



UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA DE
TELECOMUNICACIONES
DEPARTAMENTO DE SEÑALES Y SISTEMAS



**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE DETECCIÓN
DE PATRONES DE AUDIO UTILIZANDO EL DISPOSITIVO
RASPBERRY PI MODELO B, PARA LA IDENTIFICACIÓN DE
CADENAS NACIONALES DE RADIO Y TV PARA SER INTEGRADO
EN LA CONMUTACIÓN AUTOMATIZADA DE LA
PROGRAMACIÓN DE LAS ESTACIONES FMUC Y UCTV**

LUZ MARÍA PENAGOS CARRILLO
WILDER DAVID JIMENEZ SILVA

Bárbula, 20 de Febrero del 2015



UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA DE
TELECOMUNICACIONES
DEPARTAMENTO DE SEÑALES Y SISTEMAS



**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE DETECCIÓN
DE PATRONES DE AUDIO UTILIZANDO EL DISPOSITIVO
RASPBERRY PI MODELO B, PARA LA IDENTIFICACIÓN DE
CADENAS NACIONALES DE RADIO Y TV PARA SER INTEGRADO
EN LA CONMUTACIÓN AUTOMATIZADA DE LA
PROGRAMACIÓN DE LAS ESTACIONES FMUC Y UCTV**

TRABAJO ESPECIAL DE GRADO PRESENTADO ANTE LA ILUSTRE UNIVERSIDAD DE
CARABOBO PARA OPTAR AL TÍTULO DE INGENIERO DE TELECOMUNICACIONES

LUZ MARÍA PENAGOS CARRILLO
WILDER DAVID JIMENEZ SILVA

Bárbula, 20 de Febrero del 2015



UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA DE
TELECOMUNICACIONES
DEPARTAMENTO DE SEÑALES Y SISTEMAS



CERTIFICADO DE APROBACIÓN

Los abajo firmantes miembros del jurado asignado para evaluar el trabajo especial de grado titulado «DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE DETECCIÓN DE PATRONES DE AUDIO UTILIZANDO EL DISPOSITIVO RASPBERRY PI MODELO B, PARA LA IDENTIFICACIÓN DE CADENAS NACIONALES DE RADIO Y TV PARA SER INTEGRADO EN LA CONMUTACIÓN AUTOMATIZADA DE LA PROGRAMACIÓN DE LAS ESTACIONES FMUC Y UCTV», realizado por los bachilleres LUZ MARÍA PENAGOS CARRILLO, cédula de identidad 17800974, WILDER DAVID JIMENEZ SILVA, cédula de identidad 20180922, hemos decidido otorgar la máxima calificación y la mención honorífica al presente trabajo, con base a los siguientes motivos:

1) El alcance del trabajo se extendió más allá de lo planteado. 2) Esmerada elaboración de la interfaz de usuario del dispositivo desarrollado. 3) Los autores demuestran originalidad e innovación en los algoritmos desarrollados para el procesamiento digital de señales. 4) Asertiva integración de software y hardware.

Firma

Prof. EDUARDO GONZÁLEZ

TUTOR

Firma

Prof. FABIÁN ROBLEDO

JURADO

Firma

Prof. AHMAD OSMAN

JURADO

Bárbula, 20 de Febrero del 2015

Dedicatoria

A Dios

A mi familia

A mis amigos

A mi casa de estudios UC

LUZ MARÍA PENAGOS CARRILLO

A Dios

A mis padres

A mis amigos

WILDER DAVID JIMENEZ SILVA

Agradecimientos

Primero a nuestros padres y hermanos que siempre han estado con nosotros apoyandonos en todo momento, a nuestros compañeros de carrera, a nuestros amigos que de alguna u otra forma han colaborado en la culminación de este proyecto de grado. A nuestro tutor Eduardo González por guiarnos en el desarrollo de este trabajo. A todo el equipo de DIMETEL por facilitarnos todos los recursos necesarios, en especial al Ing. Angelo Gasparini por su apoyo y dedicación en este proyecto.

Índice general

Índice de Figuras	XI
Índice de Tablas	XIII
Acrónimos	XV
Resumen	XVII
I. El Problema	1
1.1. Motivación	1
1.2. Objetivos	3
1.2.1. Objetivo General	3
1.2.2. Objetivos Específicos	3
1.3. Alcance	3
1.4. Recursos Utilizados	4
II. Marco conceptual	5
2.1. Antecedentes	5
2.2. Marco Teórico	6
2.2.1. Leyes y Sanciones	6
2.2.2. DIMETEL	8
2.2.2.1. FMUC Emisora Universitaria 104,5 FM	8
2.2.3. Comparación de Patrones	9
2.2.3.1. Técnica de Comparación de Patrones	9
2.2.3.2. Técnica de Reconocimiento por modelos ocultos de Markov	9
2.2.3.3. Transformada de Fourier	10
2.2.3.4. Correlación	10
2.2.3.5. Autocorrelación	11
2.2.3.6. Correlación Cruzada	12
2.2.4. Espectrograma	12
2.2.5. Análisis de audio utilizando algoritmos de procesamiento de imágenes	12

2.2.6.	Raspberry Pi Modelo B	13
2.2.7.	Tarjeta de audio	16
2.2.8.	Entorno de trabajo: Raspbian	16
2.2.9.	Python	17
III. Procedimientos de la investigación		19
3.1.	Procedimiento Metodológico.	19
3.1.1.	Fase I.	19
3.1.2.	Fase II.	20
3.1.2.1.	Selección del algoritmo de comparación de señales de audio.	20
3.1.2.2.	Descripción del algoritmo seleccionado.	20
3.1.2.3.	Librerías y herramientas utilizadas.	21
3.1.2.4.	Programa diseñado en Python.	22
3.1.3.	Fase III.	27
3.1.3.1.	Integración de programa en la Raspberry Pi Modelo B.	27
3.1.3.2.	Elaboración del circuito de conmutación.	27
3.1.4.	Fase IV.	29
3.1.4.1.	Visitas técnicas a la sede de DIMETEL.	29
IV. Análisis, interpretación y presentación de los resultados		33
4.1.	Pruebas estandarizadas <i>in situ</i> FMUC	33
4.1.1.	Prueba 1: Audio de la Cadena en condiciones ideales (SNR 96 dB).	34
4.1.2.	Prueba 2: Audio de la Cadena con cambio del orden del audio.	34
4.1.3.	Prueba 3: Audio de la Cadena añadiendo tono de 800Hz.	38
4.1.4.	Prueba 4: Audio de la cadena añadiendo tono de 440 Hz.	45
4.1.5.	Prueba 5: Audio de la cadena añadiendo ruido blanco.	55
4.1.6.	Definir umbral mínimo de nivel de detección.	69
4.2.	Dispositivo final de monitoreo y conmutación.	69
V. Conclusiones y recomendaciones		73
5.1.	Conclusiones	73
5.2.	Recomendaciones	74
Referencias Bibliográficas		77

- A. Especificaciones de la estructura de la Raspberry Pi Modelo B**
- B. Código principal del sistema de detección de patrones de audio para la identificación de cadenas nacionales de radio y televisión**
- C. Manual de usuario para el sistema de monitoreo y detección de cadenas nacionales de radio y televisión**

Índice de figuras

2.1. Ejemplo de espectrograma de una señal de audio [6]	13
2.2. Raspberry Pi [10]	13
2.3. Esquema Raspberry Pi Modelo B [10]	15
2.4. Esquema de los pines de entrada/salida de la Raspberry Pi Modelo B [10]	15
2.5. Adaptador USB de tarjeta de audio externa Canal 7.1 [12]	16
3.1. Diagrama general del programa en Python para la detección de cadenas nacionales de radio y televisión.	21
3.2. Diagrama completo del programa final en Python para la detección de cadenas nacionales de radio y televisión.	24
3.3. Espectrograma de la coetilla de entrada de la cadena nacional de radio y televisión.	25
3.4. Espectrograma de una muestra generada por el algoritmo.	26
3.5. Gráfica del barrido generada por el algoritmo.	26
3.6. Esquemático del circuito de conmutación.	28
3.7. Circuito de Conmutación.	29
3.8. Diagrama general estación FMUC.	30
3.9. Interfaz de Conmutación Balanceada.	31
4.1. Interfaz página web control remoto Programación FMUC.	70
4.2. Interfaz página web control remoto Programación FMUC.	71
4.3. Sistema final (hardware) a implementar de detección y conmutación de Cadenas Nacionales.	71

Indice de tablas

3.1. Resumen de librerías utilizadas en el sistema de monitoreo y detección de cadenas de radio y televisión	23
4.1. Pruebas de detección de la coletilla de entrada de la cadena con audio en condiciones ideales (SNR 96 dB).	35
4.2. Pruebas de detección de la coletilla de salida de la cadena con audio en condiciones ideales (SNR 96 dB).	36
4.3. Pruebas de detección de la coletilla de entrada de la cadena desordenada.	37
4.4. Pruebas de detección de la coletilla de salida de la cadena desordenada.	38
4.5. Pruebas de detección de la coletilla de entrada de la cadena añadiendo un tono de 800 Hz a -30 dBmV.	39
4.6. Pruebas de detección de la coletilla de salida de la cadena añadiendo un tono de 800 Hz a -30 dBmV.	41
4.7. Pruebas de detección de la coletilla de entrada de la cadena añadiendo un tono de 800 Hz a -10 dBmV.	42
4.8. Pruebas de detección de la coletilla de salida de la cadena añadiendo un tono de 800 Hz a -10 dBmV.	43
4.9. Pruebas de detección de la coletilla de entrada de la cadena añadiendo un tono de 800 Hz a -5 dBmV.	44
4.10. Pruebas de detección de la coletilla de salida de la cadena añadiendo un tono de 800 Hz a -5 dBmV.	45
4.11. Pruebas de detección de la coletilla de entrada de la cadena añadiendo un tono de 800 Hz a 0 dBmV.	46
4.12. Pruebas de detección de la coletilla de salida de la cadena añadiendo un tono de 800 Hz a 0 dBmV.	47
4.13. Pruebas de detección de la coletilla de entrada de la cadena añadiendo un tono de 440 Hz a -30 dBmV.	48
4.14. Pruebas de detección de la coletilla de salida de la cadena añadiendo un tono de 440 Hz a -30 dBmV.	49
4.15. Pruebas de detección de la coletilla de entrada de la cadena añadiendo un tono de 440 Hz a -10 dBmV.	50
4.16. Pruebas de detección de la coletilla de salida de la cadena añadiendo un tono de 440 Hz a -10 dBmV.	51

4.17. Pruebas de detección de la coletilla de entrada de la cadena añadiendo un tono de 440 Hz a -5 dBmV.	52
4.18. Pruebas de detección de la coletilla de salida de la cadena añadiendo un tono de 440 Hz a -5 dBmV.	53
4.19. Pruebas de detección de la coletilla de entrada de la cadena añadiendo un tono de 440 Hz a 0 dBmV.	54
4.20. Pruebas de detección de la coletilla de salida de la cadena añadiendo un tono de 440 Hz a 0 dBmV.	55
4.21. Pruebas de detección de la coletilla de entrada de la cadena añadiendo ruido blanco SNR 39 dB.	57
4.22. Pruebas de detección de la coletilla de salida de la cadena añadiendo ruido blanco SNR 39 dB.	58
4.23. Pruebas de detección de la coletilla de entrada de la cadena añadiendo ruido blanco SNR 19 dB.	59
4.24. Pruebas de detección de la coletilla de salida de la cadena añadiendo ruido blanco SNR 19 dB.	60
4.25. Pruebas de detección de la coletilla de entrada de la cadena añadiendo ruido blanco SNR de 14 dB.	61
4.26. Pruebas de detección de la coletilla de salida de la cadena añadiendo ruido blanco SNR de 14 dB.	62
4.27. Pruebas de detección de la coletilla de entrada de la cadena añadiendo ruido blanco SNR de 9 dB.	63
4.28. Pruebas de detección de la coletilla de salida de la cadena añadiendo ruido blanco SNR de 9 dB.	64
4.29. Pruebas de detección de la coletilla de entrada de la cadena añadiendo ruido blanco SNR de 4 dB.	65
4.30. Pruebas de detección de la coletilla de salida de la cadena añadiendo ruido blanco SNR de 4 dB.	66
4.31. Resumen de desempeño del sistema de monitoreo de cadenas nacionales, coletilla de entrada.	67
4.32. Resumen de desempeño del sistema de monitoreo de cadenas nacionales, coletilla de salida.	68

Acrónimos

CONATEL	Comisión Nacional de Telecomunicaciones
DSP	Digital Signal Processing
LOT	Ley Orgánica de Telecomunicaciones
RBPI	Raspberry Pi
RESORTE	Responsabilidad Social de Radio y Televisión
VTV	Venezolana de Televisión
SNR	Signal to Noise Ratio

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE DETECCIÓN
DE PATRONES DE AUDIO UTILIZANDO EL DISPOSITIVO
RASPBERRY PI MODELO B, PARA LA IDENTIFICACIÓN DE
CADENAS NACIONALES DE RADIO Y TV PARA SER INTEGRADO
EN LA CONMUTACIÓN AUTOMATIZADA DE LA
PROGRAMACIÓN DE LAS ESTACIONES FMUC Y UCTV**

por

LUZ MARÍA PENAGOS CARRILLO y WILDER DAVID JIMENEZ SILVA

Presentado en el Departamento de Señales y Sistemas
de la Escuela de Ingeniería en Telecomunicaciones
el 20 de Febrero del 2015 para optar al Título de
Ingeniero de Telecomunicaciones

RESUMEN

El presente proyecto plantea una solución automatizada para el problema que representa el factor humano a la hora de enlazar a las estaciones FMUC y UCTV, con los correspondientes mensajes del estado transmitidos en Cadena Nacional, a fin de cumplir con lo estipulado en la ley de responsabilidad social de Radio y Televisión (RESORTE). Para ello se implementó un sistema de detección de patrones de audio basado en procesamiento digital de señales (DSP), y de esta manera

identificar la entrada y salida de cadenas nacionales de radio y TV, para posteriormente conmutar de forma automática la señal del estado y difundirla por el espacio correspondiente del espectro radioeléctrico. Para esto se utilizó el dispositivo Raspberry Pi Modelo B (System on Chip), el sistema operativo Raspbian como entorno gráfico y el lenguaje de programación Python. Se realizaron pruebas de desempeño sobre diversos escenarios utilizando el software libre Adobe Audition CS6 para determinar el alcance del dispositivo, de los cuales se obtuvo una efectividad de 96.7 % para la detección de la coetilla de entrada en condiciones ideales (SNR 96 dB) y de 100.0 % para la coetilla de salida. También se observó que al perturbar las señales monitoreadas con ruido blanco su desempeño se veía afectado, sin embargo alcanzó una efectividad de hasta 73.4 % para valores de SNR de 14 dB. Adicional a esto se elaboró un sistema de respaldo remoto, para la conmutación vía web.

Palabras Claves: Patrones de audio, Procesamiento Digital de Señales, Raspberry Pi, System on Chip, Python, Radiodifusión, Coetilla, Relación Señal a Ruido, Ruido Blanco

Tutor: EDUARDO GONZÁLEZ

Profesor del Departamento de Señales y Sistemas

Escuela de Telecomunicaciones. Facultad de Ingeniería

Capítulo I

El Problema

1.1. Motivación

En Venezuela el cumplimiento de la Ley de Responsabilidad Social de Radio y Televisión (RESORTE) es un requisito que deben acatar los servicios de radiodifusión sonora en amplitud modulada (AM), radiodifusión sonora en frecuencia modulada (FM), radiodifusión sonora comunitaria de servicio público, televisión UHF, televisión VHF, servicios de difusión por suscripción, entre otros servicios de medios de comunicaciones. En esta ley RESORTE, aprobada por la Asamblea Nacional de Venezuela, establece en el artículo 10 que “El Estado podrá difundir sus mensajes a través de los servicios de radio y televisión (Cadena Nacional). A tales fines, podrá ordenarle a los prestadores de estos servicios la transmisión gratuita de los mensajes y alocuciones del Estado, ordenados por el órgano rector del Ejecutivo Nacional, con competencia en la materia para la comunicación y la información” [1].

En los casos más comunes de los medios de comunicación nacionales de radio y TV, para prestar el servicio de transmisión gratuita de mensajes del Estado, se cuenta con un operador, el cual se encarga de enlazar la señal del canal del Estado, operando un sistema que depende exclusivamente de la capacidad y agilidad humana. Esto hace que se puedan producir retrasos en la conexión a cadena, y sean

detectados por el ente regulador CONATEL, el cual tendrá la potestad de recurrir a sanciones por incumplimiento de ley a todo medio prestador de servicios de radio y TV, que no se encuentre difundiendo en su respectivo espectro radioeléctrico las alocuciones del Estado. La radio Universitaria 104,5 FM (FMUC) y Televisión Universitaria (UCTV) requieren de un sistema de detección de patrones de audio basado en Procesamiento Digital de Señales (DSP), con el fin de identificar la entrada y salida de cadenas nacionales de radio y TV, para posteriormente activar un sistema de conmutación automatizada mediante el cual se enlaza la señal del canal del Estado, para difundir en cadena nacional, cumpliendo con el artículo 10 de la ley RESORTE y el artículo 192 de la Ley Orgánica de Telecomunicaciones [2] los mensajes y alocuciones transmitidos por el Estado.

Las empresas prestadoras de servicios de radiodifusión y teledifusión, siempre en búsqueda de la optimización y mejora de la calidad de sus servicios, se atribuyen la necesidad de reducir el error por factor humano en su máxima posibilidad. Este sistema estará integrado a una plataforma de conmutación automatizada de canales, en el que se busca de manera precisa cumplir con el objeto de eliminar en su totalidad las fallas de enlace que pueda ocasionar un operador. A su vez este sistema puede beneficiar a todo el sector de las empresas prestadoras de servicio de radiodifusión y teledifusión, ya que puede ser implementado para todo tipo de medio de comunicación, tal como teledifusoras UHF, VHF, radios comunitarias AM y FM, entre otras. Este trabajo genera diversos tipos de beneficios, ya que busca consolidar al sector de los medios de comunicación ante cualquier imposición de tipo económica o social, a su vez es un aporte a la investigación en el área de procesamiento digital de señales ya que se plantea obtener la detección de una señal de audio mediante la comparación de señales utilizando la transformada de Fourier y la correlación. A la vez contribuye al desarrollo tecnológico en esta área debido a la realización de un sistema automatizado para el monitoreo y detección de cadenas de radio y TV.

1.2. Objetivos

1.2.1. Objetivo General

Diseñar e implementar un sistema de detección de cadenas nacionales de radio y TV. utilizando el dispositivo Raspberry Pi Modelo B para la conmutación automatizada.

1.2.2. Objetivos Específicos

- Identificar los parámetros de una señal de audio que permita generar criterios para la detección de patrones.
- Diseñar el algoritmo que permita la identificación de las señales de audio de cadenas nacionales.
- Implementar el sistema de detección de patrones de audio utilizando el dispositivo Raspberry Pi Modelo B para la identificación de cadenas.
- Implementar una interfaz de hardware para interconectar la RBPi y la consola mezcladora de canales.

1.3. Alcance

El propósito y objetivo de este trabajo es diseñar e implementar un sistema de detección de patrones de audio para la identificación de cadenas nacionales de radio y TV, con la finalidad de automatizar el sistema utilizado actualmente en las estaciones de FMUC y UCTV que está a cargo de un operador. En el diseño se incluye la programación del dispositivo Raspberry Pi Modelo B (mini computadora basada en DSP) y el desarrollo de una interfaz elemental de hardware para la interconexión con el sistema de gestión. Este sistema cuenta con una serie de patrones de audio almacenados los cuales se utilizaran para la comparación y detección de cadenas

nacionales. Si estos patrones son modificados, se generaran fallas en la comparación de las señales de audio, lo cual conllevaría a una modificación o adecuación en el dispositivo.

1.4. Recursos Utilizados

Para este proyecto se utilizaron diversos recursos literarios enfocados al tema de Procesamiento Digital de Señales y al análisis de patrones de señales de audio. Se programó la Raspberry Pi modelo B (RBPi) basada en DSP para realizar la comparación de las señales de audio, la identificación de cadenas nacionales y la ejecución de la conmutación automatizada de la programación de las estaciones FMUC y UCTV. Como software, se utilizó Raspbian, el cual es un sistema operativo derivado de la distribución de Linux y es básicamente una versión modificada de Debian, adaptada para los requisitos de programación de la RBPi. Este sistema operativo soporta el lenguaje Python que se usó para el diseño del algoritmo de comparación. Para las pruebas de desempeño del equipo se empleó el software libre Adobe Audition CS6. Se contó con el apoyo y asesoría del Profesor Eduardo González, adicionalmente se recibió el soporte de los departamentos de Ingeniería, Rediuc (Red Dorsal Digital Integrada de la Universidad de Carabobo) y FMUC, adscritos a DIMETEL (Dirección de Medios Electrónicos y Telemática de la Universidad de Carabobo).

Capítulo II

Marco conceptual

2.1. Antecedentes

Para la realización del presente trabajo se consideró conveniente hacer un análisis de algunos proyectos de grado y de trabajos de investigación los cuales guardan cierta similitud con la investigación en cuestión. Los mismos fueron los siguientes:

- Sanabria, A., Pedraza, C., Vitola, J., "Algoritmo rápido para la búsqueda de audio por contenido sobre GPU's". Artículo derivado del proyecto de investigación "Búsqueda de audio por contenido" por INVTEL. 2011.

En este trabajo se desarrolló un algoritmo para la comparación de señales de audio, mediante el uso de la transformada de Fourier y el método de la correlación, lo cual resulta de gran utilidad para este trabajo de grado.

- Macías, J., "Diseño e implementación de un detector automático de Acordes". Tesis de Pregrado Universidad Carlos III de Madrid. 2012.

En este proyecto se habla de la toma de muestras de audio y el análisis frecuencial de cada una de ellas, lo cual presenta similitud con lo que se realizó en el presente trabajo.

- Cano, P., Battle, E., Kalker, T., Haitzma, J., “A Review of audio Fingerprinting” Universitat Pompeu Fabra, Barcelona, España. 2005.

En este proyecto se estudiaron las firmas digitales de audio para la búsqueda de información y comparación de patrones de audio. Se estudian diferentes técnicas para la extracción de las firmas digitales (Audio Fingerprints) de las piezas de audio para su posterior análisis.

2.2. Marco Teórico

2.2.1. Leyes y Sanciones

El 5 de septiembre de 1991 mediante el decreto N°1826 en la Gaceta Oficial N°34801 fue creada la Comisión Nacional de Telecomunicaciones (CONATEL) ente moderador del estado para la regulación, supervisión y control de las telecomunicaciones en Venezuela. En junio de 2000 fue promulgada por la Asamblea Nacional la Ley Orgánica de Telecomunicaciones (LOT) en la Gaceta Oficial N°36970 en la cual le confiere a CONATEL el rango de Instituto autónomo, con personalidad jurídica, patrimonio propio e independiente del Fisco Nacional, autonomía técnica, financiera, organizativa y administrativa [2]. Adicional a esto en el artículo 37 del Capítulo II de la LOT en el punto 2 se establece lo siguiente: “Son competencias de la Comisión Nacional de Telecomunicaciones las siguientes: Velar por el cumplimiento de las disposiciones de esta Ley, de las leyes que la desarrollen, de los reglamentos y demás actos que dicte la Comisión cuya vigilancia le compete”. En el artículo 192 del Capítulo I: Disposiciones Finales del Título XIII: Disposiciones Finales y Transitorias de la LOT se establece: “Sin perjuicio de las disposiciones legales en materia de seguridad y defensa, el Presidente de la República podrá, directamente o a través de la Comisión Nacional de Telecomunicaciones, ordenar a los operadores que presten servicios de televisión por suscripción, a través del canal de información a sus clientes y a las empresas de radiodifusión sonora y televisión abierta la transmisión gratuita de mensajes o alocuciones oficiales, de la

Presidencia o Vicepresidencia de la República o de los Ministros. Mediante reglamento se determinarán las modalidades, limitaciones y demás características de tales emisiones y transmisiones. No estará sujeta a la obligación establecida en este artículo la publicidad de los entes públicos” [2]. En el Capítulo II De las Sanciones Administrativas, Sección Primera. De las Infracciones Administrativas y sus Sanciones de la Ley Orgánica de Telecomunicaciones, se encuentra el artículo 166 el cual establece que: “Será sancionada con multa por hasta cincuenta mil Unidades Tributarias (50000 UT) de conformidad con lo que prevea el reglamento de esta Ley: 6. La abstención de un operador a acatar en forma inmediata la orden de la Comisión Nacional de Telecomunicaciones de permitir la interconexión con las redes de otro operador en los términos y condiciones específicas que establezca al efecto, en los casos previstos en esta Ley” [2].

Debido a que la transmisión de una cadena nacional o alocución oficial para un medio de radiodifusión privado depende o se basa en realizar una interconexión con la señal del estado o el medio oficial por el cual se establecen inicialmente las alocuciones oficiales, se entiende del artículo 166 de la LOT, citado anteriormente que al no realizar esta interconexión con la señal del medio del estado para transmitir el mensaje oficial se está expuesto a recibir sanciones y multas por medio de la Comisión Nacional de Telecomunicaciones.

En diciembre de 2010 se aprueba por la Asamblea Nacional una nueva Ley referente a las Telecomunicaciones en Venezuela, con un enfoque más social pero aún así con competencias administrativas llamada Ley de Responsabilidad Social en Radio y Televisión (RESORTE); el 7 de febrero de 2011 por un decreto en la Gaceta Oficial N°39610 es reformada parcialmente ésta Ley, la cual establece en el artículo 10 del Capítulo II lo siguiente: “El Estado podrá difundir sus mensajes a través de los servicios de radio y televisión. A tales fines, podrá ordenarle a los prestadores de estos servicios la transmisión gratuita de: 1. Los mensajes previstos en la Ley Orgánica de Telecomunicaciones. La orden de transmisión gratuita y obligatoria de mensajes o alocuciones oficiales podrá ser notificada válidamente, entre otras formas, mediante la sola difusión del mensaje o alocución a través de los servicios de radio o televisión administrados por el Ejecutivo Nacional” [1].

Es importante recalcar que al referirse a “servicios de radio y televisión” se entienden los servicios de radiodifusión sonora en amplitud modulada (AM), radiodifusión sonora en frecuencia modulada (FM), radiodifusión sonora comunitaria de servicio público, televisión UHF, televisión VHF, servicios de difusión por suscripción según lo establecido por la Comisión Nacional de Telecomunicaciones.

Toda violación o incumplimiento de una Ley conlleva a sanciones y penalizaciones, sin ser excepción la Ley de Responsabilidad Social en Radio y Televisión, es por esto que en el Capítulo VII Del Procedimiento Administrativo Sancionatorio en el punto 4 del artículo 28 se observa el siguiente enunciado: “Se sancionará al prestador de servicios de radio, televisión o difusión por suscripción, en los casos que le sea aplicable, con multa desde tres por ciento 3 % hasta cuatro por ciento 4 % de los ingresos brutos causados en el ejercicio fiscal, inmediatamente anterior a aquel en el cual se cometió la infracción, así como con cesión de espacios para la difusión de mensajes culturales y educativos, cuando: 1) Incumpla con la obligación de difundir los mensajes del Estado, según lo previsto en el artículo 10 de esta Ley. 2) Interfiera los mensajes y alocuciones del Estado, infringiendo lo previsto en el artículo 10 de esta Ley” [1].

2.2.2. DIMETEL

Es una Dirección adscrita a la Dirección Superior de la Universidad de Carabobo con el fin de brindar a la comunidad universitaria y el público de la región, un servicio de calidad en los medios masivos de Radio y Televisión, así como en los medios dedicados de telefonía, conectividad intra y extra universitaria, radiocomunicaciones móviles, entre otros [3].

2.2.2.1. FMUC Emisora Universitaria 104,5 FM

Estación de radio FM adscrita a DIMETEL, se transmite por la frecuencia 104,5 FM en los estados Aragua, Carabobo y Cojedes. Transmite contenido educativo,

científico, tecnológico, cultural y artístico que se produce en la Universidad de Carabobo [3].

2.2.3. Comparación de Patrones

2.2.3.1. Técnica de Comparación de Patrones

Consiste en la comparación directa entre las características espectrales de la señal y las almacenadas en una base de datos existente, viene dada por una etapa de muestreo de la señal, una etapa de cálculo de propiedades espectrales y posteriormente comparar con una base de datos de patrones preestablecidos; se utilizan criterios o normas de decisión para llegar a un resultado preciso [4]. Como desventaja o limitación de este método se encontró que:

- Se debe registrar la señal de audio con la cual se va a comparar en sus componentes espectrales en la base de datos.
- Los patrones se deben construir en un conjunto de sonidos o fonemas cortos.
- Este tipo de trabajo tiene problemas de normalización de las señales, generando un error considerable si no se restringen las condiciones al registrar la señal.

2.2.3.2. Técnica de Reconocimiento por modelos ocultos de Markov

Esta estrategia se basa en el uso de elementos estocásticos, a continuación se muestran los puntos más relevantes de esta técnica:

- Se utilizan sistemas autónomos de estados finitos. El registro de estas señales contiene cantidad de ruido ambiental que debe ser modelado.
- Los modelos de Markov constan de un proceso estocástico con un modelo visible y un módulo oculto, por ello su nombre. En este caso aparece el algoritmo de identificación conocido como ME (Máxima Estimación).

- En la actualidad estos modelos han evolucionado a modelos Baum-Welch y Viterbi. Estos algoritmos están mejorados y adecuados al procesamiento de señales [5].

2.2.3.3. Transformada de Fourier

El concepto del dominio de la frecuencia es fundamental para entender las señales discretas y el comportamiento de los sistemas LIT. El espectro de una señal nos enseña cómo es esa señal en el dominio frecuencial; la respuesta en frecuencia de un sistema nos aporta el conocimiento de cómo se comporta ese sistema para diferentes entradas, gracias a la perspectiva que aporta el dominio de la frecuencia [6]. Es por esto que la transformada de Fourier es una potente herramienta en el procesamiento digital de señales. Sea $f(t)$ una función integrable se define como su transformada de Fourier, lo siguiente [6]:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt \quad (2.1)$$

La inversa de la transformada de Fourier, es decir la obtención de la señal en el dominio del tiempo a partir del dominio de la frecuencia viene dada por [5]:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)e^{j\omega t} d\omega \quad (2.2)$$

2.2.3.4. Correlación

Como método para la comparación de dos señales de audio es necesario definir un proceso llamado correlación, se define como un indicador estadístico determinado por un coeficiente de correlación R y es medido en una escala de -1 a +1. El valor de +1 indica una correlación perfecta y directa; en cambio el valor de -1 indica que hay una correlación perfecta e inversa. El valor de $R=0$, significa ausencia de correlación entre las variables, lo cual indica que las variables son independientes entre sí. El análisis de correlación puede aplicarse cuando se disponen de variables continuas o discretas de muchos valores donde se quiere saber si están asociadas o no

[7]. Desde el punto de vista de análisis de señales se desarrolla el significado de la correlación como medida de parecido entre señales, se establecen las propiedades de la correlación cuando las señales bajo estudio están relacionadas por sistemas lineales e invariantes. La función de correlación para dos señales continuas viene dada por la siguiente ecuación [8]:

$$R_{xy}(t) = \int_{-\infty}^{\infty} x(\tau)y(\tau - t)d\tau \quad (2.3)$$

$$R_{yx}(t) = \int_{-\infty}^{\infty} y(\tau)x(\tau - t)d\tau \quad (2.4)$$

Para señales discretas la función de correlación viene dada por lo siguiente [8]:

$$R_{xy}[n] = \sum_{k=-\infty}^{\infty} x[k]y[k - n] \quad (2.5)$$

$$R_{yx}[n] = \sum_{k=-\infty}^{\infty} y[k]x[k - n] \quad (2.6)$$

Con $n = 0, \pm 1, \pm 2, \dots$

2.2.3.5. Autocorrelación

De igual forma existe la Autocorrelación, que no es más que la correlación de una señal con ella misma, ésta resulta de gran utilidad para encontrar patrones repetitivos dentro de una señal [8]:

$$R_{xx}(t) = \int_{-\infty}^{\infty} x(\tau)x(\tau - t)d\tau \quad (2.7)$$

Para señales discretas, la función de autocorrelación viene dada por:

$$R_{xx}[n] = \sum_{k=-\infty}^{\infty} x[k]x[k - n] \quad (2.8)$$

2.2.3.6. Correlación Cruzada

La correlación cruzada es una medida de la similitud entre dos señales, frecuentemente usada para encontrar características relevantes en una señal desconocida por medio de la comparación con otra que sí se conoce. Se define la función de correlación mutua o cruzada entre dos señales de energía finita (potencia media nula) la siguiente [8]:

$$R_{xy}(t) = \int_{-\infty}^{\infty} x(\tau + t)y^*(\tau)d\tau = \int_{-\infty}^{\infty} x(\tau)y^*(\tau - t)d\tau \quad (2.9)$$

Donde : * implica el conjugado de la función

Para señales discretas, la función de correlación cruzada viene dada por:

$$R_{xy}[n] = \sum_{k=-\infty}^{\infty} x[k + n]y^*[k] = \sum_{k=-\infty}^{\infty} x[k]y^*[k - n] \quad (2.10)$$

2.2.4. Espectrograma

Un espectrograma es una representación visual del espectro de frecuencias en una señal de sonido o de otro, ya que varían con el tiempo o alguna otra variable. Se utilizan ampliamente en el desarrollo de los campos de la música, sonar, radar y el procesamiento del habla, etc [6]. En la figura 2.1 se muestra una gráfica ejemplo de un espectrograma:

2.2.5. Análisis de audio utilizando algoritmos de procesamiento de imágenes

Soundhound y *Shazaam* son conocidos programas a nivel mundial de reconocimiento de audio, ambos sistemas realizan el proceso de análisis de plantillas de imágenes que poseen la información del audio, es decir espectrogramas, por medio de programas especializados logran generar la identificación de una coincidencia en los patrones de audio plasmados en arreglos 2D [9].

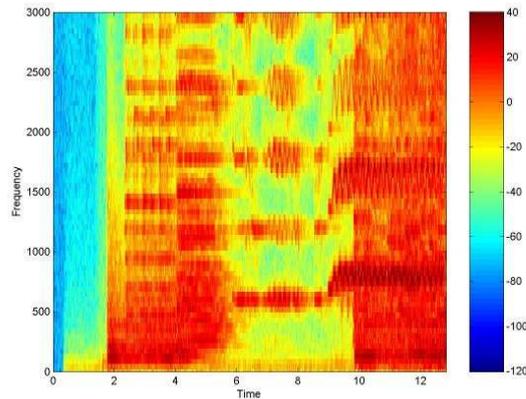


Figura 2.1: Ejemplo de espectrograma de una señal de audio [6]

2.2.6. Raspberry Pi Modelo B

La Raspberry Pi es una “computadora Linux de ultra bajo costo del tamaño de una tarjeta de crédito” [10] creado principalmente con el propósito de enseñar a los niños cómo programar computadoras. Este dispositivo ha sido desarrollado por la Fundación Raspberry Pi (Raspberry Pi Fundación, Cambridge, Reino Unido, Número de registro 1129409), que es una organización benéfica cuyo objetivo es fomentar la informática y otros temas relacionados en particular a nivel de la escuela.



Figura 2.2: Raspberry Pi [10]

Existen varios modelos de la Raspberry Pi actualmente en el mercado, el modelo B posee 512 Mb de RAM, dos puertos USB y un puerto Ethernet y se adquirió

en Venezuela a un precio de 4500 Bs para el año 2014. El Raspberry Pi incluye periféricos como puertos de comunicación con protocolos estándar de la electrónica y el lenguaje de programación de alto nivel capaz de ejecutar instrucciones exigidas por los algoritmos. Las características de este dispositivo son las siguientes [10]:

- SoC (System on Chip) Broadcom BCM2835 (CPU, GPU, DSP y SDRAM).
- CPU: 700MHz ARM1176JZF-S core (De la familia ARM11).
- Procesador gráfico Videocore IV.
- Memoria SDRAM: 512 MB
- Memoria (SD) mínimo: 6 GB, recomendado: 8 GB, máximo: 32 GB.
- Almacenamiento en tarjeta/ Almacenamiento a través: SD, MMC, ranura para tarjetas SDIO.
- 10/100 Ethernet RJ-45 incluido.

Otra característica de este dispositivo es que puede ejecutar distintos sistemas operativos de la distribución de Linux y a su vez soporta una gran variedad de lenguajes de programación como Python, C++, entre otros. Esto significa que el dispositivo es capaz de realizar varios cálculos matemáticos y dar resultados prácticamente instantáneamente debido a su procesador de 700 MHz. Como desventaja principal se puede señalar que la Raspberry Pi no incluye convertidor Analógico/Digital (ADC) [11]. A continuación se muestra un esquema de la Raspberry Pi Modelo B y sus puertos:

Con unas dimensiones de placa de 8.5 cm por 5.3 cm, el modelo B de la Raspberry Pi es capaz de generar altas capacidades gráficas similares a la consola XBOX de Microsoft y con la posibilidad de reproducir vídeo en 1080p. La Raspberry Pi cuenta a su vez con una salida de vídeo y audio a través de un conector HDMI, con lo que se consigue conectar la tarjeta tanto a televisores como a monitores que cuenten con dicha conexión. En cuanto a vídeo se refiere, también cuenta con una salida de vídeo compuesto y una salida de audio a través de un minijack de 3.5

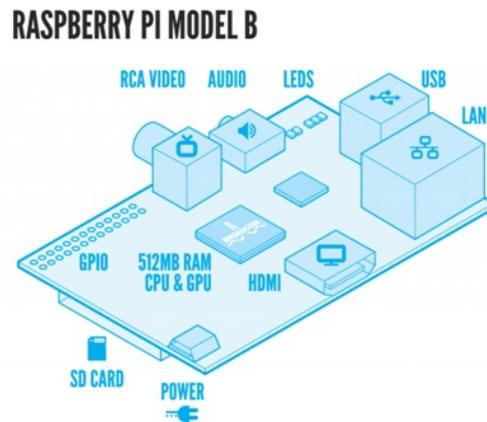


Figura 2.3: Esquema Raspberry Pi Modelo B [10]

mm. Este dispositivo requiere componentes adicionales que deben ser comprados por separado, tales como: una tarjeta SD, teclado y ratón, adaptador USB externo y cables HDMI o RCA de video compuesto [11].

La Raspberry Pi Modelo B posee 26 pines que funcionan como interfaces físicas entre la RBPi y algún circuito externo. Diecisiete (17) de los veintiseis (26) pines son GPIO (General Porpouse Input/Output). En la figura se observa un diagrama de estos pines.

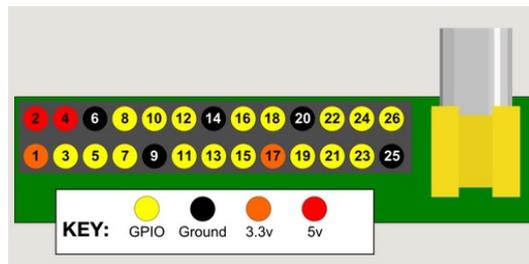


Figura 2.4: Esquema de los pines de entrada/salida de la Raspberry Pi Modelo B [10]

En el anexo A se encuentran extensas especificaciones de la Raspberry Pi Modelo B.

2.2.7. Tarjeta de audio

En vista de que el dispositivo Raspberry Pi no tiene una entrada de audio física disponible, es necesario contar con otros tipos de solución. Para ello se cuenta con un adaptador USB de tarjeta de audio externa Canal 7.1, el cual cuenta con una entrada y una salida de audio 3.5 mm externa. Su incorporación añade un convertor analógico/digital, no necesita de drivers para ningún sistema operativo, lo cual lo hace totalmente compatible con Linux [12].



Figura 2.5: Adaptador USB de tarjeta de audio externa Canal 7.1 [12]

2.2.8. Entorno de trabajo: Raspbian

Al hablar de Raspberry Pi, es necesario conocer sus sistemas operativos, entre estos; Raspbian es conocido como el distro oficial de la Raspberry Pi [13]. Es básicamente una distribución del sistema operativo GNU/Linux y por lo tanto libre basado en Debian Wheezy (Debian 7.0), orientado a la enseñanza de informática. Su lanzamiento inicial fue en junio de 2012. Hasta este punto, Debian era la distro recomendada para dar los primeros pasos con el hardware, pero existían ciertas limitaciones, ya que con esa versión no se podía aprovechar la unidad de punto flotante en el SoC Broadcom BCM2835 que usa el Raspberry Pi. Sin embargo, con la

llegada de Raspbian, los usuarios podrán contar con una distro especialmente optimizada para este dispositivo. El promedio del salto en el rendimiento obtenido bajo Raspbian está en el orden del 18 %, pero en ciertas aplicaciones, como la decodificación mp3, el aumento supera el 40 %. La distribución usa LXDE (Lightweight X11 Desktop Environment) como escritorio y Midori como navegador web. Además contiene herramientas de desarrollo como IDLE para el lenguaje de programación Python o Scratch. Para su instalación, se puede hacer uso de la aplicación NOOBS (New Out Of the Box Software), la cual contiene todas las imágenes disponibles del sistema operativo, NOOBS es una aplicación que facilita la instalación de diversas distribuciones Linux, hace innecesario el acceso a internet durante la instalación, lo cual facilita muchas cosas, tan solo se debe descargar y descomprimir en una tarjeta SD de al menos 4 GB de capacidad que se integrará a la Raspberry Pi. Al hacerlo se dará la opción de instalar Raspbian, además de otros distros de Linux como Arch Linux, RaspBMC, el recién salido Pidora u OpenELEC.

2.2.9. Python

Como se mencionó anteriormente, una de las ventajas de utilizar Raspbian es que es capaz de ejecutar diferentes tipos de lenguaje de programación de alto nivel, entre ellas Python, Java, C++, Groovy, entre otras. Python fue elegido como el lenguaje de programación para implementar los algoritmos. Este lenguaje de programación es similar a C++, pero es más interactivo a la hora aprender y aplicar. Este lenguaje de programación es libre de usar debido a su licencia de código abierto [14]. Se trata de un lenguaje de programación multi-paradigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma. Los programas son muy compactos, un programa en Python suele ser mucho más corto que su equivalente en lenguajes como C++. Es también conocido como un lenguaje de scripts, sencillo pero potente. Un script es un conjunto de instrucciones que se ejecutan paso a paso, instrucción a instrucción. Esto significa que Python no genera ejecutables, si no que es él mismo, el encargado de ejecutar el código.

Capítulo III

Procedimientos de la investigación

3.1. Procedimiento Metodológico.

En el presente proyecto de grado se planteó el diseño e implementación de un sistema de detección de patrones de audio utilizando el dispositivo Raspberry Pi Modelo B, para la identificación de cadenas nacionales de radio y TV para ser integrado en la conmutación automatizada de la programación de las estaciones FMUC y UCTV. Por lo cual se desarrollaron los objetivos específicos en fases de la siguiente manera:

3.1.1. Fase I.

Recopilación de documentos e información referente a comparación de patrones de audio, proceso de correlación, elección del entorno de trabajo y lenguaje de programación, selección de los periféricos o dispositivos complementarios para el correcto funcionamiento del dispositivo basado en DSP.

3.1.2. Fase II.

3.1.2.1. Selección del algoritmo de comparación de señales de audio.

Para realizar una comparación de señales de audio y verificar la relación entre ellas se utilizó el método de la correlación cruzada en el dominio de la frecuencia, que no es más que la relación existente entre dos variables. La elección de este método se basó en la alta precisión, el fácil entendimiento de sus resultados, el uso de los valores de salida para la toma de decisiones y su baja vulnerabilidad al ruido, además de la capacidad de cómputo del dispositivo Raspberry Pi Modelo B. Se utilizó la transformada de Fourier para llevar la señal de entrada al dominio frecuencial. Con la intención de generar un sistema de comparación más robusto, adicionalmente se realizó un análisis de audio con algoritmos de procesamiento de imágenes, que es la comparación de los espectrogramas de las respectivas señales.

3.1.2.2. Descripción del algoritmo seleccionado.

Para la correcta selección del entorno de trabajo y el lenguaje a utilizar es necesario el planteamiento del algoritmo que se eligió como fundamento del proyecto. Para el desarrollo del programa encargado del cómputo de comparación de señales de audio basado en la correlación se diseñó el siguiente flujograma de manera general:

Como se observa en la figura 3.1, el algoritmo para la comparación de la señales de audio se puede dividir en cuatro (4) grandes bloques:

- Señal de entrada. Se recibe la señal a procesar.
- Muestreo y procesamiento. Selección del tamaño de muestras y parametrización de la señal.
- Comparación por correlación. Comparación de la señal en la base de datos y la señal recibida, comprobación de la existencia de relación.
- Decisión. Orden de activación de la conmutación.

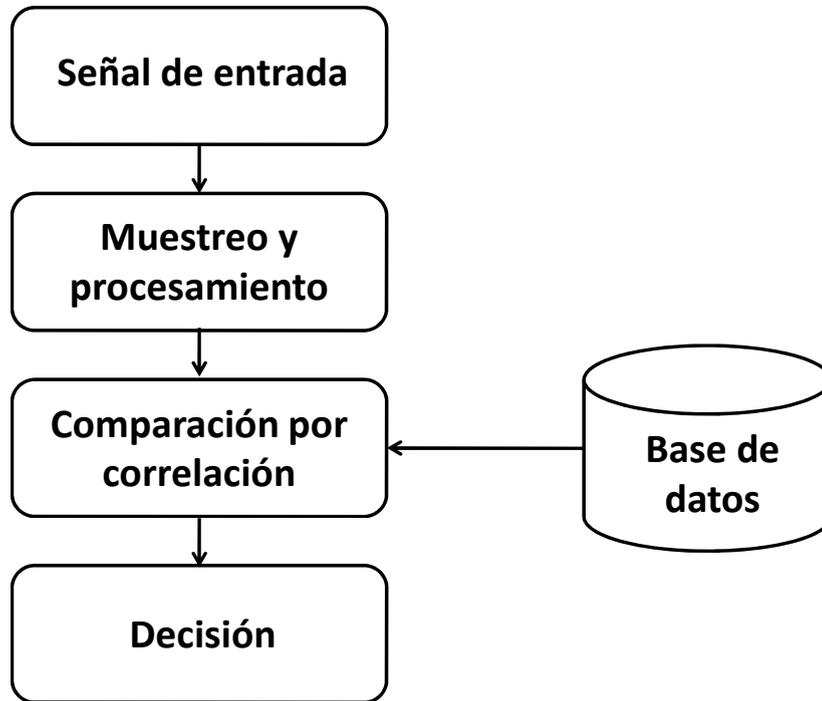


Figura 3.1: Diagrama general del programa en Python para la detección de cadenas nacionales de radio y televisión.

3.1.2.3. Librerías y herramientas utilizadas.

Debido a las características de la Raspberry Pi Modelo B, dispositivo seleccionado por sus múltiples ventajas mencionadas en el capítulo II, se eligió como lenguaje de programación a Python ya que es un lenguaje de alto nivel conveniente para un desarrollo rápido de código en el que varios estilos de programación son compatibles es gratuito y multiplataforma. La ventaja que más resalta a la vista de utilizar este lenguaje es la existencia y disponibilidad de múltiples librerías que facilitan notablemente una gran cantidad de cálculos matemáticos. Las librerías utilizadas para este proyecto fueron las siguientes:

- Numpy.
- Scipy.

- PyAudio.
- Matplotlib.
- Skimage.
- RPi.GPIO.
- Time.

En la tabla 3.1 se tiene un resumen con las funciones principales de las librerías utilizadas en Python.

3.1.2.4. Programa diseñado en Python.

Para una visualización más concreta del programa diseñado en Python se hace referencia a la figura 3.2 donde se observa el proceso detallado del algoritmo desarrollado y se explica cada etapa del mismo:

Para el desarrollo del código de detección de patrones de audio se tomaron en cuenta tres partes diferentes. En la primera se trabaja exclusivamente con el archivo de audio almacenado (.wav) a detectar, en la segunda parte se elabora la extracción de muestras de audio en tiempo real, las cuales posteriormente serán correlacionadas con el archivo de audio, lo que indicará dependiendo de los resultados, la existencia del audio en estudio durante la transmisión en tiempo real, esto último pertenece a la tercera parte.

En principio se define una función de nombre *spectrogram* la cual servirá para tomar el archivo de audio almacenado y crear una variable donde se guardará el espectrograma completo de este archivo, esta variable será posteriormente utilizada para verificaciones de la detección del patrón. Por otra parte, se extrae una muestra equivalente a 20 milisegundos la cual será normalizada para aplicarle una transformada rápida de Fourier. El valor absoluto de esta muestra es almacenada.

Tabla 3.1: Resumen de librerías utilizadas en el sistema de monitoreo y detección de cadenas de radio y televisión

Librería	Descripción	Función	Salida de la Función
Numpy	Librería con gran cantidad de paquetes para el cómputo científico, arreglos de vectores y matrices	Corrcoef, abs, Hamming	Arroja el coeficiente de correlación, valor absoluto, ventana de Hamming
Scipy	Librería de rutinas numéricas, módulos de optimización y álgebra.	Scipy.fftpack	Arroja la transformada de Fourier de un arreglo
PyAudio	Librería para manejo de archivos de audio en Python	PyAudio.Open	Permite abrir con un formato específico un archivo de audio
Matplotlib.pyplot	Generación de gráficos a partir de datos en línea o arreglos	plot, show	Si se desea muestra el espectrograma y el punto de coincidencia
Skimage	Librería con colección de algoritmos para procesamiento de imágenes	skimage.feature	Genera un vector cuyo máximo indica donde existe mayor correlación entre imágenes
RPI.GPIO	Librería que activa las funciones para utilizar los GPIO de la RBPi	setmode, setup, out	Activa como salida o entrada un pin GPIO para generar la conmutación
Time	Librería con variables de tiempo	sleep	Suspende la ejecución del programa según los segundos indicados

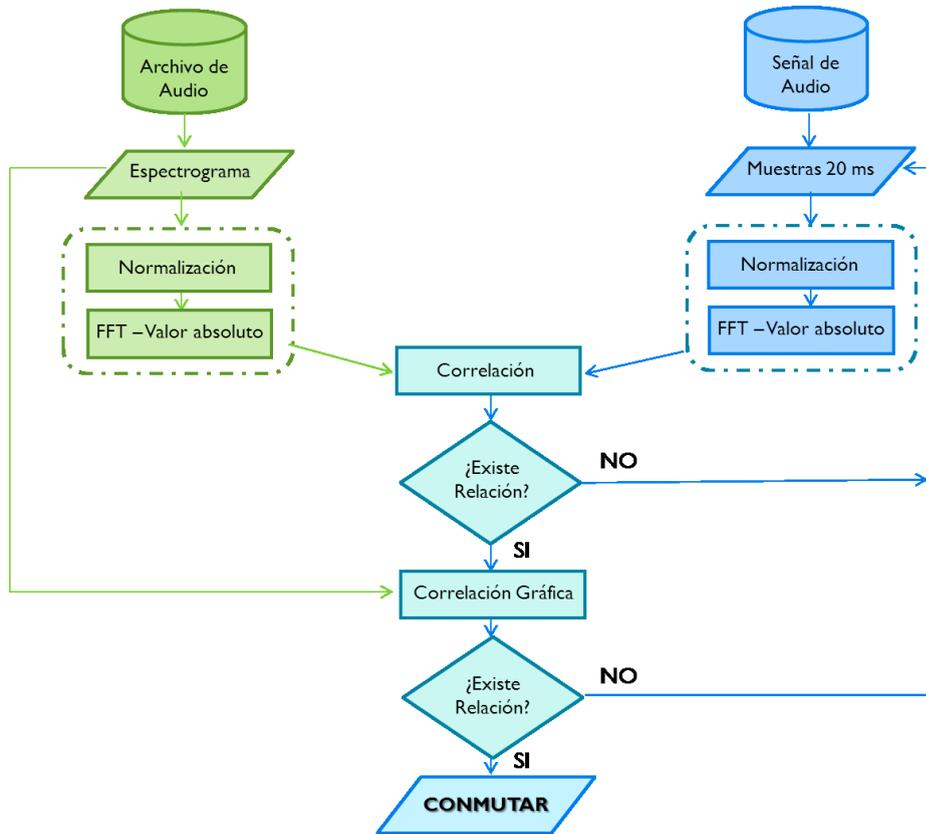


Figura 3.2: Diagrama completo del programa final en Python para la detección de cadenas nacionales de radio y televisión.

En la segunda parte, se extraen las muestras a comparar durante un ciclo infinito, cada una de estas muestras pasan por el mismo proceso aplicado en la primera parte, y van siendo almacenadas en una variable *buffer*. Adicionalmente se extraen muestras equivalentes a un segundo, estas serán procesadas con la función *spectrogram* y servirán para una posterior verificación.

Durante el mismo ciclo infinito se procede a la última parte, la cual realizará la correlación cruzada entre las muestras extraídas y la almacenada del archivo de audio. En este proceso se establece un nivel de umbral el cual al ser superado indica la posibilidad de un *match*, el cual activa el proceso de verificación, en el que se toman los dos espectrogramas de las dos primeras partes y se evalúan con

una correlación gráfica, la cual funciona como un barrido de toda la señal, para confirmar o no la existencia del match y de esta manera pasar a activar el proceso de conmutación.

Para fines ilustrativos en la figura 3.3 se observa el espectrograma de la coletilla de entrada de la cadena nacional, la cual tiene una duración de 19 s, almacenada en la RBPi para realizar la búsqueda de coincidencia con la muestra tomada.

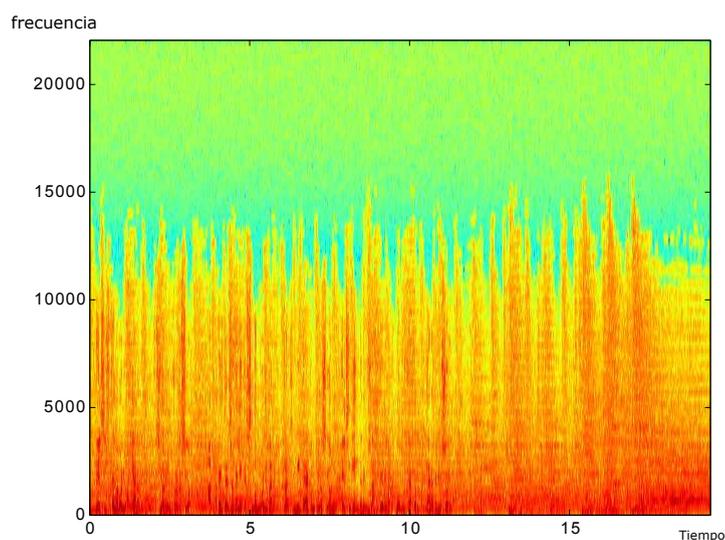


Figura 3.3: Espectrograma de la coletilla de entrada de la cadena nacional de radio y televisión.

En la figura 3.4 se observa el espectrograma de una muestra generada por la función *sample spec* para realizar el barrido sobre el espectrograma almacenado de la figura 3.3.

En la figura 3.5 se observa el espectrograma del barrido para ese ejemplo en particular y se observa un pico significativo en la gráfica que representa el punto de coincidencia de la muestra tomada sobre el espectrograma de la coletilla almacenado.

Para la toma de decisión se realizaron pruebas en ambiente controlado para determinar el valor de los umbrales que deben cumplirse cuando existe una relación entre las dos señales.

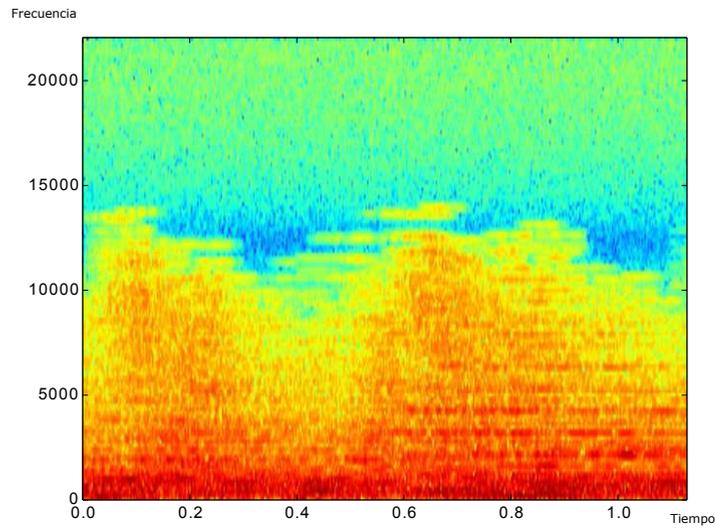


Figura 3.4: Espectrograma de una muestra generada por el algoritmo.

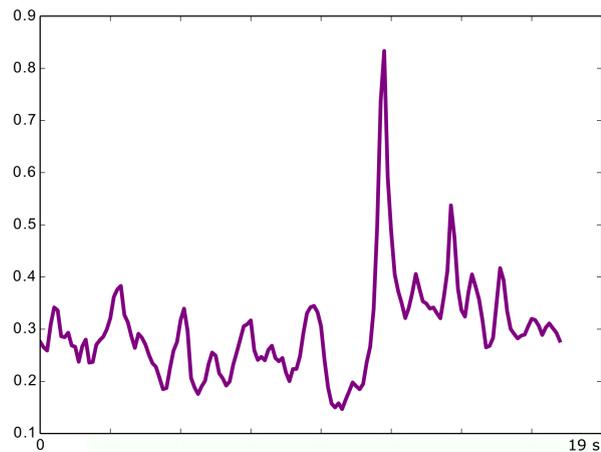


Figura 3.5: Gráfica del barrido generada por el algoritmo.

En el Anexo B se encuentra el código final comentado de Python para el sistema de monitoreo y detección de cadenas nacionales para radio y televisión, incluyendo las líneas de conmutación para activar al sistema físico que realizará mecánicamente el enlace, el cual se explica en el punto siguiente.

3.1.3. Fase III.

3.1.3.1. Integración de programa en la Raspberry Pi Modelo B.

Luego de completar el código en Python se procede a implementar dicho programa en la Raspberry Pi Modelo B. Se utilizó como entorno de trabajo el sistema operativo Raspbian. La Raspberry Pi no posee entrada A/D por lo que se conectó una tarjeta de audio USB externa Canal 7.1 compatible con controladores Linux, una vez instalada esta tarjeta se contaba con entrada de audio a la RBPi. Para la integración del código como tal se utilizó un entorno de línea de comando.

3.1.3.2. Elaboración del circuito de conmutación.

Una vez definidos los umbrales de decisión se procede a adecuar el código para enviar la orden de conmutación con la finalidad de hacer el *switch* físico del canal y realizar el enlace con la señal del estado generada por VTV. Haciendo uso de la librería RPi.GPIO, se pueden generar salidas de hasta 3.3 V en los pines GPIO (General Purpose Input/Output) con la finalidad de activar el circuito de conmutación.

Una vez definida la salida de los GPIO, se procede al diseño y cálculo de los parámetros del circuito de conmutación, se concluye el diagrama en la figura 3.6 para ser implementado:

El circuito consta de 4 relay modelo Songle de 5 V, 2 transistores 2N3904, 2 diodos 1N4148, 4 resistencias, 7 conectores para el cableado de audio y alimentación, 2 leds indicadores de estado y 2 switches para apagado total del sistema y conmutación manual. En la figura 3.7 se muestra una fotografía del circuito de conmutación conectado al GPIO 7 de la RBPI con una salida de 3.3 V bajo la orden de decisión que se produzca del algoritmo lógico.

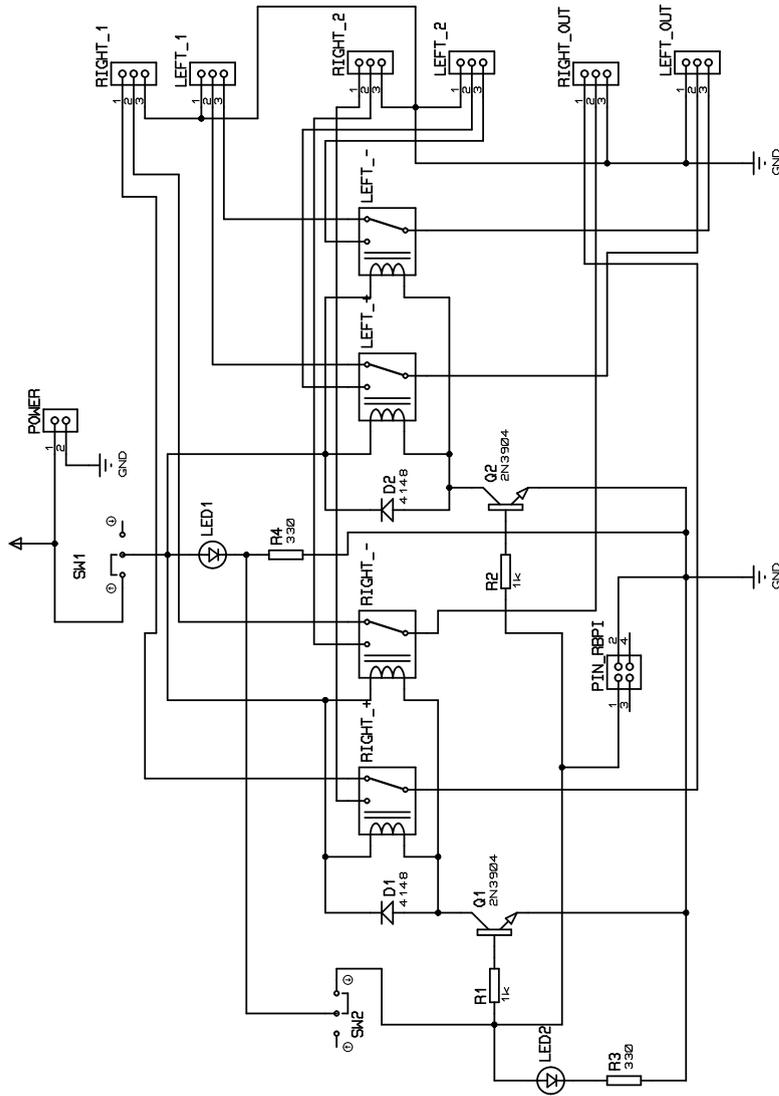


Figura 3.6: Esquemático del circuito de conmutación.

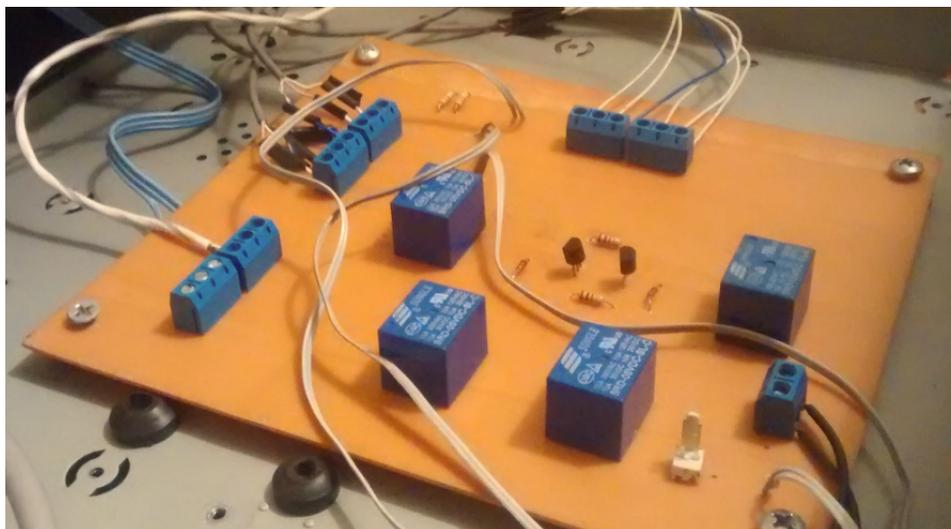


Figura 3.7: Circuito de Conmutación.

3.1.4. Fase IV.

3.1.4.1. Visitas técnicas a la sede de DIMETEL.

Con la finalidad de hacer las respectivas evaluaciones en el sitio donde será implementado el sistema de monitoreo y detección de patrones de audio se realizaron visitas a la estación FMUC. De estas visitas se definieron varios puntos y consideraciones relevantes para la implementación del proyecto. En la figura 3.8 se muestra el diagrama general de funcionamiento de la estación FMUC donde se genera la señal que se envía en la repetidora para el broadcast del contenido.

En la consola se genera el contenido que se quiere transmitir y cuenta con un canal como salida principal o señal PGM (Programación), luego en el bloque llamado "Legalizador" la señal es modulada y parametrizada según los estándares para luego enviarla al bloque STL (Studio Transmisor Link) donde la señal es llevada a niveles de Radio Frecuencia para su transmisión al medio y de esta manera llegue al repetidor.

Adicionalmente se observa el módulo de conmutación en caso de cadena nacional como medio automatizado de soporte adicional al operador. Básicamente se

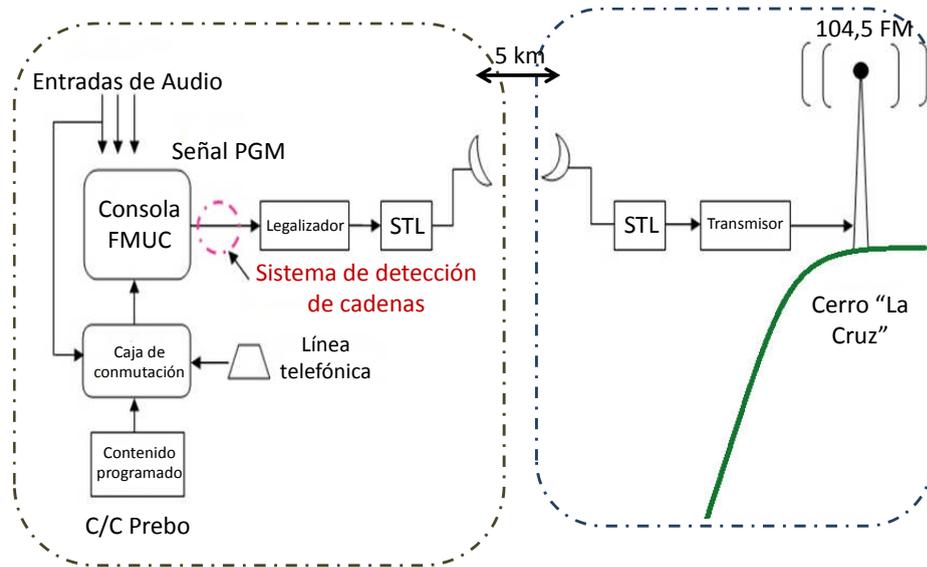


Figura 3.8: Diagrama general estación FMUC.

transmite un contenido programado o plantilla y al recibir la notificación o alerta de una cadena por vía telefónica se envía un comando a la caja de conmutación para enlazar la señal del estado.

En cuanto a la propuesta realizada del sistema de detección de cadenas nacionales y conmutación, se concluyó que la posición ideal del conmutador es en la salida principal señal PGM, ya que en las entradas de audio del mismo se tiene un radio receptor sintonizado con VTV que es el generador de la cadena nacional. De este modo, en esta posición el dispositivo puede realizar el monitoreo de la salida de audio y con su respectivo algoritmo de decisión realizar la conmutación y el enlace a la señal oficial.

Existen dos maneras de llevar señal eléctrica de audio: no-balanceada y balanceada, en este caso se utiliza audio balanceado, por lo que se trabaja con tres entradas de audio (la señal positiva, negativa y el neutro). La figura 3.9 muestra la interfaz de conmutación lógica, a manera de implementar el sistema físicamente se procedió a realizar un cajón con las entradas y salidas de audio, donde contiene el

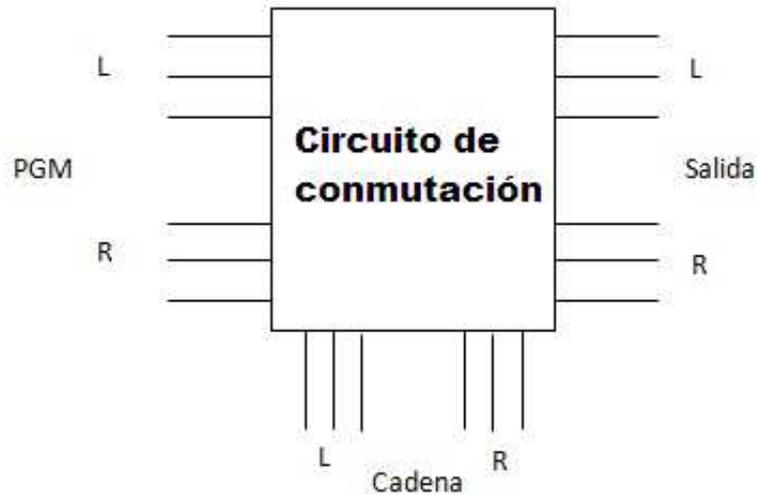


Figura 3.9: Interfaz de Conmutación Balanceada.

dispositivo Raspberry Pi realizando el monitoreo. Este cajón contiene con seis (6) conectores balanceados XLR (conocidos como Cannon), cuatro hembra (4) serán los dos (2) de entrada que comprenden la salida (Sonido estéreo L y R), dos (2) la señal de VTV a monitorear, y dos (2) conectores más XLR macho de salida. Este prototipo estará permanentemente en el cuarto de radio FMUC, en el rack de equipos de la cabina de radio.

Capítulo IV

Análisis, interpretación y presentación de los resultados

4.1. Pruebas estandarizadas *in situ* FMUC

El 8 de Diciembre de 2014 se realizó la visita a DIMETEL para realizar las pruebas del sistema de monitoreo y conmutación. Con el apoyo del Ing. Angelo Gasparini se realizaron cinco (5) pruebas diferentes para validar el funcionamiento y conocer alcance del sistema. Con la ayuda del software libre Adobe Audition CS6 se crearon diferentes escenarios: audio condiciones ideales (SNR 96 dB), coletilla con tramas desordenadas, tono añadido de 800 Hz, tono añadido de 440 Hz y ruido blanco, estas tres últimas se realizaron para perturbar el audio de las coletillas de entrada y de salida con diferentes intensidades, a fin de conocer el rendimiento del dispositivo. Estas pruebas se realizaron en la sala de reuniones de DIMETEL. En cada prueba se utilizó un archivo de audio con una duración de 19 s de coletilla de entrada y de 30 s de coletilla de salida.

4.1.1. Prueba 1: Audio de la Cadena en condiciones ideales (SNR 96 dB).

Se realizaron 30 intentos para comprobar el funcionamiento del dispositivo monitoreando la coletilla de entrada de la cadena la cual tiene una duración de 19 s, los resultados se muestran en la tabla 4.1. Donde se observa que de 30 intentos en condiciones ideales (incluyendo un SNR de 96 dB intrínseco de la tarjeta de audio), el dispositivo reconoce la cadena 29 veces, es decir tiene un porcentaje de error del 3.3 % de acuerdo con estas pruebas. De lo que se puede concluir que el sistema funciona con una baja probabilidad de error, cuando monitorea la coletilla de entrada de la cadena nacional. Con respecto al tiempo de reacción se tiene un promedio de 15.1 s para la detección. De acuerdo a estas pruebas el tiempo más corto de reacción fue de 10 s y el *peor caso* fue de 19 s, es decir justo cuando culmina el audio de la coletilla de entrada.

Para el caso de la coletilla de salida se realizan 30 intentos en condiciones ideales y se obtienen los resultados mostrados en la tabla 4.2.

De la tabla 4.2 se observa que la probabilidad de error para estas pruebas en condiciones ideales es de 0 %. Con respecto al tiempo de reacción se tiene un promedio de 17.3 s, siendo el más rápido o bien, mejor caso 12 s y el peor tiempo 25 s considerando que la coletilla de salida tiene una duración de 30 s. De lo que se concluye que el sistema de monitoreo y detección de cadenas nacionales es confiable en cuanto a la detección de ambas coletillas en condiciones de audio ideales (SNR 96 dB).

4.1.2. Prueba 2: Audio de la Cadena con cambio del orden del audio.

Para esta prueba se procedió a desordenar la coletilla de entrada de la cadena, de manera que la secuencia es prácticamente indescifrable. Se realizaron 30 intentos para comprobar el funcionamiento del dispositivo con la coletilla de entrada, los resultados se muestran en la tabla 4.3

De la tabla 4.3 se observa que de 30 intentos el dispositivo reconoce la cadena 19 veces es decir tiene un porcentaje de error del 36.6 % en las pruebas realizadas. De

Tabla 4.1: Pruebas de detección de la coletilla de entrada de la cadena con audio en condiciones ideales (SNR 96 dB).

Coletilla de Entrada	Tiempo de reacción (s)
SI	19
SI	10
SI	17
SI	19
SI	15
SI	13
SI	10
SI	14
SI	19
SI	10
SI	16
SI	19
SI	19
SI	17
SI	13
SI	17
SI	11
SI	15
SI	10
SI	19
SI	19
NO	-
SI	18
SI	14
SI	12
SI	19
SI	13
SI	10
SI	16
SI	19
Efectividad	Tiempo Prom.
96.7 %	15.1 s

lo que se puede concluir que la detección de la coletilla de entrada de la cadena en tramas desordenadas afecta la precisión del sistema. El tiempo promedio de reacción para las veces que sí detectó es de 16.6 s siendo la reacción más veloz en 10 s y la más lenta 19 s.

Tabla 4.2: Pruebas de detección de la coletilla de salida de la cadena con audio en condiciones ideales (SNR 96 dB).

Coletilla de Salida	Tiempo de reacción (s)
SI	17
SI	21
SI	15
SI	12
SI	15
SI	18
SI	15
SI	22
SI	15
SI	17
SI	21
SI	16
SI	18
SI	13
SI	20
SI	17
SI	19
SI	15
SI	12
SI	22
SI	25
SI	21
SI	17
SI	18
SI	13
SI	11
SI	15
SI	21
SI	23
SI	16
Efectividad	Tiempo Prom.
100 %	17.3 s

De igual forma se realizaron las respectivas pruebas para la coletilla de salida.

Para el caso de la coletilla de salida el sistema solo detectó 12 de las 30 pruebas que se le plantearon con el audio en tramas desordenadas, lo cual se traduce en una probabilidad de error de 60 %. Esto indica que este tipo de afectación en el audio

Tabla 4.3: Pruebas de detección de la coletilla de entrada de la cadena desordenada.

Coletilla de Entrada	Tiempo de reacción (s)
SI	18
NO	-
SI	15
SI	16
NO	-
NO	-
NO	-
SI	15
SI	16
SI	18
NO	-
SI	19
SI	15
SI	16
NO	-
SI	18
NO	-
SI	15
SI	16
SI	19
SI	18
NO	-
SI	15
SI	16
NO	-
SI	18
SI	18
NO	-
NO	-
SI	14
Efectividad	Tiempo Prom.
63.4 %	16.6 s

provoca la vulnerabilidad de efectividad del sistema considerablemente. Para el caso de la detección exitosa se tiene un tiempo promedio de respuesta de 17.5 s, siendo el más rápido 15 s y el más lento 20 s.

Tabla 4.4: Pruebas de detección de la coletilla de salida de la cadena desordenada.

Coletilla de Salida	Tiempo de reacción (s)
NO	-
NO	-
SI	15
SI	18
NO	-
SI	18
NO	-
SI	17
NO	-
NO	-
SI	15
NO	-
SI	19
NO	-
SI	20
NO	-
SI	15
SI	18
NO	-
NO	-
NO	-
SI	20
SI	17
SI	19
Efectividad	Tiempo Prom.
40 %	17.5 s

4.1.3. Prueba 3: Audio de la Cadena añadiendo tono de 800Hz.

Al realizar estas pruebas se verificó que el nivel de la señal que se generaba para monitorear era de +9dB equivalente a 60mV (valor promedio). Se generó un tono de 800 Hz a la coletilla de entrada de la cadena, se realizaron 30 intentos para

comprobar el funcionamiento del dispositivo colocando un volumen de -30 dBmV con respecto al nivel de la señal. Con referencia a a este tono, los resultados se muestran en la tabla 4.5.

Tabla 4.5: Pruebas de detección de la coletilla de entrada de la cadena añadiendo un tono de 800 Hz a -30 dBmV.

Coletilla de Entrada	Tiempo de reacción (s)
NO	-
SI	12
SI	15
SI	19
SI	20
SI	12
SI	17
SI	15
SI	19
SI	19
NO	-
SI	15
SI	12
SI	19
SI	20
NO	-
SI	12
SI	15
SI	19
NO	-
NO	-
SI	12
SI	19
NO	-
SI	13
NO	-
SI	12
SI	15
SI	19
SI	20
Efectividad	Tiempo Prom.
76.7 %	16 s

Donde se observa que de 30 intentos el dispositivo reconoce la cadena 23 veces

es decir tiene un porcentaje de error del 23.3 % según las pruebas realizadas. Con un tiempo de reacción promedio en el caso de detección de 16.0 s. El tiempo de reacción más rápido de 12 s y el peor 20 s.

Para el caso de la coetilla de salida con 800 Hz a -30 dBmV se tienen los resultados en la tabla 4.6.

En esta prueba el sistema detectó el 86.7 % de las muestras, es decir probabilidad de error de 13.3 %. Con un promedio de tiempo de reacción de 16.5 s para los casos que si realizó la detección, el mejor tiempo: 10 s y el peor: 23 s. En general se puede afirmar que esta condición aumenta la probabilidad de error del sistema, sin embargo en ningún caso (coetilla de entrada o salida) el porcentaje de error supera el 25 %. Para formalizar el alcance del sistema se realizaron pasos de aumento de volumen del tono a 800 Hz con respecto el nivel de señal de la entrada monitoreada.

Para el caso de la tabla 4.7 donde se realizan las pruebas con el tono de 800 Hz pero a -10 dBmV se observa que el sistema detectó la coetilla de entrada de la cadena el 80 % de las veces, es decir con una probabilidad de error del 20 % con un promedio de tiempo de reacción de 15.7 s para los casos de detección positiva, el tiempo más rápido de reacción es de 12 s y el más lento 19 s. Para las pruebas con la coetilla de salida se tiene la tabla 4.8.

En este caso de los 30 intentos el sistema fue capaz de detectar 22 coetillas de salida añadiendo un tono de 800 Hz a -10 dBmV de intensidad. Dando un porcentaje de error del 26.6 %. Y un tiempo de reacción promedio de 16.9 s, mejor tiempo 20 s y peor tiempo 22 s.

Se realizaron las pruebas nuevamente con el tono de 800 Hz pero en esta ocasión con una intensidad de -5 dBmV, con la finalidad de conocer hasta donde es capaz de reconocer las coetillas el sistema de detección.

Se observa que para la detección de la coetilla de entrada con el tono de 800 Hz a -5 dBmV las probabilidades de error del sistema aumentan a un 43.3 % con un tiempo de reacción promedio de 16.1 s, el tiempo más rápido de reacción es de 10 s y el más lento de 19 s. Se observa una diferencia considerable en comparación al

Tabla 4.6: Pruebas de detección de la coletilla de salida de la cadena añadiendo un tono de 800 Hz a -30 dBmV.

Coletilla de Salida	Tiempo de reacción (s)
SI	16
SI	10
SI	23
SI	15
SI	20
SI	17
SI	21
NO	-
SI	15
SI	19
SI	13
SI	10
SI	21
SI	15
SI	20
SI	16
SI	10
SI	19
NO	-
SI	20
SI	16
SI	10
SI	23
SI	17
SI	19
SI	20
SI	10
NO	-
NO	-
SI	15
Efectividad	Tiempo Prom.
86.7 %	16.5 s

porcentaje de error en la detección de las coletillas de entrada con el mismo tono de 800 Hz de -30 dBmV y -10 dBmV.

Para este caso se observa que aumentan las probabilidades de error en la decisión del sistema de monitoreo, de 30 intentos sólo reconoce el patron en 14 casos,

Tabla 4.7: Pruebas de detección de la coletilla de entrada de la cadena añadiendo un tono de 800 Hz a -10 dBmV.

Coletilla de Entrada	Tiempo de reacción (s)
SI	12
SI	15
SI	19
SI	13
SI	18
SI	19
SI	13
SI	15
NO	-
NO	-
SI	19
SI	17
SI	13
SI	15
SI	19
SI	13
NO	-
SI	19
SI	16
SI	18
SI	17
SI	15
SI	15
SI	19
SI	12
NO	-
SI	15
NO	-
SI	13
NO	-
Efectividad	Tiempo Prom.
80 %	15.7 s

disminuyendo su precisión con respecto al tono de 800 Hz en -10 dBmV y en -30 dBmV con una probabilidad de error de 53.3 % y un promedio de tiempo de reacción de 16.2 s, la detección más rápido la hizo a los 10 s y la más lenta a 20 s.

Nuevamente se realizaron pruebas añadiendo el tono de 800 Hz a la coletilla de

Tabla 4.8: Pruebas de detección de la coletilla de salida de la cadena añadiendo un tono de 800 Hz a -10 dBmV.

Coletilla de Salida	Tiempo de reacción (s)
SI	22
SI	18
SI	14
SI	15
NO	-
NO	-
SI	17
SI	15
SI	21
NO	-
SI	16
SI	18
NO	-
SI	15
NO	-
SI	19
SI	21
SI	15
SI	18
NO	-
SI	19
SI	18
SI	14
SI	15
NO	-
SI	17
SI	15
SI	20
SI	16
NO	-
Efectividad	Tiempo Prom.
73.4 %	16.9 s

entrada pero para este caso se generó con una intensidad de 0 dBmV, obteniendo los resultados mostrados en las tablas 4.11 y 4.12.

Como resultado de este escenario se obtuvo un umbral en el cual se afirma que el sistema de monitoreo no podrá detectar ninguna de las dos coletillas en estas

Tabla 4.9: Pruebas de detección de la coetilla de entrada de la cadena añadiendo un tono de 800 Hz a -5 dBmV.

Coetilla de Entrada	Tiempo de reacción (s)
NO	-
SI	19
SI	10
SI	15
NO	-
SI	19
NO	-
NO	-
SI	15
NO	-
SI	17
SI	13
SI	19
NO	-
SI	10
NO	16
NO	-
SI	19
NO	-
SI	16
NO	-
SI	15
SI	17
SI	19
SI	16
SI	19
NO	-
Efectividad	Tiempo Prom.
56.7 %	16.1 s

condiciones de audio afectado. (Tono de 800 Hz a 0 dB con respecto a la señal de audio a monitorear en +9 dB)

Tabla 4.10: Pruebas de detección de la coletilla de salida de la cadena añadiendo un tono de 800 Hz a -5 dBmV.

Coletilla de Salida	Tiempo de reacción (s)
NO	-
NO	-
SI	10
SI	15
NO	-
SI	15
NO	-
SI	19
SI	14
NO	-
NO	-
NO	-
SI	12
NO	-
SI	18
SI	19
NO	-
SI	19
SI	13
NO	-
NO	-
SI	15
NO	-
SI	19
NO	-
SI	14
NO	-
NO	-
NO	-
SI	15
Efectividad	Tiempo Prom.
46.7 %	16.2 s

4.1.4. Prueba 4: Audio de la cadena añadiendo tono de 440 Hz.

Para estas pruebas se generó un tono de 440 Hz a la coletilla de entrada de la cadena, se realizaron 30 intentos para comprobar el funcionamiento del dispositivo

Tabla 4.13: Pruebas de detección de la coletilla de entrada de la cadena añadiendo un tono de 440 Hz a -30 dBmV.

Coletilla de Entrada	Tiempo de reacción (s)
SI	15
SI	19
SI	15
SI	10
SI	19
SI	19
SI	15
SI	19
NO	-
SI	18
SI	16
SI	19
SI	15
SI	10
SI	19
SI	15
SI	19
SI	15
SI	13
NO	-
SI	15
SI	19
SI	15
SI	10
SI	18
SI	15
SI	19
SI	12
SI	18
SI	19
Efectividad	Tiempo Prom.
93.4 %	16.1 s

Para esta condición nuevamente el sistema logra detectar el 96.7 % lo que se traduce en una probabilidad de error de 3.3 %, con un promedio de tiempo de reacción de 18.1 s, la detección más rápido la hizo a los 13 s y la más lenta a los 22 s.

Tabla 4.14: Pruebas de detección de la coletilla de salida de la cadena añadiendo un tono de 440 Hz a -30 dBmV.

Coletilla de Salida	Tiempo de reacción (s)
SI	20
SI	22
SI	19
SI	15
SI	19
SI	20
SI	15
SI	16
SI	17
SI	19
SI	21
SI	16
SI	19
SI	16
SI	19
SI	18
SI	21
SI	17
SI	15
SI	19
SI	17
SI	22
SI	21
SI	15
SI	20
SI	16
SI	13
SI	17
SI	19
SI	16
Efectividad	Tiempo Prom.
96.7 %	18.1 s

Siguiendo el patrón de niveles de intensidad para el caso del tono de 800 Hz se realizaron las pruebas con diferentes niveles de intensidad del tono de 440 Hz.

Para el tono de 440 Hz a una intensidad de -10 dBmV el sistema falla en un 36.6 %. Con un tiempo promedio de reacción de 16.2 s, el caso más rápido en 15

Tabla 4.15: Pruebas de detección de la coetilla de entrada de la cadena añadiendo un tono de 440 Hz a -10 dBmV.

Coetilla de Entrada	Tiempo de reacción (s)
SI	19
NO	-
SI	19
SI	17
NO	-
NO	-
SI	17
SI	19
SI	15
SI	16
NO	-
NO	-
SI	14
SI	19
NO	-
SI	19
NO	-
SI	15
SI	16
NO	-
SI	19
NO	-
SI	15
SI	17
NO	-
SI	19
NO	-
SI	19
SI	16
SI	18
Efectividad	Tiempo Prom.
63.4 %	16.2 s

s y el más lento 19 s. Lo cual evidencia la afectación de la precisión del sistema a medida que incrementa la intensidad del tono para ambos casos.

En la tabla 4.16 se observa que el sistema tiene un porcentaje de error de 30 % con un tiempo promedio de reacción de 17.7 s, mejor tiempo: 13 s y peor tiempo: 22

Tabla 4.16: Pruebas de detección de la coletilla de salida de la cadena añadiendo un tono de 440 Hz a -10 dBmV.

Coletilla de Salida	Tiempo de reacción (s)
SI	16
NO	-
NO	-
SI	13
SI	20
SI	19
NO	-
NO	-
NO	-
SI	15
SI	13
SI	17
SI	21
SI	15
SI	20
SI	19
NO	-
SI	15
NO	-
SI	22
SI	19
NO	-
SI	16
SI	20
SI	20
NO	-
SI	16
SI	15
SI	22
SI	19
Efectividad	Tiempo Prom.
70 %	13 s

s.

Al incrementar el volumen del tono añadido a la señal monitoreada: coletilla de entrada, se evidencia una falla considerable, representado en esta prueba una probabilidad de error del 60 %. Tiempo promedio de reacción 17.6 s, detección más

Tabla 4.17: Pruebas de detección de la coletilla de entrada de la cadena añadiendo un tono de 440 Hz a -5 dBmV.

Coletilla de Entrada	Tiempo de reacción (s)
NO	-
NO	-
NO	-
SI	19
NO	-
NO	-
SI	19
SI	15
NO	-
NO	-
SI	19
NO	-
SI	15
SI	17
SI	16
NO	-
NO	-
NO	-
SI	19
NO	-
SI	17
NO	-
NO	-
SI	19
NO	-
NO	-
NO	-
SI	18
SI	19
NO	-
Efectividad	Tiempo Prom.
40 %	17.6 s

rápida en 15 s, más lenta en 19 s. Se evidencia un comportamiento parecido en la tabla 4.18 con respecto a la coletilla de salida.

En la tabla 4.18 se observa un porcentaje de error de 63.3 % detectando la coletilla de salida de la cadena 11 de 30 veces, con un tiempo promedio de 17 s, el mejor

Tabla 4.18: Pruebas de detección de la coletilla de salida de la cadena añadiendo un tono de 440 Hz a -5 dBmV.

Coletilla de Salida	Tiempo de reacción (s)
SI	23
NO	-
NO	-
SI	19
NO	-
NO	-
NO	-
SI	16
SI	18
NO	-
SI	22
NO	-
NO	-
SI	15
NO	-
SI	18
NO	-
SI	19
NO	-
SI	21
NO	-
SI	19
SI	16
NO	-
Efectividad	Tiempo Prom.
36.7 %	17 s

tiempo fue de 16 s y el peor de 23 s.

Para el caso de las coletillas de entrada y salida añadiendo un tono de 440 Hz con una intensidad de 0 dBmV se evidencia que el sistema no es capaz de reconocer el patrón para ninguno de los casos, como se observa en las tablas 4.19 y 4.20.

de frecuencias [15]. Esta afectación es bastante común para señales de broadcast como es el caso de la radio. Para esta prueba se estandarizaron diferentes valores de intensidad de ruido.

A la coetilla de entrada de 19 s se le añadió ruido blanco con SNR de 39 dB, obteniendo los resultados mostrados en la tabla 4.21.

Se observa una ligera alteración en la precisión del sistema para estos niveles de SNR añadiendo ruido blanco en la coetilla de entrada, con un tiempo de reacción promedio de 16.6 s y un porcentaje de error de 6.7 %, el mejor tiempo fue de 13 s y el peor de 19 s.

Para el caso de la coetilla de salida con SNR de 39 dB la probabilidad de error es de 6.7 %, para el caso que si identificó la coetilla de salida se tiene un tiempo promedio de 17.3 s, con el mejor tiempo de respuesta en 13 s y el peor 23 s.

Para el caso de ruido blanco con SNR de 19 dB, se tienen las tablas de resultados 4.23 y 4.24

Para ambos casos se observa que la confiabilidad del sistema de detección añadiendo ruido blanco con SNR de 19 dB se ve comprometida. Para el caso de la coetilla de entrada se tiene un porcentaje de error de 3.3 % con un tiempo promedio de 16 s, mejor tiempo: 10 s y peor tiempo: 19 s; para el caso de la coetilla de salida se obtiene un porcentaje de error de 13.3 % con un promedio de tiempo de detección de 18.6 s, el tiempo más rápido de detección fue de 15 s y el más lento 23 s. Con la finalidad de confirmar el alcance del sistema, se realizaron pruebas con mayores intensidades, hasta conseguir un umbral de funcionamiento del sistema. Para el siguiente caso se realizan las pruebas con un SNR de 14 dB.

En la tabla 4.25 se observa la degradación de la efectividad del sistema, ya que arroja una probabilidad de error del 26.7 % para las pruebas realizadas. Con un tiempo promedio para el caso de detección de 17.6 s, el mejor tiempo de respuesta fue de 15 s y el peor 19 s.

Tabla 4.21: Pruebas de detección de la coletilla de entrada de la cadena añadiendo ruido blanco SNR 39 dB.

Coletilla de Entrada	Tiempo de reacción (s)
SI	13
SI	17
SI	16
SI	19
SI	15
NO	-
SI	19
SI	16
SI	17
SI	18
SI	15
SI	17
SI	16
SI	19
SI	15
SI	15
SI	17
SI	19
SI	15
SI	16
SI	14
SI	19
SI	15
SI	19
SI	19
SI	17
NO	-
SI	13
SI	17
SI	19
Efectividad	Tiempo Prom.
93.3 %	16.6 s

De igual forma que con la coletilla de entrada, se genera un porcentaje de error para la coletilla de salida con SNR de 14 dB: 26.7 %, con un tiempo promedio de detección de 18.4 s, el tiempo más rápido de detección: 13 s y el más lento: 19 s.

Tabla 4.22: Pruebas de detección de la coletilla de salida de la cadena añadiendo ruido blanco SNR 39 dB.

Coletilla de Salida	Tiempo de reacción (s)
SI	23
SI	19
SI	16
SI	12
SI	16
SI	19
SI	17
SI	13
SI	19
SI	22
SI	15
SI	23
SI	19
SI	15
SI	17
SI	21
NO	-
NO	-
SI	19
SI	16
SI	13
SI	19
SI	17
SI	15
SI	12
NO	-
SI	19
SI	18
SI	17
SI	18
Efectividad	Tiempo Prom.
93.3 %	17.3 s

Se realizaron 30 pruebas con la coletilla de entrada y 30 pruebas con la coletilla de salida con una relación señal a ruido SNR de 9 dB.

Para el caso de SNR de 9 dB se observa un comportamiento bastante deficiente en la efectividad del sistema, sin embargo se resalta el hecho de que una señal de

Tabla 4.23: Pruebas de detección de la coletilla de entrada de la cadena añadiendo ruido blanco SNR 19 dB.

Coletilla de Entrada	Tiempo de reacción (s)
SI	15
SI	19
SI	19
NO	-
SI	10
SI	19
SI	15
SI	19
SI	17
SI	19
SI	13
SI	15
SI	17
SI	11
SI	17
SI	19
SI	19
SI	19
SI	17
SI	17
SI	18
SI	17
SI	15
SI	17
SI	17
SI	18
SI	14
SI	16
SI	13
SI	17
Efectividad	Tiempo Prom.
96.7 %	16 s

audio con ruido blanco con esta relación señal a ruido representa una señal bastante degradada en calidad y legibilidad y sin embargo el sistema de detección y conmutación logra reconocer ambos patrones (entrada y salida) en al menos 9 ocasiones para cada caso. Para la coletilla de entrada presenta un error del 63 % y para

Tabla 4.24: Pruebas de detección de la coletilla de salida de la cadena añadiendo ruido blanco SNR 19 dB.

Coletilla de Salida	Tiempo de reacción (s)
SI	16
NO	-
SI	23
SI	16
SI	20
SI	21
SI	18
SI	19
SI	17
NO	-
NO	-
SI	16
SI	21
SI	17
SI	19
SI	19
SI	21
SI	21
SI	16
SI	15
NO	-
SI	22
SI	16
SI	15
SI	17
SI	23
SI	19
SI	17
SI	15
SI	19
Efectividad	Tiempo Prom.
86.7 %	18.6 s

la coletilla de salida 70 %. Tiempo promedio de detección de entrada y salida 17.4 s y 18 s respectivamente. Lo cual refleja la robustez del algoritmo de comparación de patrones, mediante la correlación y comparación de espectrogramas, aún cuando la SNR afecta fuertemente la calidad de la señal a monitorear.

Tabla 4.25: Pruebas de detección de la coletilla de entrada de la cadena añadiendo ruido blanco SNR de 14 dB.

Coletilla de Entrada	Tiempo de reacción (s)
NO	-
NO	-
SI	19
NO	-
SI	17
SI	19
SI	15
NO	-
SI	16
SI	19
SI	17
SI	19
SI	17
NO	-
SI	18
SI	16
NO	-
SI	19
SI	19
SI	17
NO	-
SI	17
SI	19
SI	17
SI	18
SI	15
SI	19
SI	17
SI	16
NO	-
Efectividad	Tiempo Prom.
73.3 %	17.6 s

Para finalizar estas pruebas se elevó la intensidad del ruido blanco con SNR igual a 4 dB para ambos casos, obteniendo los resultados mostrados en las tablas 4.29 y 4.30.

Como conclusión para estas pruebas se observa que a partir de SNR de 4 dB,

Tabla 4.26: Pruebas de detección de la coletilla de salida de la cadena añadiendo ruido blanco SNR de 14 dB.

Coletilla de Salida	Tiempo de reacción (s)
NO	-
SI	21
NO	-
SI	19
NO	-
SI	15
NO	-
NO	-
SI	19
SI	17
NO	-
SI	22
NO	-
SI	19
SI	18
NO	-
SI	15
NO	-
SI	15
SI	19
NO	-
SI	19
SI	16
SI	17
SI	13
SI	17
SI	19
NO	-
SI	19
SI	14
Efectividad	Tiempo Prom.
73.3 %	18.4 s

el sistema no detecta la coletilla ni de entrada ni de salida, siendo entonces, la presencia de ruido blanco con este nivel de SNR una limitante para el correcto funcionamiento del dispositivo. Cabe destacar que SNR igual a 4 dB de ruido blanco añadido a una señal genera una fuerte degradación de calidad.

Tabla 4.27: Pruebas de detección de la coetilla de entrada de la cadena añadiendo ruido blanco SNR de 9 dB.

Coetilla de Entrada	Tiempo de reacción (s)
NO	-
NO	-
NO	-
SI	19
SI	19
NO	-
NO	-
NO	-
SI	19
SI	17
NO	-
NO	-
NO	-
SI	19
NO	-
SI	19
NO	-
NO	-
SI	19
SI	13
NO	-
NO	-
NO	-
SI	16
NO	-
SI	17
NO	-
SI	15
NO	-
NO	-
Efectividad	Tiempo Prom.
37 %	17.4 s

En los casos en que se detectó la cadena la conmutación se realizó el 100 % de las veces. Con la finalidad de generalizar los resultados obtenidos en las pruebas, se presenta a continuación una tabla resumen de resultados sobre el desempeño y alcance del sistema de monitoreo y detección de cadenas nacionales. La tabla 4.31

Tabla 4.28: Pruebas de detección de la coletilla de salida de la cadena añadiendo ruido blanco SNR de 9 dB.

Coletilla de Salida	Tiempo de reacción (s)
NO	-
NO	-
NO	-
SI	23
NO	-
SI	16
SI	19
NO	-
SI	15
NO	-
NO	-
SI	14
NO	-
NO	-
NO	-
SI	16
NO	-
NO	-
SI	19
NO	-
SI	19
SI	21
Efectividad	Tiempo Prom.
30 %	18 s

muestra un resumen de los resultados de efectividad del sistema para las diferentes pruebas en el monitoreo de la coletilla de entrada de las cadenas nacionales.

Tabla 4.31: Resumen de desempeño del sistema de monitoreo de cadenas nacionales, coletilla de entrada.

ESCENARIO	Condiciones Ideales	Tramas Desordenadas	Tono 800Hz	Tono 440Hz	Ruido Blanco
NIVEL TONO - SNR/EFFECTIVIDAD	96.7 %	63.4 %	-	-	-
-30 dBmV/39 dB	-	-	76.7 %	93.4 %	93.3 %
-10 dBmV/19 dB	-	-	80.0 %	63.4 %	96.7 %
-5 dBmV/14 dB	-	-	56.7 %	40.0 %	73.4 %
0 dBmV/9 dB	-	-	-	-	37.0 %
5 dBmV/4 dB	-	-	-	-	-

Tabla 4.32: Resumen de desempeño del sistema de monitoreo de cadenas nacionales, coletilla de salida.

ESCENARIO	Condiciones Ideales	Tramas Desordenadas	Tono 800Hz	Tono 440Hz	Ruido Blanco
NIVEL TONO - SNR/EFFECTI-VIDAD	100 %	40 %	-	-	-
-30 dBmV/39 dB	-	-	86.7 %	96.7 %	93.3 %
-10 dBmV/19 dB	-	-	73.4 %	70 %	86.7 %
-5 dBmV/14 dB	-	-	53.4 %	36.7 %	73.4 %
0 dBmV/9 dB	-	-	-	-	30.0 %
5 dBmV/4 dB	-	-	-	-	-

puntualizar estas condiciones y de esta manera se define el alcance global del *performance* del sistema.

4.1.6. Definir umbral mínimo de nivel de detección.

Adicionalmente se realizaron pruebas en condiciones de audio normales pero variando las intensidades de la señal de entrada para definir el umbral de detección del sistema, el cual resultó ser de -5dB referente del software lo que representa 20 mV (valor promedio) en la entrada del sistema de monitoreo. Es importante tomar en cuenta esta consideración para asegurar la confiabilidad.

De acuerdo a los valores promediados del tiempo de reacción de detección de ambas coletillas se puede observar que en términos generales el tiempo de reacción para la identificación de la coletilla de salida es mayor que el tiempo de reconocimiento de las coletillas de entrada. Sin embargo no se observa un patrón fijo de los tiempos de detección. Esto representa una ganancia cuando se compara con el sistema que actualmente utiliza FMUC cuando no se encuentra un operador en las instalaciones, debido a que deben llamar a una consola para activar la conmutación remotamente y esto puede tardar entre 5-15 min, con el sistema automatizado que se diseñó se reducen considerablemente estos tiempos ya que no pasa de los 30 s la ejecución de la conmutación.

4.2. Dispositivo final de monitoreo y conmutación.

Para facilitar la implementación del sistema en FMUC se programo la RBPi para que inicie el programa de monitoreo *por defecto* es decir, apenas se energice el sistema comienza a correr el loop infinito para la detección de cadenas de radio y televisión. Una vez iniciado el programa se configuro la RBPi para que envíe un correo electrónico a una dirección determinada con su dirección IP. [16] Se generó una función en el algoritmo de manera que cada vez que detecte una cadena y conmute envíe un mensaje por correo electrónico notificando la presencia de la cadena de tal forma de conocer la hora y fecha en que realizó la detección y conmutación,

igualmente para el proceso de salida de la cadena. En caso de que por alguna razón el dispositivo no haya detectado una cadena y realizado la conmutación, o caso contrario, una vez terminada la cadena el sistema se haya quedado inhibido y siga transmitiendo la señal del estado se incluyó en el programa, mediante un código PHP y el servidor Apache en la RBPi [17], una página de control remoto al sistema de conmutación solicitando autenticación del usuario para evitar sabotajes externo a la red y al sistema de conmutación. Con ayuda del coordinador del departamento Rediuc Rubén Cheng, se configuró la entrada remota a la red local para acceder a la RBPi. La interfaz de la página de control remoto se observa en la figura 4.11



Figura 4.1: Interfaz página web control remoto Programación FMUC.

Una vez realizadas las visitas a FMUC y pruebas del algoritmo, se construyó el dispositivo completo, incluyendo el sistema de monitoreo, el circuito de conmutación, la fuente de alimentación dentro del cajón con sus respectivas entradas y salidas; incluyendo dos leds de estado para indicar la señal que se está transmitiendo, se incluyó un switch de apagado del sistema y un switch para realizar la conmutación manualmente. Adicionalmente se realizó un manual de usuario acerca del sistema de monitoreo y detección de cadenas nacionales de radio y televisión para los operadores de FMUC, de manera que puedan realizar cambios en la base de datos (señal para comparar) en el caso de que la coletilla de la cadena originada por VTV sea modificada.

CONTROL REMOTO DE LA CONMUTACION FMUC DE LA RASPBERRY PI



Figura 4.2: Interfaz página web control remoto Programación FMUC.

En el anexo C se encuentra el manual de usuario para el sistema de monitoreo y detección de cadenas nacionales para radio y televisión.

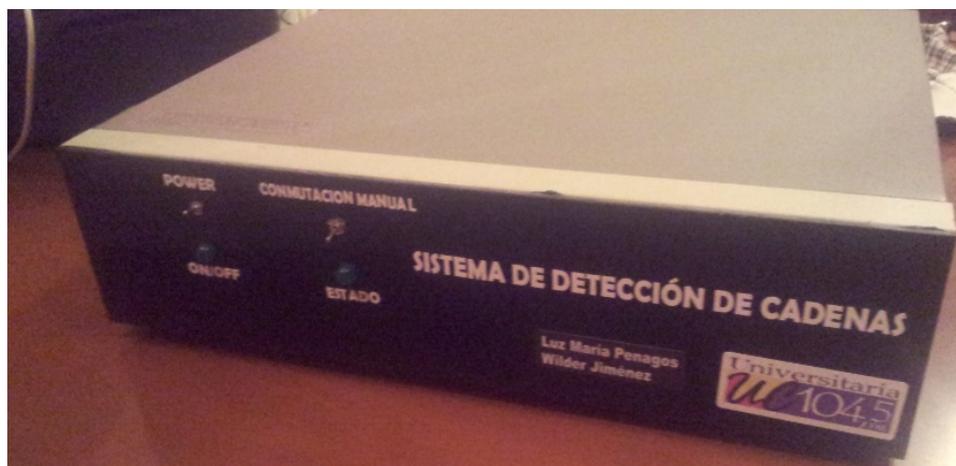


Figura 4.3: Sistema final (hardware) a implementar de detección y conmutación de Cadenas Nacionales.

Capítulo V

Conclusiones y recomendaciones

5.1. Conclusiones

Se analizaron los parámetros generales de señales de audio y las consideraciones que se deben tener para realizar una comparación de patrones. Se tomaron en cuenta recomendaciones y documentación relacionada al procesamiento de señales, considerando que cada señal en el dominio del tiempo posee un patrón único en el dominio de la frecuencia y puede ser evidenciado con un espectrograma. Existen variables de las señales de audio que son necesarias considerar para realizar un procesamiento de la misma tal como el formato, los canales y el muestreo de la señal. También es importante tomar en cuenta la normalización y estandarización de las señales a procesar.

Se diseñó el algoritmo para la identificación de las señales de audio de cadenas nacionales (coletillas de entrada y salida). Para esto se utilizó el método de la correlación cruzada en varias etapas para asegurar el resultado, seleccionando umbrales ideales y complementando con una etapa de comparación de espectrogramas la cual aporta mayor confiabilidad y precisión a dicho algoritmo. Este algoritmo se utilizó gracias a las herramientas disponibles para el procesamiento de señales de audio derivadas del punto anterior. Para este diseño se utilizó el lenguaje Python

debido a las ventajas que representa y al dispositivo seleccionado para la implementación de este algoritmo: Raspberry Pi Modelo B.

Se implementó con éxito el programa generado en Python en el dispositivo basado en DSP Raspberry Pi Modelo B, el cual trabaja con Raspbian derivado de Linux y ofrece gran compatibilidad con el lenguaje utilizado. Fue necesaria la adecuación de la RBPi intengrandole una tarjeta de audio que permitiera la entrada de señales para su procesamiento en el algoritmo diseñado. Se realizaron pruebas de desempeño del sistema mostrando resultados esperados con alta confiabilidad.

Se implementó una interfaz de hardware la cual comprende de la tarjeta Raspberry Pi (encargada del cómputo de procesamiento), un circuito de conmutación físico con sus respectivas entradas y salidas de conectores de audio XLR para su instalación en las oficinas de FMUC, con funciones extras como sistema de conmutación manual en caso de fallas del sistema lógico y acceso remoto al circuito de conmutación.

De acuerdo a las pruebas realizadas al sistema, se observa un alto rendimiento en el *performance*, mostrando un 96.7 % de efectividad para la detección de la coletilla de entrada y un 100 % de efectividad para la coletilla de salida en condiciones ideales (SNR 96 dB), además fue capaz de detectar ambas coletillas con presencia de ruido blanco a una intensidad de 9 dB. Se obtuvieron rápidos tiempos de respuesta en la detección de las coletillas de entrada (promedio de 15.1 s) y salida (promedio de 17.3 s) de las cadenas nacionales en condiciones ideales. Se evidenció un *feedback* positivo de parte del personal de FMUC al observar el comportamiento del sistema.

Se alcanzaron en su totalidad los objetivos planteados, identificando en el camino desempeño, alcance y posibles mejoras en el sistema.

5.2. Recomendaciones

Se recomienda extender o fortalecer el alcance del sistema diseñado complementandolo con un sistema de detección de vídeo para cadenas de radio y televisión.

Se recomienda la verificación de las condiciones ideales controlables por personal, como el nivel de entrada de señal y las posibles obstrucciones o modificaciones para la detección del sistema, de esta forma se garantiza con mayor seguridad el correcto desempeño del sistema.

Se recomienda mantener activos los procesos actuales de conmutación establecidos en FMUC, de manera de tener redundancia en la identificación y conmutación de cadenas de radio y televisión.

Se recomienda el estudio del manual de usuario para aclarar dudas y aplicar la correcta manipulación del sistema diseñado e implementado.

Se recomienda extender el estudio de los algoritmos de comparaciones de patrones de audio, a fin de comparar con resultados las ventajas y desventajas de otros sistemas de detección, utilizando el mismo hardware en este caso Raspberry Pi Modelo B.

Referencias Bibliográficas

- [1] Ley Responsabilidad Social en Radio y Televisión. Gaceta oficial n°39610. Caracas, Diciembre 2010. Art. 10.
- [2] Ley Orgánica de Telecomunicaciones. Gaceta oficial n°36970. Caracas, Junio 2000. Art. 37, Art. 166, Art. 192.
- [3] 5 Emisora Universitaria FMUC 104, 2014. URL <http://www.dimetel.uc.edu.ve/departamentos/fmuc.html>.
- [4] J. Camargo. *Sistema de Reconocimiento de voz humana por Hardware*. Bucaramanga. Universidad Pontifica Bolivariana, 2010.
- [5] Reconocimiento de la voz. Breve historia de las redes neuronales, 2014. URL <http://www.redcientifica.com/doc/doc199908210001.html>.
- [6] J. Faundez. *Teoría de la Comunicación*. Universidad de Valladolid, 2001.
- [7] H. Pedroza. *Sistema de Análisis Estadístico con SPSS*. Nicaragua, 2007.
- [8] A. Moreno. *Correlación y espectro de señales deterministas*. Universidad Politécnica de Cataluña. España, 2009.
- [9] Análisis de Audio usando algoritmos de Procesamiento de Imágenes, 2014. URL <http://jack.minardi.org/software/computational-synesthesia>.
- [10] Fundación Raspberry Pi, 2014. URL <http://www.raspberrypi.org>.
- [11] M. Bejarano. Development of a compact patient ventilator base on novel compressor technology. Maestría en Ingeniería Clínica. Escuela de Ingeniería y Ciencias Matemáticas, Facultad de Ingeniería, Universidad de Londres, 2012.

-
- [12] J. Merrick. Using a usb audio device with a raspberry pi (7.1 channel usb external sound card audio adapter). Technical report, 2013. URL <http://computers.tutsplus.com/articles/using-a-usb-audio-device-with-a-raspberry-pi--mac-55876>.
- [13] Raspbian Operating System Organization, 2014. URL <http://www.raspbian.org/>.
- [14] Fundación de Software Python. Acerca de Python, 2014. URL <http://www.python.org/about>.
- [15] A. Bateman. *Comunicaciones Digitales*. España, 2003.
- [16] J. Rivera. Desarrollo de un sistema domótico básico usando rbpi, php y python. Technical report, 2013. URL http://jeffersonrivera.com/pi/Proyecto_raspberry_pi.pdf.
- [17] A. Rodrigues. Activación de alarmas remotas mediante wifi entre minicomputadoras rbpi en aplicaciones de seguridad. Facultad de Ingeniería en Electricidad y Computación. Guayaquil. Ecuador, 2013.

Anexo A

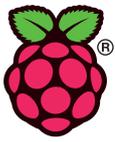
**Especificaciones de la estructura de
la Raspberry Pi Modelo B**

Anexo B

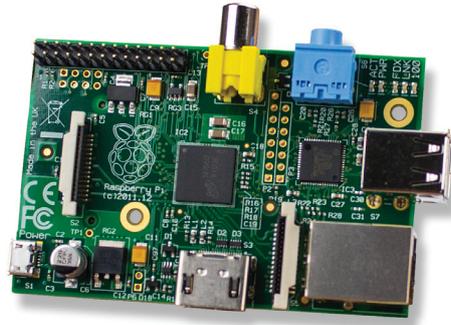
**Código principal del sistema de
detección de patrones de audio
para la identificación de cadenas
nacionales de radio y televisión**

Anexo C

Manual de usuario para el sistema de monitoreo y detección de cadenas nacionales de radio y televisión



Raspberry Pi



MODEL B

Product Name Raspberry Pi Model B

Product Description The Raspberry Pi is a small, powerful and lightweight ARM based computer which can do many of the things a desktop PC can do. The powerful graphics capabilities and HDMI video output make it ideal for multimedia applications such as media centres and narrowcasting solutions. The Raspberry Pi is based on a Broadcom BCM2835 chip. It does not feature a built-in hard disk or solid-state drive, instead relying on an SD card for booting and long-term storage.

RS Part Number 756-8308

Specifications

Chip Broadcom BCM2835 SoC (a)

Core architecture ARM11

CPU 700 MHz Low Power ARM1176JZFS Applications Processor

GPU Dual Core VideoCore IV® Multimedia Co-Processor
Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode
Capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure

Memory 512MB SDRAM

Operating System Boots from SD card, running a version of the Linux operating system

Dimensions 85.6 x 53.98 x 17mm

Power Micro USB socket 5V, 1.2A (l)

Connectors:

Ethernet 10/100 BaseT Ethernet socket (b)

Video Output HDMI (rev 1.3 & 1.4) (c); Composite RCA (PAL and NTSC) (d)

Audio Output 3.5mm jack (e), HDMI

USB 2.0 Dual USB Connector (f)

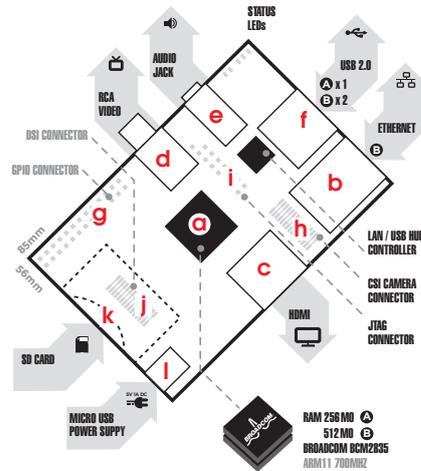
GPIO Connector 26-pin 2.54 mm (100 mil) expansion header: 2x13 strip. Providing 8 GPIO pins plus access to I²C, SPI and UART as well as +3.3 V, +5 V and GND supply lines (g)

Camera Connector 15-pin MIPI Camera Serial Interface (CSI-2) (h)

JTAG Not populated (i)

Display Connector Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane (j)

Memory Card Slot SDIO (k)



Accessories



▲ Camera Module
775-7731



▲ International power supply
765-3311



▲ 8GB SD card pre-programmed with NOOBS - 779-6770



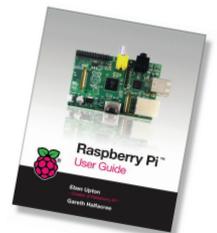
▲ Expansion board
772-2974



▲ WiFi dongle
760-3621



▲ 10400mAh Li-Ion battery pack
775-7517



▲ Raspberry Pi user guide
768-6686



```

import pyaudio

import time

import numpy as np

from scipy.fftpack import fft, fftfreq

from skimage.feature import match_template    ###Librerias utilizadas

import matplotlib.pyplot as plt

import RPi.GPIO as GPIO

###Parametros y variables

CHUNK = 1024    #Sample de 20 ms

CHUNK_1SEC = 44100    #Sample de 1 s

FORMAT = pyaudio.paInt16

RATE = 44100    #Tasa de muestreo

CHAN = 1    #Mono

GPIO.setmode(GPIO.BOARD) #Configuracion de GPIOs

GPIO.setwarnings(False)

###Funcion para convertir data en espectrograma

def spectrogram(data, segment_size=60):

    end = len(data) - len(data) % segment_size

    stacked = data[:end].reshape((-1, segment_size))

    freq_space = np.fft.fft(stacked)

    real = np.abs(freq_space)

    trimmed = real.T[:segment_size/2, :]

    return trimmed

```

```
np.set_printoptions(threshold='nan')  
sound=np.fromfile(open(r'/home/pi/IntroCadena.wav'),np.int16) #Archivo de Coletilla de entrada  
sound = sound[0:856312]  
downsampled = sound.reshape((-1,4)).mean(1)  
spec_in = spectrogram(downsampled, segment_size=512)
```

```
sound_out=np.fromfile(open(r'/home/pi/OutCadena.wav'),np.int16) #Archivo de Coletilla de salida  
sound_out = sound_out[0:856312]  
downsampled_out = sound_out.reshape((-1,4)).mean(1)  
spec_out = spectrogram(downsampled_out, segment_size=512)
```

#Extraccion de FFT de tamaño de ventana de 20ms (1024 muestras)

```
snd1 = sound[617400:617400+CHUNK]  
snd1 = snd1 / (2.**15)  
window1 = np.hamming(CHUNK)  
snd1 = snd1*window1  
n1 = len(snd1)  
Y1 = fft(snd1) / n1  
Y1 = abs(Y1)  
Y1 = Y1.tolist()
```

#Procesamiento de data (Streaming)

```
pa = pyaudio.PyAudio()  
stream = pa.open(format = FORMAT,
```

```
channels = CHAN,  
rate = RATE,  
input = True,  
frames_per_buffer = CHUNK)
```

#Variable buffer para almacenamiento de resultados de correlacion

```
buff = [np.zeros(1024, dtype=np.int16),np.zeros(1024, dtype=np.int16),np.zeros(1024,  
dtype=np.int16),np.zeros(1024, dtype=np.int16)]
```

```
spec = spec_in
```

```
while stream.is_active(): #Ciclo infinito de procesamiento continuo de audio
```

```
try:
```

```
data = stream.read(CHUNK) #Extraccion de muestras de 20 ms
```

```
data_FUN = stream.read(CHUNK_1SEC) #Extraccion de muestras de 1 s
```

```
snd = np.fromstring(data, dtype=np.int16) #Parametrizacion de la muestra
```

```
snd = snd / (2.**15) #Normalización de la muestra
```

```
window = np.hamming(CHUNK) #Ventana de hamming
```

```
snd = snd*window
```

```
n = len(snd)
```

```
Y = fft(snd) / n #FFT y valor absoluto
```

```
Y = abs(Y)
```

```
Y = Y.tolist()
```

```
buff.append(Y) #Almacenamiento de variable buffer
```

```
buff.pop(0)
```

```
np.seterr(divide='ignore',invalid='ignore')
```

```

correlation1 = np.corrcoef(Y, Y1)      #Correlacion cruzada
correlation2 = np.corrcoef(buff[2], Y1)
correlation3 = np.corrcoef(buff[1], Y1)
print correlation1[0,1]

#Umbral definido en 0.7 para posible candidato de cadena

if correlation1[0,1] > 0.7 or correlation2[0,1] > 0.7 or correlation3[0,1] > 0.7:
    print 'Verificando CADENA'

#Etapas de verificación mediante barrido de espectrogramas

sample = np.fromstring(data_FUN, dtype=np.int16)
downsampled_sample = sample.reshape((-1,4)).mean(1)
sample_spec = spectrogram(downsampled_sample, segment_size=512)
result = match_template(spec, sample_spec)
print result.max()

#Umbral definido para decisión de conmutación

if result.max() > 0.7:
    print '....Verificación comprobada.... CONMUTANDO!!!'
    GPIO.setup(7, GPIO.OUT) #Orden de conmutación
    if GPIO.input(7) == 0:
        print 'Entrando a Cadena Nacional de Radio y Televisión...'
        GPIO.output(7, not GPIO.input(7))
        spec = spec_out #Cambio de coetillas
        time.sleep(30) #break de 30 segundos
        execfile('/home/pi/Date_time.py')
    else:

```

```
print 'Saliendo de Cadena Nacional de Radio y Television...'  
  
GPIO.output(7, not GPIO.input(7))  
  
spec = spec_in  
  
time.sleep(30)  
  
execfile('/home/pi/Date_timeOut.py')  
  
continue  
  
else:  
  
continue  
  
  
stream.close()  
  
pa.terminate()
```

Manual de usuario para el sistema de detección de cadenas nacionales de radio y televisión

Wilder Jimenez y Luz María Penagos

December 12, 2014

1 Bienvenido

Bienvenido al manual de usuario para el sistema de monitoreo y detección de cadenas nacionales de radio y televisión diseñado para FMUC. Este manual le será de utilidad para el correcto uso del equipo así como también le será útil para la modificación (de ser necesario) del código de detección. Se recomienda leerlo en su totalidad.

2 Especificaciones Físicas

El sistema de monitoreo de cadenas nacionales está compuesto por dos partes fundamentales:

2.1 Raspberry Pi

Dispositivo electrónico basado en DSP (microcomputador Linux) el cual realiza el procesamiento del código diseñado para hacer los cálculos de comparación de patrones y tomar la decisión de conmutar.

2.2 Circuito de Conmutación

Circuito de conmutación diseñado para realizar el enlace con la señal del estado una vez que la Raspberry Pi detecte una cadena y “de la orden”, es decir envía una señal al circuito el cual activa a los relay y se transmite la señal con la cadena nacional.

A continuación se muestra el esquema para el panel frontal y el panel trasero:

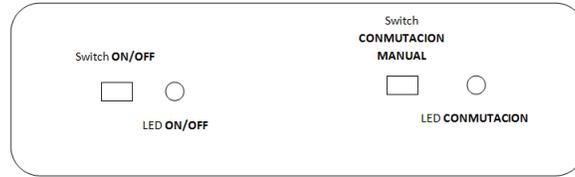


Figure 1: Panel Frontal

El panel frontal como se observa en la Figura 1 consta basicamente por dos leds y dos switches. El switch principal de encendido y apagado del sistema y su led correspondiente que indica el estado ON/OFF del sistema. El segundo switch es para la conmutación manual de salida, en caso de que por alguna razón el sistema se inhiba y no logré activar el circuito de conmutación y el led que indica si el sistema está transmitiendo FMUC o la señal del estado.

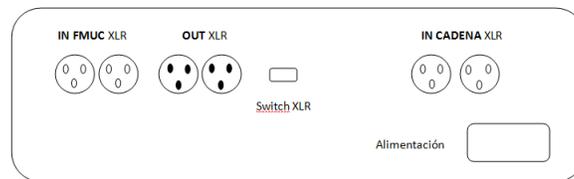


Figure 2: Panel Trasero

El panel trasero el equipo cuenta con dos entradas de audio balanceado de conectores XLR, en una de las entradas se tiene la programación de FMUC y por la otra entrada la señal a monitorear. En la salida se obtiene la señal seleccionada dependiendo del circuito de conmutación. Se tiene un switch que cambia la entrada del conector que llega a la Raspberry Pi y la entrada de alimentación del sistema.

3 Código de Procesamiento

Como se mencionó anteriormente el código que realiza la comparación de patrones está almacenado en la memoria de la Raspberry Pi, a continuación se comentan los pasos a seguir en caso de que la coetilla de entrada/salida sea modificada. Cuando esto ocurra solo se debe acceder a la carpeta de almacenamiento y modificar el archivo .wav con el cual la Raspberry Pi realiza la comparación de patrones.

3.1 Acceso a la RBPi mediante SSH

Para acceder a la RBPi debemos estar conectados a nuestra red local mediante un cable UTP y el puerto Ethernet de la RasPi. Es necesario instalar el cliente SSH Putty (para Windows) para tener acceso al prompt de la RBPi, una vez instalado Putty e iniciarlo se obtiene la siguiente pestaña:

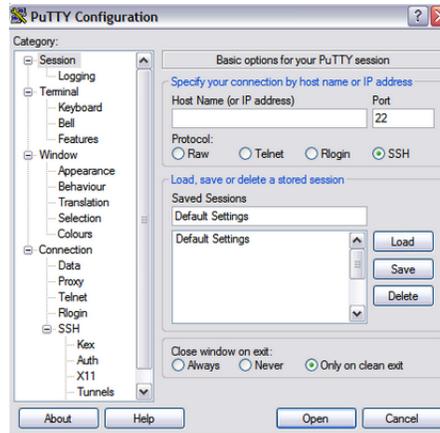


Figure 3: Configuración Putty

Donde se introduce la dirección IP que nuestra red le ha asignado a la RBPi, existe la opción de guardar una sesión la cual conserva la dirección IP que se le asignó a la RBPi, esto es de utilidad si es una dirección IP fija. Luego seleccione la opción “OPEN” y si la conexión es exitosa deberá aparecer lo siguiente:

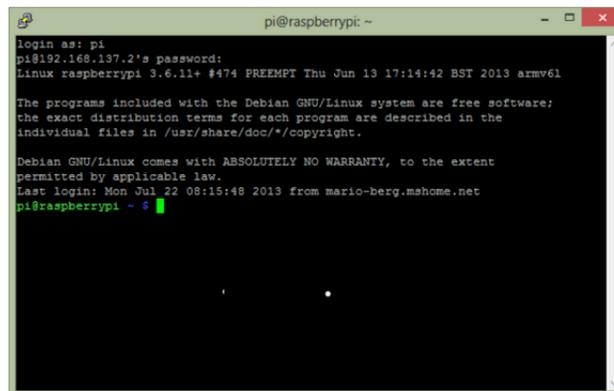


Figure 4: Conexión SSH a la RBPi con Putty

3.2 Cambio de archivo .wav

Del paso anterior hemos accedido a la RBPi mediante SSH, para el siguiente paso es necesario utilizar la herramienta VNC Server (la cual puede ser descargada desde internet) y en el prompt solo se debe introducir: “vncserver:1” donde se solicita usuario y contraseña para configurar la sesión, una vez realizado esto se tiene acceso a la interfaz gráfica de la Raspberry Pi y debe lucir de la siguiente manera:

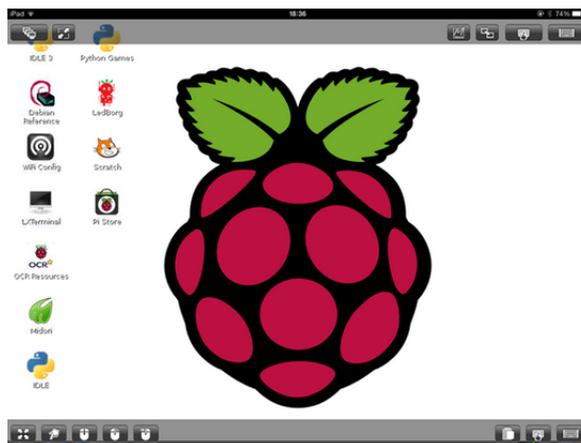


Figure 5: Interfaz desde VNC: Raspberry Pi

Una vez aquí se debe acceder a la carpeta “/home/pi” donde se encuentra almacenado el archivo .wav con el cual el sistema compara las muestras tomadas en stream. Para el código actual se tiene para la coetilla de entrada “IntroCadena.wav” y la coetilla de salida “OutCadena.wav”.

Se recomienda que en caso de existir una modificación oficial de alguna de estas coetillas, sólo es necesario tomar el nuevo archivo base para comparar y reemplazarlo por el archivo existente, incluyendo el mismo nombre.

4 Contáctenos

Para cualquier duda o soporte avanzado sobre el sistema de monitoreo y detección de cadenas nacionales de radio y televisión contáctenos por los correos: wilderjimenez9@gmail.com y luzmapenagosc@gmail.com