador digital se basa en el caso extremo de un resonador con polos complejos en el círculo unitario, cuando el sistema se alimenta con $\delta(n)$ oscila en base a la siguiente ecuación.

$$y(n) = -a_1 y(n-1) - y(n-2) + b_0 \delta(n) \quad (1)$$

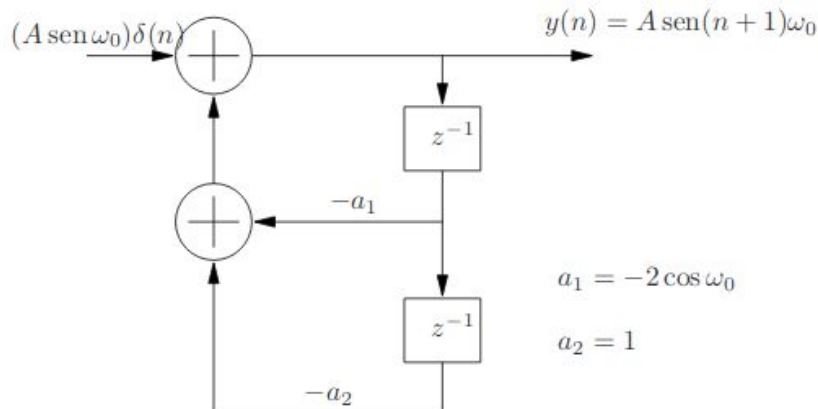


Figura 1. Generador digital sinusoides

Generación directa

La generación de sinusoidales de forma directa se basa en calcular el seno o coseno de cada valor dentro de un vector definido por pasos de tiempo. La ecuación siguiente define el comportamiento de generación.

$$\left(\sin\left(\frac{2\pi n \text{ frequency}}{hz}\right) + \sin\left(\frac{2\pi n \text{ freq}2}{hz}\right) \right) \quad (2)$$

donde n es el valor en el tiempo

- Suma de señales: Para la generación de tonos se realiza la suma de dos señales sinusoidales a frecuencias específicas según la tabla de frecuencias normalizadas DTMF. Dicha sumatoria de señales compone, de forma conjunta, el tono final de cada uno de los dígitos establecidos.

Tabla 1. Frecuencias DTMF

	1209 Hz	1336 Hz	1447 Hz	1663 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

Detección de Tonos

- Filtros por frecuencia: La primera etapa para la detección e identificación del tono, es la implementación de filtros que sesgen ciertas frecuencias. Para el caso de este sistema, cada tono está formado por dos frecuencias diferentes, donde con 8 frecuencias distintas forman 16 posibles tonos, por lo que se requieren 8 distintos filtros para poder identificar las frecuencias de columnas y de filas para cada tono dual.

Para este caso se utilizarán filtros Chebyshev, Tipo II (polos y ceros) de orden 6. El filtro se compone de 3 filtros de orden 2 en cascada. Entre sus características principales está el rizado constante en la banda de rechazo y característica monótona en la banda de paso. Se eligió este tipo de filtro ya que tiene una banda de corte plana, ancho de banda fácilmente ajustable, y cálculos computacionales rápidos, su modelo matemático es el siguiente.

$$|H(\Omega)|^2 = \frac{1}{1 + \epsilon^2 \left[\frac{T_N^2(\frac{\Omega_*}{\Omega_P})}{T_N^2(\frac{\Omega_S}{\Omega})} \right]} \quad (3)$$

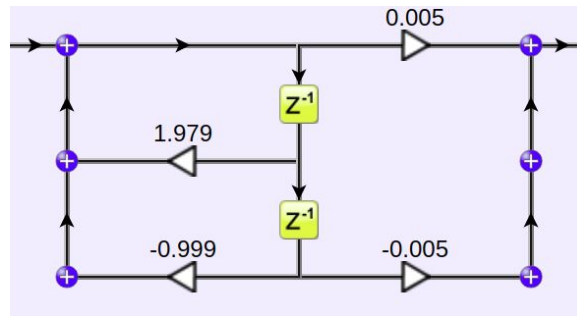


Figura 2. Diagrama de bloques de una etapa del filtro

La figura anterior representa el diagrama de bloques de una de las etapas que se colocan en cascada en los filtros.

- **Análisis de Potencia:** Para identificar si una señal existe se analiza la energía en la salida de cada uno de los filtros. Al analizar la potencia de una señal se puede identificar si el valor calculado supera un umbral predefinido. Para calcular la potencia se utiliza la siguiente ecuación:

$$\frac{1}{N} \sum_{n=0}^N \|x(n)\|^2 \quad (4)$$

Interfaz de Usuario Final

- **QT:** La herramienta QT permite desarrollar aplicaciones en el lenguaje de C++. Provee herramientas que facilitan la creación e interacción con una interfaz gráfica y herramientas para la depuración del programa.

4. Metodología de diseño

Modelo de Cascada

El modelo cascada consiste en un proceso secuencial en el cual existe un marcado punto en el cual se pasa de una etapa a otra en un proyecto de desarrollo de software involucrando posibles acciones específicas para cada etapa. Las etapas más comunes de un modelo de cascada son:

- **Adquisición de requerimientos:** En esta fase se analizan las necesidades de los usuarios finales del software para determinar qué objetivos debe cubrir. De esta fase surge una memoria llamada SRD (documento de especificación de requisitos), que contiene la especificación completa de lo qué debe hacer el sistema sin entrar en detalles internos.
- **Diseño del sistema:** Es la fase en donde se realizan los algoritmos necesarios para el cumplimiento de los requerimientos del usuario así como el diseño de las posible arquitecturas para la solución del problema como también los análisis necesarios para saber que herramientas usar en la etapa de Implementación.
- **Implementación del sistema:** Durante la implementación, las estructuras y los flujos de trabajo se implementan teniendo en cuenta las condiciones marco y los objetivos sistémicos. El diseño de software se convierte en un programa directamente relacionado con un sistema operativo, uno o más lenguajes de programación y la infraestructura. El resultado suele ser un software operativo, a menudo en versión beta.
- **Testing:** La fase de implementación es seguida por la prueba de todos los componentes de software, módulos y todo el sistema. También se comprueba la integración en sistemas operativos específicos.
- **Producción:** Es la fase en donde el usuario final ejecuta el sistema, para ello el o los programadores ya realizaron exhaustivas pruebas para comprobar que el sistema no falle.

5. Requisitos de Software

A continuación se enumeran los requerimientos del proyecto, los cuales fueron extraídos tanto del enunciado del proyecto como de indicaciones por parte del profesor en clase.

Tabla 2. Requisitos del Proyecto

1	La codificación de cada dígito por medio de la señal compuesta por la suma de dos tonos según la tabla 1, deben tener una duración de 40 ms.
2	Entre cada dígito en una secuencia de marcación deben existir 40 ms. de separación, en los cuales se interpretan como una pasa o un nulo en la señal
3	Las amplitudes de los dos tonos que conforman un dígito, el del grupo de alta y el del grupo de bajas y alta frecuencia (Filas y columnas) deberán ser iguales.
4	Un número telefónico se conformará por una secuencia de 8 dígitos
5	Las tramas enviadas al llamar de un teléfono a otro estarán conformados por la siguiente secuencia: la tecla numeral seguida de 8 dígitos de del 0 al 9 y concluirán con el *
6	Todos los teléfonos estarán en contacto entre sí de forma inalámbrica, utilizando el aire como canal de comunicación (funcionan por medio de sonido: parlantes y micrófonos del computador)
7	La identificación deberá basarse por el métodos de filtrado en frecuencia bajo la implementación de filtros a partir de sus ecuaciones de diferencias(no se puede usar DFT)
8	El sistema deberá de contar con un teclado matricial
9	Una Tecla (Elemento gráfico) de función interruptor que permite emular la función de colgar o descolgar el teléfono. Además , una llamada solo se puede aceptar una vez que se detecta sólo si se descuelga el teléfono, si el Teléfono está descolgado no puede recibir llamadas.
10	Contar con una biblioteca dinámica que permita el control y adquisición de las señales.
11	Una sección visible donde sea posible observar las últimas 10 llamadas, tanto salientes como entrantes. Debe observar la hora en que se realizó y el número telefónico asociado indicando si fue saliente o entrante.

Tabla 3. Chequeo de los requisitos del proyecto.

	Requisito	Implementado
1	La codificación de cada dígito por medio de la señal compuesta por la suma de dos tonos según la tabla 1, deben tener una duración de 40 ms.	si
2	Entre cada dígito en una secuencia de marcación deben existir 40 ms. de separación, en los cuales se interpretan como una pasa o un nulo en la señal	si
3	Las amplitudes de los dos tonos que conforman un dígito, el del grupo de alta y el del grupo de bajas y alta frecuencia (Filas y columnas) deberán ser iguales.	si
4	Un número telefónico se conformará por una secuencia de 8 dígitos	si
5	Las tramas enviadas al llamar de un teléfono a otro estarán conformados por la siguiente secuencia: la tecla numeral seguida de 8 dígitos de del 0 al 9 y concluirán con el *	si
6	Todos los teléfonos estarán en contacto estarán en contacto entre sí de forma inalámbrica, utilizando el aire como canal de comunicación (funcionan por medio de sonido: parlantes y micrófonos del computador)	si
7	La identificación deberá basarse por el métodos de filtrado en frecuencia bajo la implementación de filtros a partir de sus ecuaciones de diferencias(no se puede usar DFT)	si
8	El sistema deberá de contar con un teclado matricial	si
9	Una Tecla (Elemento gráfico) de función interruptor que permite emular la función de colgar o descolgar el teléfono. Además , una llamada solo se puede aceptar una vez que se detecta sólo si se descuelga el teléfono, si el Teléfono está descolgado no puede recibir llamadas.	si
10	Contar con una biblioteca dinámica que permita el control y adquisición de las señales.	si

6. Diagramas de Diseño

A continuación se presentan una serie de diagramas que ayudan al lector a comprender la organización e interconexión de cada uno de los componentes del proyecto. Así como una breve explicación de cada uno de los diagramas.

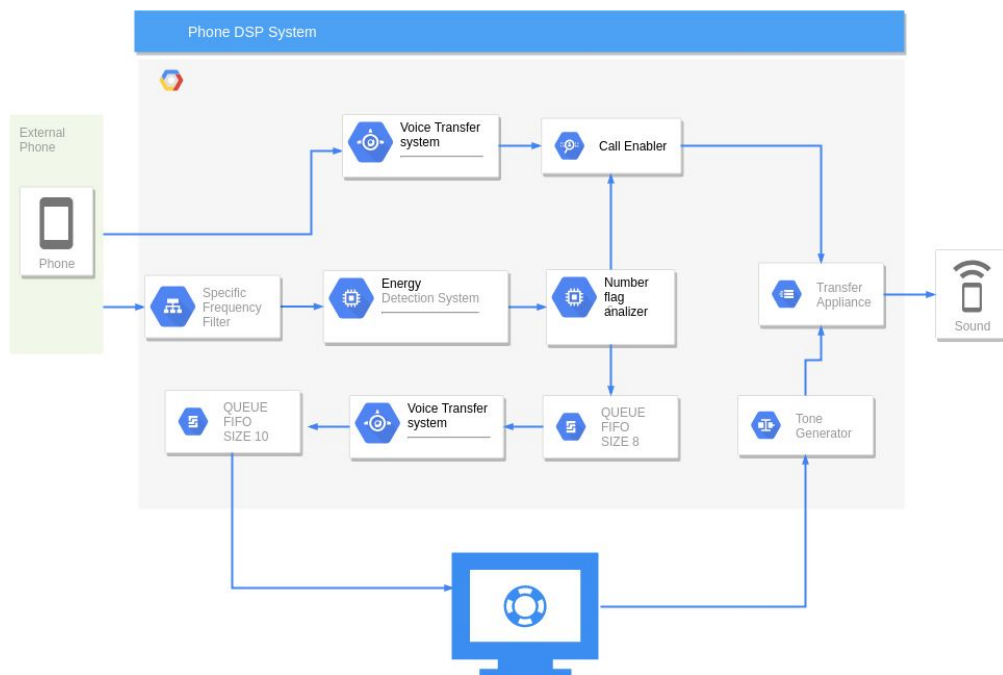


Figura 3. Arquitectura de la solución

El diagrama de arquitectura de la solución proporciona una perspectiva de los componentes principales del proyecto a grandes rasgos haciendo principal énfasis en las interacciones intermedias que deben ocurrir en el sistema para que se lleve a cabo cada una de las tareas específicas. Para este diseño se contemplan conexiones externas las cuales son necesarias para la interacción del usuario final con el sistema implementado.

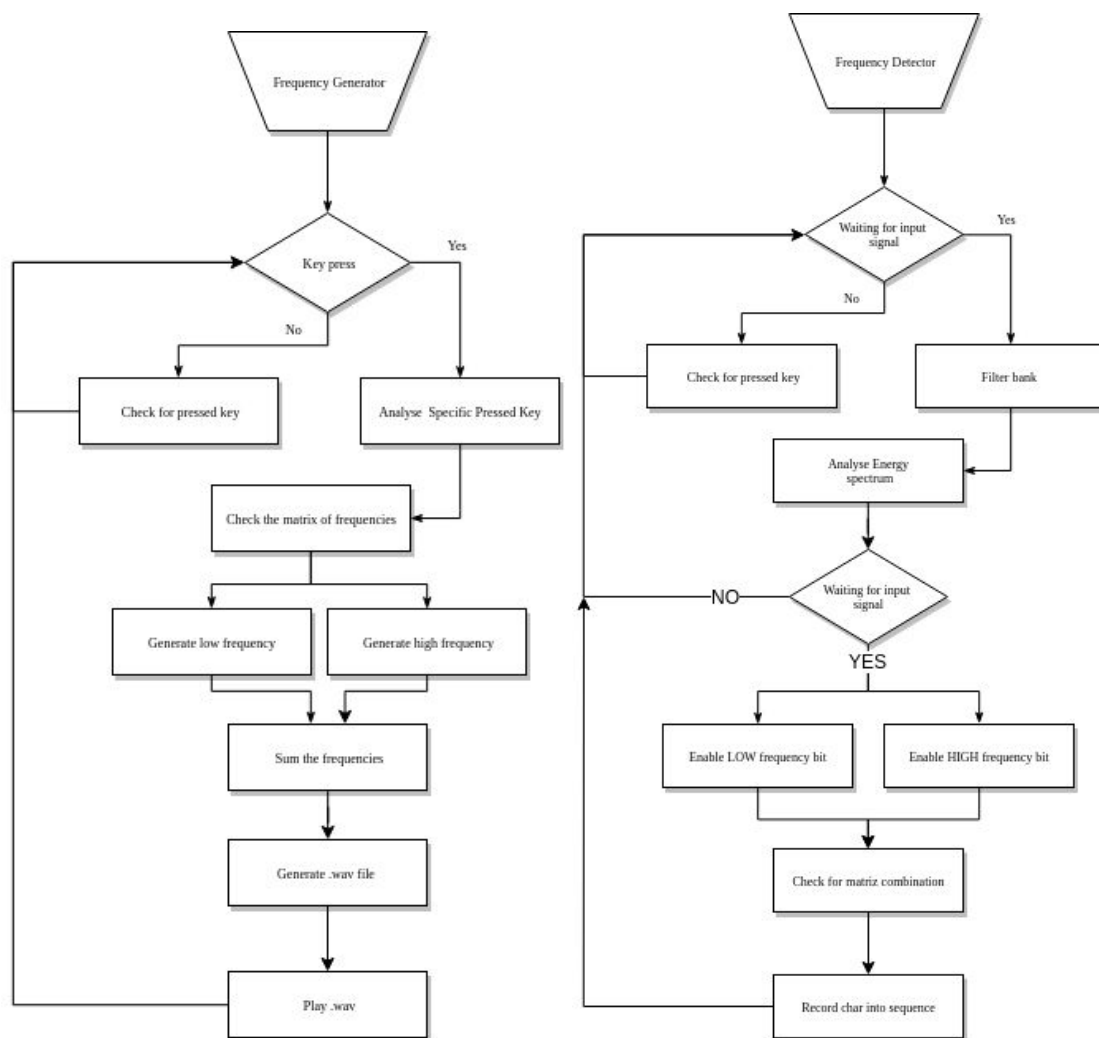


Figura 4. Diagramas de Flujo de Detección y Generación

Para el entendimiento del flujo que lleva a cabo el sistema para interpretar la información se presentan los diagramas de dos procesos que se dan de manera simultánea que son la detección de los posibles tonos así como la generación de tonos a los números que se desea llamar. Estos procesos son una vez completan su flujo (happy path) de forma automática vuelven al inicio del flujo ya que al ser análisis de datos en vivo el sistema no puede detenerse. Como subsistemas cada uno de los flujos para una ejecución de instrucciones completas debe ser finalizado en más de una ocasión con la finalidad de poder detectar o marcar un número completo compuesto de 8 dígitos.

7. Descripción de métodos

Generador de Tonos

Para la generación de tonos se utilizó el método directo. La decisión de utilizar este método se basó en la facilidad de implementación. Al presionar un número, se generan las señales sinusoidales correspondientes para luego conformar la señal o tono correspondiente al número seleccionado. Tras definir la secuencia de tonos que componen el número a marcar se genera un archivo “.wav” que contiene el sonido que representa dicha cadena de números.

Detector de Tonos

Para la detección de los tonos se procede a filtrar la señal de entrada por medio de 8 filtros en paralelo. Estos filtros determinan si la señal de entrada se encuentra dentro de un rango determinado de frecuencia. Consecuentemente se calcula la potencia de cada una de las señales de salida de los filtros y finalmente se comparan todas las señales entre sí y además con un umbral definido. Si la señal con mayor potencia supera el umbral, se concluye que la señal se ha detectado.

Interfaz de Usuario

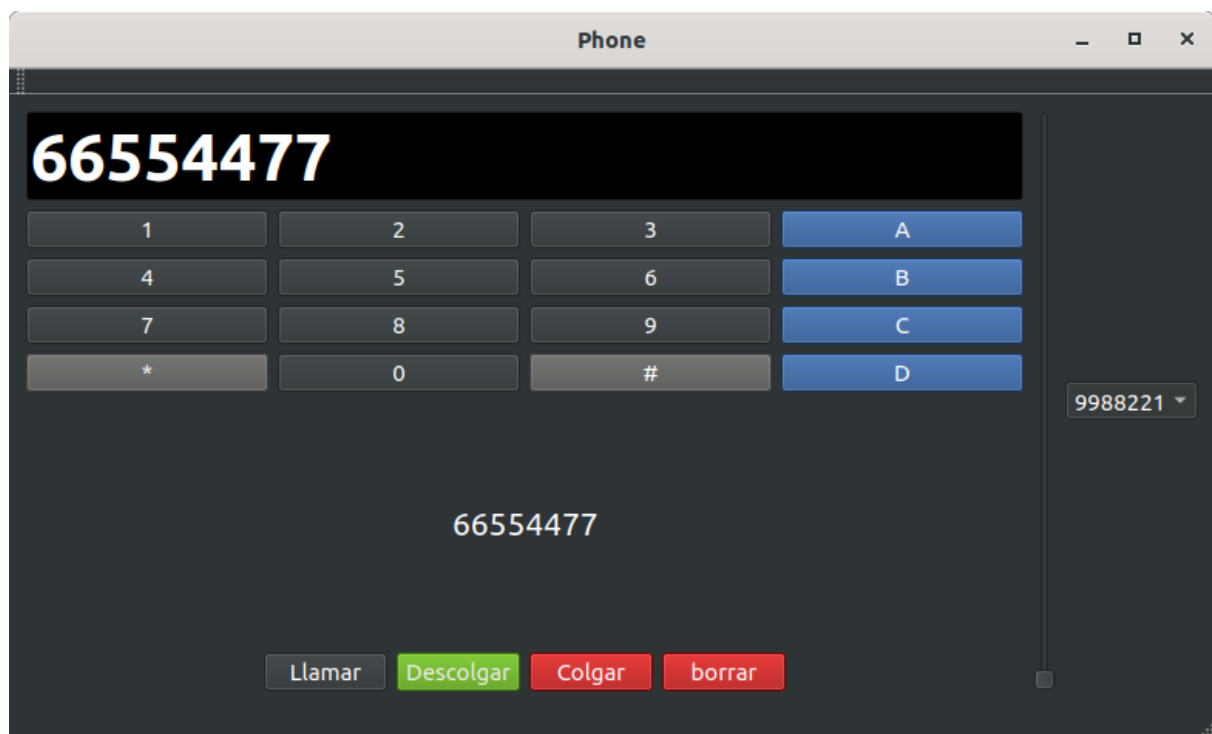


Figura 5. Pantalla de Interfaz de Usuario.

La interfaz permite interactuar con ella a partir de la matriz conformada por los botones respectivos. Presenta botones que permiten al usuario colgar o descolgar el teléfono, así como un botón que permite al usuario borrar un dígito del número que se ha escrito.

8. Bibliotecas

Biblioteca	Funcionalidad
QMutex	Es proteger un objeto, estructura de datos o sección de código para que solo un hilo pueda acceder a él a la vez
QThread	The QThread class provides a platform-independent way to manage threads.
jack	is a low-latency audio server, written for any operating system that is reasonably POSIX compliant. It currently exists for Linux, OS X, Solaris, FreeBSD and Windows. It can connect several client applications to an audio device, and allow them to share audio with each other. Clients can run as separate processes like normal applications, or within the JACK server as "plugins".
QMainWindow	Una ventana principal proporciona un marco para construir la interfaz de usuario de una aplicación. Qt tiene QMainWindow y sus clases relacionadas para la administración de ventanas principales. QMainWindow tiene su propio diseño al que puede agregar QToolBars, QDockWidgets, un QMenuBar y un QStatusBar.
QTimer	La clase QTimer proporciona una interfaz de programación de alto nivel para los temporizadores. Para usarlo, cree un QTimer, conecte su señal de timeout () a las ranuras apropiadas y llame a start (). A partir de ese momento, emitirá la señal de timeout () a intervalos constantes.

9. Herramientas de ingeniería

Matlab

Es un sistema de cómputo numérico que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows, Mac OS X y GNU/Linux .

Micromodeler DSP

Es una herramienta de cómputo Online y gratuita que permite realizar diversas tareas o algoritmos en el área del procesamiento digital de señales. Entre las tareas realizadas en la aplicación se encuentra el diseño de los filtros.

Ubuntu 18.04.02

Ubuntu es un sistema operativo de código abierto, basado en la arquitectura Linux Debian. Que tiene múltiples licencias de software libre o de código abierto.

Qjack

Qjack es una aplicación Qt simple que controla y configura varios parámetros del daemon Jack..

QT

Qt es un marco de desarrollo de aplicaciones multiplataforma para computadoras de escritorio, integradas y móviles. Las plataformas compatibles incluyen Linux, OS X, Windows, VxWorks, QNX, Android, iOS, BlackBerry, Sailfish OS y otros.

Octave-Forge

GNU Octave es un lenguaje interpretado de alto nivel, destinado principalmente a cálculos numéricos. Proporciona capacidades para la solución numérica de problemas lineales y no lineales, y para realizar otros experimentos numéricos. También proporciona amplias capacidades gráficas para la visualización y manipulación de datos.

10. Resultados iniciales

Generación de Tonos:

La generación de tonos con el método directo cumple su objetivo, un ejemplo de generación de tono para es el siguiente que corresponde al tono 1 con una frecuencia en bajo (697 Hz) y una en alto (1209 Hz)

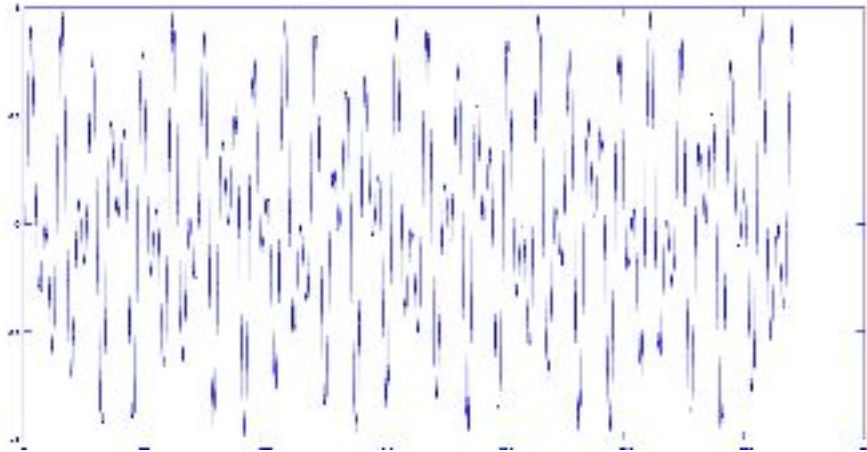


Figura 6. Tono 1

También la generación del tono se comprobó en audacity pero para este caso se hizo para los tonos * y 1 respectivamente.

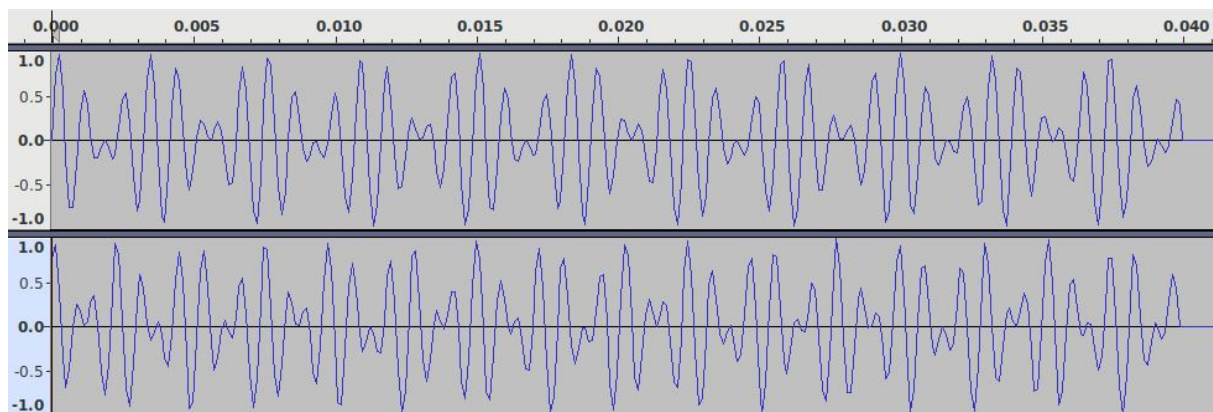


Figura 7. Tonos *,1

Detector de Tonos

- Filtros por frecuencia:

Los filtros Chebyshev se delimitaron usando Micromodeler DSP, y su funcionamiento se comprueba analizando la figura 8, donde la señal azul es la salida ideal del filtro chebyshev que filtra una señal con frecuencia central 697 Hz, mientras que la señal roja es la salida del filtro obtenida ante un tono dual que está conformado, un tono alto (1209 Hz) y otro bajo (697 Hz).

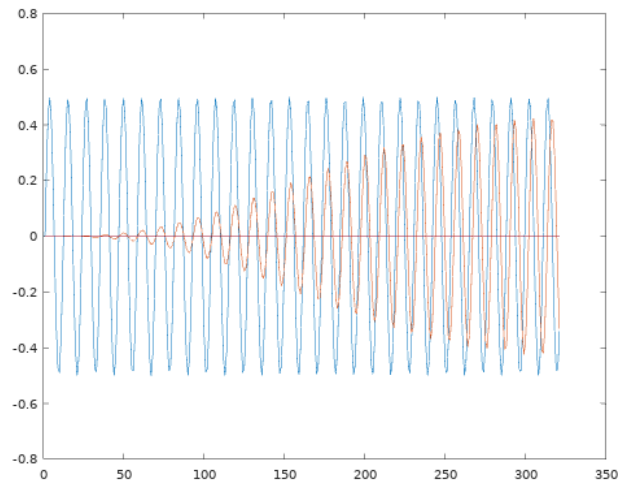


Figura 8. Señal Filtrada Teórica vs Señal obtenida

- **Análisis Potencia:**

Aplicando la ecuación (4) de potencia tanto a la señal ideal como a la salida obtenida del filtro chebyshev, si la señal obtenida por el filtro llega a cierto nivel de potencia comparado con un nivel definido de la potencia de la señal ideal, se puede saber con certeza la presencia de cierta frecuencia en el tono dual.

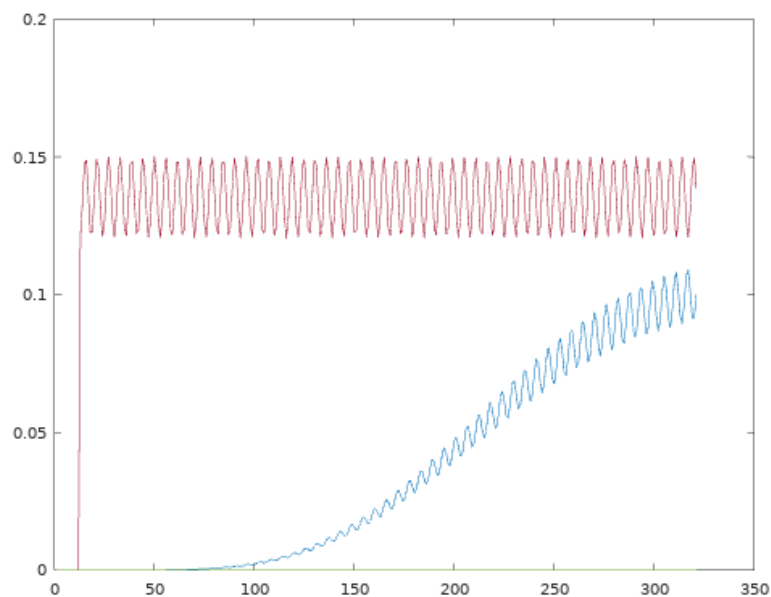


Figura 9. Potencia Teórica vs Potencia Obtenida


```
8988*654*#Tecla *
988*654*#*Tecla 8
88*654*#*8Tecla 9
8*654*#*89Tecla 8
*654*#*898Tecla 8
654*#*8988Tecla 0
54*#*89880Tecla 6
4*#*898806Tecla 5
*#*8988065Tecla 4
#*89880654Tecla #
*89880654#Estas recibiendo una llamada
```

Figura 10. Resultado de la detección del número

En la figura 10 se muestra la detección satisfactoria de la secuencia: *89880654#.

11. Conclusiones

EL filtro Chebyshev es el más indicado para este sistema pues si se compara con el filtro elíptico este último a mayor grado el filtro se vuelve más inestable.

La detección de tonos es altamente dependiente de la precisión numérica de los valores que conforman los filtros. Para que un filtro sea más preciso se puede incrementar su orden por medio de la colocación en cascada de otros filtros. Se debe considerar la facilidad de implementación de cada módulo de forma independiente, esto a pesar de las diferencias que cada forma de implementación puede involucrar.

12. Referencias

- [1]r. Capela, "QjackCtl - JACK Audio Connection Toolkit - Qt GUI Interface", Qjackctl.sourceforge.io, 2019. [Online]. Available: <https://qjackctl.sourceforge.io/>. [Accessed: 17- Jun- 2019].
- [2]"DTMF generator", Aggemam.dk, 2019. [Online]. Available: <https://aggemam.dk/code/dtmf>. [Accessed: 17- Jun- 2019].
- [3]"DTMF generator", Aggemam.dk, 2019. [Online]. Available: <https://aggemam.dk/code/dtmf>. [Accessed: 17- Jun- 2019].
- [4]"Micromodeler DSP - Digital Filter Design Software, Tutorials and Videos", Micromodeler.com, 2019. [Online]. Available: <http://www.micromodeler.com/>. [Accessed: 17- Jun- 2019].
- [5]l. Basics, "IIR Filter Basics - dspGuru", dspGuru, 2019. [Online]. Available: <https://dspguru.com/dsp/faqs/iir/basics/>. [Accessed: 17- Jun- 2019].
- [6]"DTMF DETECTION (VoIP)", What-when-how.com, 2019. [Online]. Available: <http://what-when-how.com/voip/dtmf-detection-voip/>. [Accessed: 17- Jun- 2019].