Diseño De Un Sistema De Audio Distribuido Con Tecnología WIFI Para Ser Implementado Dentro De Redes De Área Local. (DADWIFI)

Karen Janeth Morales Gómez

Trabajo de Grado para Optar al Título de Ingeniero Electrónico

Director

Ariel Yezid Villareal Solano

Magister en ingeniería de sistemas e informática

Codirector

Homero Ortega Boada

Doctor en ciencia de la ingeniería

Universidad Industrial de Santander

Facultad de Ingeniería Físico-Mecánica

Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones

Ingeniería Electrónica

Bucaramanga

2024

Dedicatoria

A mis padres, por su apoyo inquebrantable, inconmensurable y firme, quienes han sido los pilares en cada paso de mi corto trasegar. Gracias por creer y hacerme creer en mí, por brindarme su amor incondicional, motivándome para no desfallecer y seguir avante, airosa al final de cada tramo.

A mis abuelos, tanto maternos como paternos, quienes me inculcaron valores y enseñanzas que han constituido un norte en mi vida, a mi abuelo Álvaro a quien tenemos la fortuna de poder disfrutar de su compañía; sus principios y cariño han sido un ejemplo que llevo en mi corazón.

A mis tíos, quienes siempre han estado al tanto de mis avances y óbices, han creído en mí y comparten esta alegría junto a mis padres.

A mis padrinos, quienes con sus consejos y guía se convirtieron en mis sombras tutelares. Gracias por estar ahí, brindándome su sabiduría y comprensión en momentos claves.

Y al Ingeniero Jair Trujillo, compañero de carrera, cuyo apoyo y amistad han sido esenciales en este itinerario. Gracias por ser un aliado en este sendero plagado por numerosos desafíos.

Karen Janeth Morales Gómez

Agradecimientos

Tanto a la Universidad Industrial de Santander –UIS- como al Servicio Nacional de Aprendizaje –SENA-, instituciones que son grandes referentes en la formación integral y profesional en nuestro país. Gracias por su dedicación a la noble misión de llevar a cabo procesos de enseñanza/aprendizaje de las ciencias y la tecnología altamente calificados y por brindarnos oportunidades de crecimiento y desarrollo profesional en cada etapa de nuestra formación.

A mis directores y profesores de carrera, maestros sherpas, fuente de inspiración en este arduo ascenso. Sus conocimientos, compromiso y apoyo han sido fundamentales para alcanzar cada cima y superar este reto.

A la Corporación Autónoma Regional para la Defensa de la Meseta de Bucaramanga –CDMB- en cabeza del Dr. Juan Carlos Reyes Nova, por su valioso aporte a mi formación al aceptarme realizar prácticas SOPIT en la Coordinación de Conocimiento, específicamente en el seguimiento y control de las redes de monitoreo de la calidad del aire.

Tabla de Contenido

Pág.

Introducción 13

1. Objetivos 15

1.1 Objetivo General 15

1.2 Objetivos Específicos 15

2. Marco Conceptual 16

2.1 ¿Qué es un Sistema de Audio Distribuido? 16

2.2 Introducción a los Sistemas Embebidos y la Raspberry PI 17

2.3 Redes de Área Local 18

2.3.1 Tecnologías WIFI en redes de área local 19

3. Proceso de Diseño 20

3.1 Restricción de Hardware 20

3.2 Restricción de Software en el Sistema 21

3.3 Análisis y Método de Selección de Componentes de Diseño 21

3.3.1 Matriz de Selección de Enrutador de Red Local 22

3.3.2 Matriz de Sistemas Embebidos 22

3.3.3 Matriz de Selección de Salida de Audio 24

3.4 Arquitectura General del diseño de audio distribuido 24

3.5 Instalación del Sistema Operativo 26

3.6 Arquitectura Servidor-Cliente 27

4. Implementación de Diseño 28

4.1 Implementación de la Red LAN Y Configuración del Router 29

4.2 Implementación del Servidor. 30

4.2.1 Estructura de Carpetas para el Servidor e Implementación 30

4.3. Implementación del Cliente 32

4.4. Interfaz Web 33

4.5 Interfaz del aplicativo móvil. 34

5. Verificaciones De Rendimiento Y Fiabilidad 36

5.1 Pruebas de Sincronización y Estabilidad de Módulos de Audio Distribuido 36

5.2 Pruebas de rendimiento 37

5.3 Pruebas de fiabilidad. 39

6. Costo Total del Sistema de Audio 41

7. Conclusiones 42

8. Recomendaciones 42

Referencias Bibliográficas 43

Apéndices 44

Lista de Tablas

Pág.

Tabla 1. *Matriz de selección de enrutador de red local* 22

Tabla 2 *Matriz de selección sistemas embebidos* 22

Tabla 3. *Matriz de selección de salida de audio*..... 24

Lista de Figuras

Pág.

Figura 1. *Arquitectura general.* 25

Figura 2. *Red tipo estrella* 26

Figura 3. *Raspberrys Pi 4 con tarjetas microSD* 27

Figura 4. *Arquitectura Servidor-Cliente* 28

Figura 5. *Interfaz del router Linksys WRT54G* 29

Figura 6. *Carpeta Servidor_RPIa* 30

Figura 7. *Consola del Servidor ejecutada* 32

Figura 8. *Interfaz web* 34

Figura 9. *Interfaz móvil* 35

Figura 10. *Consola del Servidor, ejecución del programa D-ITG.* 38

Figura 11. *Consola del Cliente 1, Raspberry Pi, ejecución D-ITG.* 38

Figura 12. *Consola del Cliente 2, Raspberry Pi, ejecución D-ITG.* 39

Figura 13. *Consola del Cliente 3, Raspberry Pi, ejecución D-ITG.* 39

Figura 14. *Ping en la Raspberry 1* 40

Figura 15. *Ping en la Raspberry 2* 40

Figura 16. *Ping en la Raspberry 3* 41

Lista de Apéndices

pág.

Apéndice A. *Documentación oficial de la Raspberry PI en GitHub*..... 44

Apéndice B. *Carpeta Servidor_RPI en GitHub.*..... 44

Apéndice C. *Repositorio de la librería de PyAudio en GitHub.* 44

Apéndice D. *Repositorio de la librería de PyAudio en GitHub.* 44

Apéndice E. *Repositorio de la librería de Pydub en GitHub.* 44

Apéndice F.*Repositorio de la librería de FFmpeg en GiyHub.* 44

Apéndice G. *Carpeta Cliente_RPI en GitHub.*..... 44

Apéndice H.. *Repositorio de la carpeta para la creación de la app móvil.* 44

Apéndice I. *Video ilustrativo de verificación en link en drive.* 44

Glosario

802.11: Estándar por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), el cual define los protocolos de comunicación en redes WI-FI inalámbricas.

Ancho de Banda: Capacidad máxima de transmisión de datos en una red, esta dado por bits por segundo (bps), determinan la cantidad de información envidad en un tiempo determinado.

CMD: Command Prompt o también llamado símbolo del sistema, es una interfaz de línea de comandos en el sistema operativo de Microsoft, para la ejecución de comandos.

Comandos: Instrucción dada en la consola o terminal para que el sistema operativo o programa realice una tarea específica.

CSS (Cascading Style Sheets): Lenguaje utilizado para la descripción presentación visual de un documento HTML, para la aplicación de estilos y diseños de páginas web.

D-ITG (Distributed Internet Traffic Generator): Herramienta el cual permite la simulación dl tráfico de red, analizando el rendimiento y la calidad del servicio en redes IP.

HTML (HyperText Markup Language): Lenguaje utilizado para crear y estructurar el contenido en las páginas web.

IP (Internet Protocol): Protocolo de comunicación en redes informáticas que permite la identificación y selección de dispositivos para el envío y recepción de datos en una red.

JavaScript: Lenguaje de programación principalmente utilizado para la creación de contenido en páginas web.

LAN (Local Area Network): Red de área local que conecta dispositivos dentro de una ubicación limitada.

Latencia: Tiempo que tarda un paquete de datos desde el origen hasta el destino por medio de la red. Principalmente utilizado para el análisis del rendimiento de las comunicaciones en tiempo real.

Mbps (Megabits per Second): Unidad de medida de velocidad de transmisión de datos, megabits transferidos por segundo.

Python: Lenguaje de programación, utilizado para el desarrollo web, automatización y análisis de datos.

Raspberry Pi: Sistema computacional de bajo costo que permite el desarrollo de proyecto en tiempo real.

Router: Dispositivo que permite la creación de la red, la conexión de múltiples dispositivos y el enrutamiento de datos.

Sistemas Embebidos: Sistemas de computación que integran el software y hardware para la realización de tareas específicas.

Sockets: Interfaz de programación de aplicaciones (API), permita la comunicación de dispositivos por medio de una red.

WI-FI: Tecnología que permite la conexión de manera inalámbrica entre los dispositivos para la transmisión de datos en redes locales.

Resumen

Título: Diseño de un sistema de audio distribuido con Tecnología WIFI para ser implementado dentro de redes de área local. (DADWIFI)

Autor: Karen Janeth Morales Gómez[**]

Palabras Clave: Audio distribuido, Sistema embebidos, Red de Área Local (LAN), Programación en Python, Servidores web, Interfaz web y móvil.

Descripción: El documento que se presenta a continuación contiene el diseño de un sistema de audio distribuido que, de manera tal que usando tecnologías WIFI y aplicando la distribución de audios de manera compartimentada se logra la transmisión de salidas de audio controlando, por medio de sistemas embebidos.

Además de lo anteriormente expuesto, este trabajo muestra los detalles de diseño e implementación, los cuales fueron clave para la realización de este proyecto por medio de servidores web se desarrolla la programación en Python, para el control se diseña una interfaz web la que ayuda al diseño de un aplicativo móvil, todo esto conecto a una red de área local (LAN), mediante análisis y ensayos se llegó a un resultado que aquí se muestra.

En ese sentido, se parte de la base de que volumen no es igual a calidad y este trabajo lo demuestra, puesto que el diseño permite mayor control de variables que permiten una mayor calidad en el sonido que se entrega a cada usuario.

---

*Trabajo de Grado

**Escuela de Ingeniería Física y Mecánica. Escuela Técnica Superior de Ingenieros Eléctricos, Electrónicos y de Telecomunicación. Ingeniería Electrónica. Director: Ariel Villareal Losano. Máster en Ingeniería de Sistemas e Informática. Codirector: Homero Ortega Boada. Doctor en Ciencias de la Ingeniería.

Abstract

Title: Design of a distributed audio system with WIFI technology to be implemented within local area networks. (DADWIFI) *

Author(s): Karen Janeth Morales Gómez **

Key Words: Distributed audio, Embedded systems, Local Area Network (LAN), Python programming, Web servers, Web and mobile interfaces.

Description: The document presented below contains the design of a distributed audio system that, using WIFI technologies and applying the distribution of audio in a compartmentalized way, the transmission of audio outputs is achieved by controlling, through embedded systems.

In addition to the above, this work shows the details of design and implementation, which were key to the realization of this project through web servers programming is developed in Python, for the control is designed a web interface that helps the design of a mobile application, all this connected to a local area network (LAN), through analysis and testing came to a result that is shown here.

In that sense, it is assumed that volume does not equal quality and this work demonstrates it, since the design allows greater control of variables that allow higher quality sound that is delivered to each user.

---

*Degree Work

**School of Physical and Mechanical Engineering. School of Electrical, Electronic and Telecommunications Engineering. Electronic Engineering. Director: Ariel Villareal Losano. Master's degree in systems and computer engineering. Co-director: Homero Ortega Boada. Ph.D., of Engineering Sciences

Introducción

En la actualidad el aumento de las telecomunicaciones ha generado un incremento de las redes WIFI las cuales han generado flexibilidad y un bajo costo presentando grandes ventajas para su instalación en cualquier entorno, (Baldeón,2019). Este servicio inalámbrico permite la comunicación entre dispositivos de forma viable y práctica. En el actual proyecto se presenta un diseño de un sistema de audio distribuido, el cual emplea esta tecnología para ofrecer una experiencia versátil de forma inalámbrica.

El diseño e implementación se fundamenta en el estándar 802.11 que permite la comunicación entre dispositivos que se encuentren entrelazados por medio de una red de área local (LAN). Si a esto se le agrega la tecnología de las Raspberry PI usada como plataforma de desarrollo, se pueden construir prototipos capaces de generar servicios como lo es en este caso un sistema de audio distribuido que permite sectorizar el sonido por zonas según las necesidades del usuario, permitiendo la reproducción de diferentes pistas de audio en los diferentes dispositivos conectados a la red.

El resultado será una solución innovadora y práctica para la distribución de audio inalámbrico en entornos locales que sea fácil de usar y accesible para todos los usuarios. Con este diseño los usuarios podrán disfrutar de un sistema de audio en cualquier entorno y en cualquier momento con una fiabilidad en la plataforma diseñada, lo cual mejorará su experiencia.

Así beneficiará a la comunidad teniendo en cuenta que los usuarios del sistema tendrán un sistema cómodo y de mejor calidad por tener la facilidad de escuchar audio dentro de grandes entornos sin tener que utilizar sonidos de alto volumen. Al homogeneizar el sonido, este sistema permitirá disminuir la contaminación auditiva de diferentes entornos y eventos multitudinarios, favoreciendo a la salud pública y el medio ambiente. Además, al ser un trabajo realizado con

desarrollo en el área de las telecomunicaciones, será un gran aporte al grupo de investigación RadioGis.

1. Objetivos

1.1 Objetivo General

Diseñar un sistema de audio distribuido con tecnología WIFI para ser implementado dentro de redes de área local

1.2 Objetivos Específicos

Seleccionar dispositivos que permitan reproducción de audio controlado inalámbricamente mediante la tecnología WIFI con el fin de construir 3 módulos receptores del sistema de audio distribuido.

Crear una aplicación móvil que permite seleccionar el sonido específico para cada uno de los módulos construidos.

Implementar el sistema de audio distribuido usando una red WIFI con el fin de realizar pruebas de rendimiento y fiabilidad de la plataforma diseñada.

2. Marco Conceptual

Como etapa preliminar al desarrollo de este proyecto se procede a delimitar el concepto de audio distribuido y de esta manera comprender los dispositivos que se deben implementar para ejecutar de manera satisfactoria cada una de las fases que conlleven al cumplimiento de los objetivos planteados en el presente trabajo.

2.1 ¿Qué es un Sistema de Audio Distribuido?

Históricamente se ha reconocido el audio como una forma de ambientar o mejor, customizar diferentes espacios en variados entornos geográficos. Sin embargo, tradicionalmente, en numerosas y diversas culturas, se ha tenido el concepto de que, a mayor volumen e intensidad de sonido se tiene una mejor claridad en cuanto optimización de audio se refiere. Coloquialmente se puede decir, que el sistema de audio que más fuerte suene, era considerado como el de mejor calidad. Hoy en día el concepto ha cambiado, y en ello contribuye en gran medida el sistema de audio distribuido.

Es así que, un entorno digital moderno necesita que el sistema de audio esté enfocado al manejo adecuado de los tonos o frecuencias sonoras –musicales o de locución-, y elimine por completo cualquier tipo de distorsión armónica. Dicho de otro modo, este audio, para ser escuchado con claridad y calidad no requiere de potencia, sino de poseer varios puntos de salida distribuidos dentro del lugar que se desea ambientar musicalmente.

Para lograr esto, se requiere de transmisión de sonido hacia diferentes ubicaciones de manera independiente, lo cual constituye la inmersión en el sistema de dispositivos cableados, los cuales visualmente pueden destruir la armonía elegante de un entorno específico. Una solución para esto es planificar previamente el diseño estructural de las edificaciones, pero en a la mayoría de los casos, un usuario primero construye y después piensa en elementos que complementen de

manera agradable dicho ambiente. Es precisamente para estos casos en los que los sistemas inalámbricos hacen su aparición y representan la mejor solución.

Para ello se requiere el uso de tecnologías inalámbricas, y entre estas se destaca el WIFI. Esta tecnología es de banda ancha, lo cual permite que proporcione una experiencia auditiva de buena calidad. Si a esto se le añade un control por medio de una aplicación móvil, se podría estar hablando de tener un sistema controlado de manera remota y con conectividad a Internet, lo cual le da la capacidad de integrar múltiples fuentes de audio con múltiples puntos de emisión.

2.2 Introducción a los Sistemas Embebidos y la Raspberry PI

Con base en la evolución de la inteligencia computacional aplicada a los sistemas embebidos, se ha integrado una amplia gama de proyectos y aplicaciones los cuales ofrecen múltiples funcionalidades avanzadas. Entre las características que definen a los sistemas embebidos se encuentran la eficiencia energética, el tamaño reducido y la alta fiabilidad (Ganssle, 2008). Otra característica es la capacidad para operar en tiempo real, que asegura que el sistema pueda responder de manera inmediata a eventos externos (Heath, 2002).

Lo anterior se sustenta en el hecho de que el impacto de las Raspberry PI se refleja en su uso en una variedad de aplicaciones, desde la educación hasta la creación de prototipos, pasando por un interesante campo comunicacional que avanza día a día. La flexibilidad de estas placas permite que sean utilizadas para una amplia gama de proyectos, que van desde la enseñanza de conceptos básicos de programación hasta la creación de soluciones tecnológicas avanzadas. Esta versatilidad no solo fomenta el aprendizaje práctico, sino que también impulsa la innovación al permitir que los usuarios experimenten y desarrollen nuevas ideas sin las limitaciones derivadas del uso de tecnologías más costosas (Upton & Halfacree, 2014).

Es más, la flexibilidad de las Raspberry PI permite su aplicación en una variedad de proyectos, desde la creación de prototipos hasta la automatización del hogar; ello porque el uso de estas placas en la creación de prototipos permite a los desarrolladores experimentar con nuevas ideas sin las limitaciones financieras impuestas por tecnologías más costosas (Sutherland, Haddon, & Williams, 2016). Esta misma capacidad para experimentar y desarrollar innovaciones es crucial para el avance tecnológico y la creación de soluciones nuevas y creativas (Sutherland, Haddon, & Williams, 2016).

Las Raspberry PI también juegan un papel importante en la inclusión tecnológica. La capacidad de estas placas para reducir las barreras económicas y proporcionar una plataforma tecnológica y económicamente asequible, ha permitido a personas y comunidades que tradicionalmente estarían excluidas de la tecnología participar en el desarrollo tecnológico (McCormick, 2017).

2.3 Redes de Área Local

Las redes de área local (Local Area Network, LAN) constituyen una de las principales herramientas tecnológicas mayormente utilizadas para interconectar dispositivos y facilitar la comunicación dentro de un área geográfica limitada x, como una oficina, un edificio o un campus. Estas redes permiten compartir recursos como impresoras, almacenamiento, aplicaciones e incluso acceso a internet, facilitando la colaboración entre los usuarios y aumentando la productividad. La implementación de las LAN en entornos empresariales y domésticos ha demostrado ser una inversión estratégica, ya que no sólo optimiza las operaciones, sino que también mejora la accesibilidad y el rendimiento de los sistemas de comunicación (Stallings, 2020).

Por otra parte, las LAN tienen la propiedad de establecer conexiones rápidas y estables, lo cual las convierte en un componente primordial en la infraestructura de redes, aunque,

dependiendo del medio utilizado para la transmisión de datos, las LAN pueden tipificarse en dos grupos a saber, cableadas o inalámbricas. Las primeras utilizan tecnologías como Ethernet, que emplean cables físicos para conectar dispositivos, garantizando altas velocidades de transmisión y baja interferencia. Por otro lado, las segundas, que se basan en tecnologías como WIFI, han experimentado un crecimiento exponencial debido a su flexibilidad y facilidad de implementación en entornos dinámicos. Dicho fenómeno se ha visto impulsados, en gran medida, por la adopción de estándares como IEEE 802.11, los cuales han hecho que ocurra avances significativos en aspectos clave tales como cobertura, calidad, capacidad y seguridad (Kurose & Ross, 2021).

2.3.1 Tecnologías WIFI en redes de área local

El avanzado desarrollo de tecnologías WIFI ha venido transformado, de manera acelerada, la manera en que se establecen las redes de área local. WIFI, la cual a su vez se basa en estándares como IEEE 802.11, red que permite la conexión de dispositivos a una red sin necesidad de cables, lo que proporciona una mayor libertad de movimiento y un acceso más conveniente a los recursos compartidos (Sharma et al., 2022).

En consecuencia, la implementación de redes WIFI en áreas locales ofrece, hoy por hoy, múltiples beneficios, como por ejemplo la reducción de costos de instalación, lo cual es relevante para cualquier organización, puesto que se mitigan los gastos asociados con instalación e insumos, v. gr. el cableado. Sumando ventajas a lo anteriormente escrito, las redes WIFI permiten la conexión de una amplia variedad de dispositivos, desde computadoras y teléfonos inteligentes –smart phones- hasta dispositivos IoT, lo que contribuye a un entorno mucho mejor en interconexión y eficiencia. (Kumar & Singh, 2023).

3. Proceso de Diseño

En el presente capítulo, se aborda el proceso de diseño del audio distribuido, aplicando los conocimientos que se adquirieron en la disciplina –Ingeniería Electrónica- en las áreas de redes de comunicación y en Python. Este proceso de diseño tiene la finalidad implementar la transmisión de audio mediante tecnología WIFI por medio del desarrollo configuraciones específicas de Software Y Hardware; éstas a su vez establecen bases sólidas para las conexiones que requiere el sistema que aquí se diseña y se muestra.

Para lograr este propósito se realizó la selección de componentes para el diseño y la instalación de un sistema embebido con base en la capacidad de este tanto para ejecutar scripts de Python como la conexión a red, para lo cual se contempla la integración de tres sistemas embebidos que fungirán como clientes en la transmisión de audio a través de las salidas de audio como son, en esta ocasión, las bocinas previamente seleccionadas las cuales se adapten a estos sistemas embebidos. Por otra parte, la necesidad de una red local (LAN) se investiga que tipología es la adecuada para la conexión de los dispositivos y cumpla con los requisitos de una conectividad inalámbrica estable y de baja latencia.

En ese orden de ideas, este diseño explora los servidores web, el cual administra y distribuye el audio a los clientes, y como es integrado a una interfaz web, para posteriormente explorar el desarrollo de un aplicativo móvil para la transmisión de audio con el fin de ofrecer una experiencia amigable e intuitiva.

3.1 Restricción de Hardware

La integración de los componentes de hardware debe considerar el entorno físico adecuado, lo que se traduce en condiciones mínimas del escenario en el cual se va a implementar el sistema, de manera tal que la cobertura de la red resulte óptima y garantice que los sistemas embebidos

posean una conexión robusta, confiable, eficaz; para que esto ocurra, es necesario que los elementos clave sean ubicados estratégicamente distribuidos; allí es posible que las salidas de audio sean conectadas por medio de cable tipo jack, el cual permite, en este caso, la integración normal simplificada al momento de transmisión.

Para que esto ocurra la interconexión de los componentes se ha de realizar con las bondades de las configuraciones del router, lo cual requiere establecer una red local (LAN); así, de esta forma, favorece la fluidez de la comunicación entre dispositivos; en este proceso es necesario tener en cuenta las direcciones IP de cada sistema-componente para, de manera más eficiente, cumplir con las conexiones requeridas.

3.2 Restricción de Software en el Sistema

Las restricciones de software también se presentan en este sistema, las cuales se pueden ubicar en diferentes puntos de este y se identifican porque se requiere limitar y controlar elementos de los sistemas embebidos. Para disminuir el riesgo e incrementar su robustez, se vienen desarrollando estrategias como scripts en Python; éstas permiten la programación de cada dispositivo, aplicando la referencia de servidores web, en estos sistemas se está utilizando tanto el entorno base como el virtual con el objeto de fortalecer la organización estructural.

Además, se requiere una interfaz web permitiendo la facilidad de visualización del servidor y de los clientes o usuarios, lo cual se ha complementado con un aplicativo móvil de un aplicativo móvil, que permita la interacción del sistema, selección de audio, ajuste de volumen, pausa y detención del audio.

3.3 Análisis y Método de Selección de Componentes de Diseño

En este apartado se lleva a cabo el análisis y la selección de componentes clave para el diseño de audio distribuido, tales como el sistema embebido, las bocinas y el router. Se tiene en

cuenta características como la capacidad de integración, el rendimiento y la fiabilidad. Para facilidad de evaluación de múltiples alternativas y optimizar la selección, se utiliza matrices comparativas, estas matrices ofrecen criterios de selección para cada componente necesario.

3.3.1 Matriz de Selección de Enrutador de Red Local

En la Tabla 1 se puede apreciar las diferentes características de los router, el Linksys WRT54G ofrece velocidades de transmisión óptimas de hasta 150 metros, lo que es suficiente para la aplicación de audio distribuido. Además, la facilidad de configuración y soporte para múltiples estándares lo convierten en una opción accesible y segura. En contraste, el TP-Link Archer A7 y el Netgear Nighthawk presentan costos elevados y características complejas, lo que es menos ideal para proyectos de audio distribuido donde la estabilidad de la conexión es necesaria.

Tabla 1.

Matriz de selección de enrutador de red local

Características	**WRT54G**	**TP-Link Archer A7**	**Netgear Nighthawk R6700**
Velocidad	54 Mbps	1750 Mbps	1750 Mbps
Rango de Cobertura	Hasta 150 m	Hasta 200 m	Hasta 200 m
Tecnología	802.11g	802.11ac	802.11ac
Seguridad	WEP, WPA, WPA2	WPA, WPA2	WPA, WPA2
Facilidad de Configuración	Alta	Alta	Media
Precio	Bajo	Moderado	Alto

3.3.2 Matriz de Sistemas Embebidos

Al observar la Tabla 2, se compara los comportamientos de los sistemas embebidos BeagleBone Black y el ESP32, la Raspberry PI 4. Como resultado de esto, la Raspberry PI 4 es el

sistema embebido más completo, por otro lado, la BeagleBone Black es un sistema robusto y adecuado para aplicaciones, cuenta con una capacidad de procesamiento inferior y además requiere un adaptador adicional para WIFI, lo que lo es menos conveniente para el diseño.

El componente ESP32, con conexión WIFI, carece de poder de procesamiento y capacidad de manejar tareas complejas frente a la Raspberry PI 4, en consideración al rendimiento en cuanto el procesamiento de la información y conectividad más robusta, ofreciendo relación calidad/precio de lo cual se deduce que es esta la mejor opción.

Tabla 2

Matriz de selección sistemas embebidos

Características	**Raspberry PI 4**	**BeagleBone Black**	**ESP32**
Procesador	Quad-core Cortex-A72 (ARM v8) a 1.5 GHz	1GHz ARM Cortex-A8	Xtensa Dual-Core 32-bit LX6 a 160-240 MHz
RAM	2GB, 4GB, o 8GB LPDDR4	512MB DDR3	520 KB SRAM, 4MB PSRAM
Conectividad	WIFI 802.11ac, Bluetooth 5.0, Gigabit Ethernet	No WIFI integrado, Ethernet 10/100	WIFI 802.11b/g/n, Bluetooth 4.2
Almacenamiento	MicroSD, SSD externo (USB)	4GB flash interno, MicroSD	Flash integrado de 4MB
Sistema Operativo	Raspberry PI OS (Debian) y otros	Debian y otras distribuciones de Linux	FreeRTOS, otros RTOS
Costo	$35	$50	$5-$10

3.3.3 Matriz de Selección de Salida de Audio

La Tabla 3 se relacionan tres referencias distintas para el sistema de audio, esta selección se decantó por la bocina Maxell Bass 13 la cual se destaca como la opción ideal debido a su calidad-precio proporcionando un rendimiento sólido. Cuenta con conexión por cable de 3.5 mm asegura una transmisión de audio estable y sin interferencias.

Tabla 3.

Matriz de selección de salida de audio

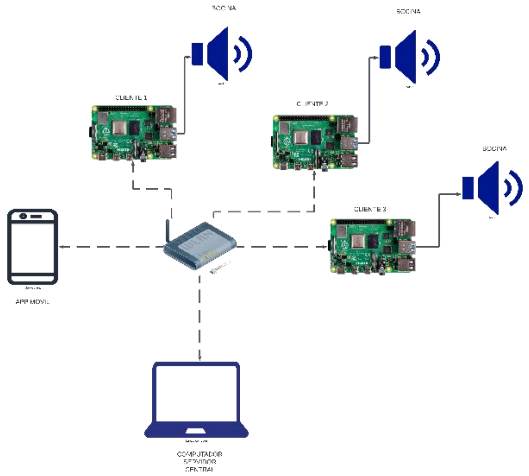
Característica	**Maxell Bass 13**	**Sony SRS-XB43**	**Bose SoundLink Flex**
Tipo de Salida	Jack 3.5 mm	Bluetooth, entrada USB	Bluetooth, entrada USB
Calidad de Sonido	Buena, sonido equilibrado	Sonido potente con graves profundos y claridad	Sonido balanceado, falta de profundidad en graves
Precio	Económica (aprox. $30)	Alta (aprox. $350)	Alta (aprox. $150)
Potencia de Salida	10W	100W	30W
Frecuencia de Respuesta	60Hz - 20kHz	20Hz - 20kHz	100Hz - 20kHz
Duración de Batería	Hasta 6 horas	Hasta 24 horas	Hasta 12 horas
Portabilidad	Moderada (cable)	Moderada (grande y pesada)	Alta (compacta y ligera)
Facilidad de Uso	Sencillo (conexión por cable)	Fácil (emparejamiento Bluetooth y USB)	Fácil (emparejamiento Bluetooth)

3.4 Arquitectura General del diseño de audio distribuido

En esta sección se presenta la estructura general del diseño de audio distribuido como se muestra en la Figura 1**,** el cual se basa en el modelo servidor-cliente. En este modelo, un computador actuará como el servidor web principal, mientras que otras tres Raspberry PI funcionarán como clientes. Este servidor web se ajusta para la gestión del audio, facilitando así la transmisión de sonido a través de la red.

Figura 1.

Arquitectura general



Nota: Estructura general llevada a cabo el proceso de diseño para el sistema de Audio Distribuido por medio de tecnologías WIFI

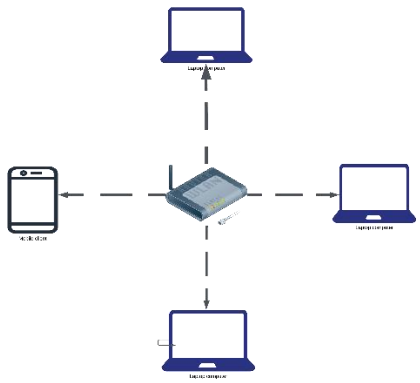
La operación de las Raspberry PI requiere la instalación del sistema operativo, en este caso, Raspberry PI OS, donde se configurará la red. Se implementó el lenguaje de programación Python para desarrollar los scripts necesarios, utilizando bibliotecas como Flask para la creación de páginas web que favorecen la interacción entre el servidor y los clientes. Además, se empleó la biblioteca PyAudio para la reproducción de audio a través de diferentes dispositivos de salida, permitiendo distribuir el sonido entre las Raspberry PI.

Esta red cuya configuración es de tipo estrella proporciona al router un punto central de conexión para todos los dispositivos; diseño que asegura que cada Raspberry PI esté conectada al servidor, lo que garantiza una transmisión de datos eficiente. Por otro lado, el desarrollo de una interfaz web donde se manejan conocimientos de HTML, CSS y Javascript lo que facilita la

interacción fluida con el sistema de audio, teniendo en cuenta la necesidad de controlar de manera intuitiva.

Figura 2.

Red tipo estrella



Nota: Topología tipo estrella

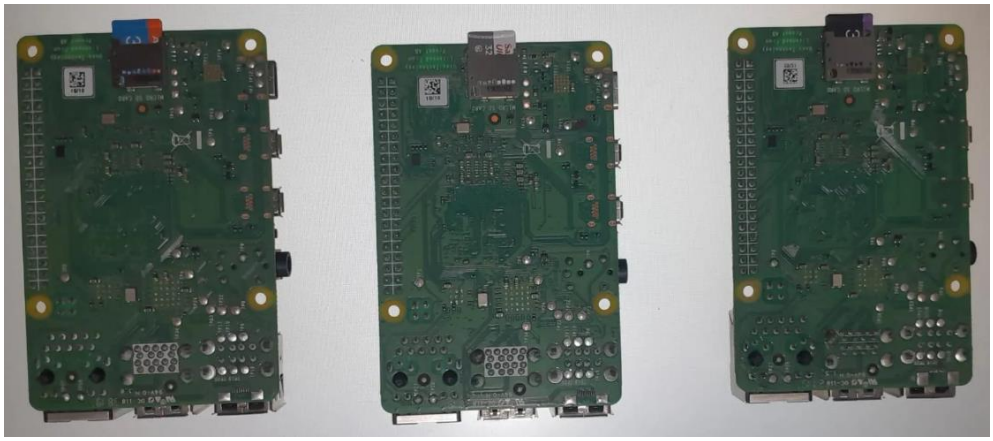
3.5 Instalación del Sistema Operativo

Para la instalación del sistema operativo, es necesario disponer de la Raspberry PI 4 y una tarjeta microSD, se sigue el procedimiento descrito en la documentación oficial que se encuentra en el Apéndice A. Se instaló en el computador la herramienta Raspberry PI Imager la cual facilita la instalación de sistema operativo.

Posteriormente, se formateó la microSD de 32GB y se utiliza la herramienta Raspberry PI Imager para seleccionar y escribir la imagen de Raspberry PI OS en la microSD. Una vez finalizado este proceso, se conectó el monitor, el teclado y el ratón a la Raspberry PI 4, y se insertó la tarjeta microSD en su ranura. Se conectó la fuente de alimentación para que la Raspberry PI 4 arranca desde la tarjeta microSD.

Figura 3.

Raspberrys Pi 4 con tarjetas microSD



Nota: Preparación de tres Raspberry PI 4 con tarjetas microSD formateadas para instalación del sistema operativo

En la pantalla se completó la configuración inicial, incluyendo la selección del idioma, la configuración del Wi-Fi y la actualización del sistema operativo. Este mismo proceso se repite dos veces más para configurar las otras dos Raspberry PI 4, utilizando sus respectivas tarjetas microSD ya preparadas.

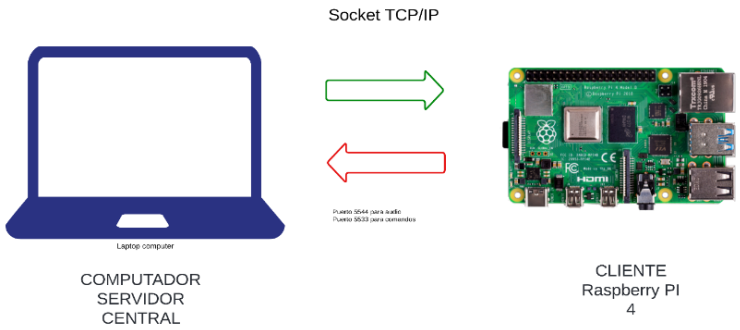
3.6 Arquitectura Servidor-Cliente

Este diseño basa su arquitectura en servidores web que, al facilitar la gestión de transmisión y recepción de datos, en este caso el servidor central que es por medio del computador, por la alta capacidad procesamiento y almacenamiento de datos, permite el manejo múltiples conexiones y audios, también favorece la integración de diversos clientes –usuarios, puntos de manera simultánea.

Esta red está conformada por tres Raspberry PI 4, que actúan como clientes y son programadas en Python para facilitar conexión con el servidor central en donde se reciben la orden y funciones mencionadas en la sección 3.2.

Figura 4.

Arquitectura Servidor-Cliente



Nota: Arquitectura Servidor-cliente por medio de sockets.

4. Implementación de Diseño

En este capítulo se presenta la implementación del diseño que se llevó a cabo en el sistema de audio distribuido a través de la arquitectura servidor-cliente, para lo cual se tuvo en cuenta la configuración de cada componente. Tomando como base al proceso estructurado de diseño, previamente analizado en el capítulo anterior y al desarrollo en Python, se dispuso de la comunicación bidireccional entre los dispositivos que componen el sistema. Esta configuración se traduce en la fase que integra el uso de diversas librerías, la configuración de sockets y la organización lógica que acceden la interacción entre el servidor y los clientes, lo que permite contar con una interfaz que facilita la interacción con los clientes y, a su vez implementa una aplicación, cimentada con las características antes descritas.

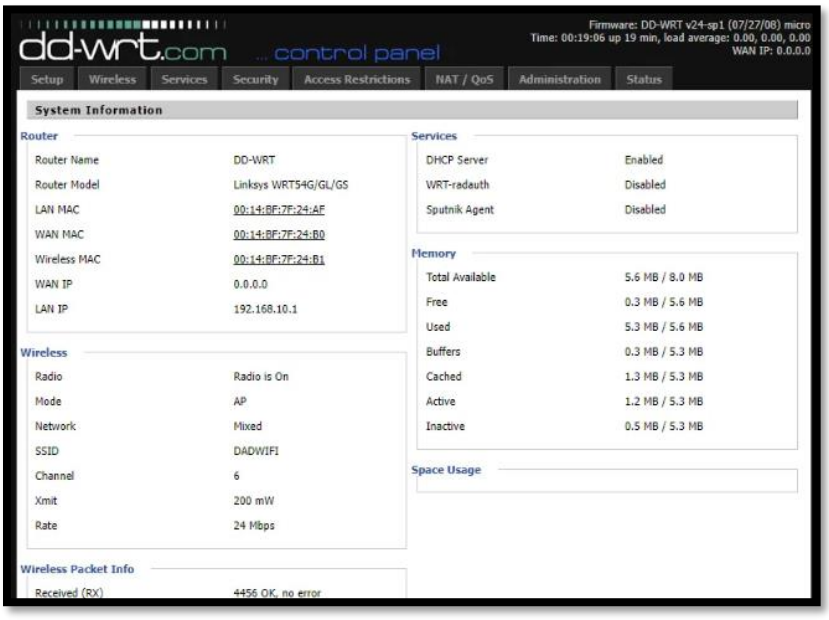
4.1 Implementación de la Red LAN Y Configuración del Router

Para la implementación de la red LAN fue necesario estructurar la topología en estrella, que consistió en la conexión a un nodo central (router). La configuración de esta clase de topología se llevó a cabo porque tiene la característica importante de favorecer la conectividad de manera confiable y eficaz.

Como complemento a lo anteriormente expuesto y con el fin de efectuar la configuración del router Linksys WRT54G, fue preciso contar con la dirección IP, la cual se encuentra en la puerta de enlace predeterminada, mediante la consola de comandos (cmd); ejecutando el comando ipconfig/all, el cual extiende una lista con los adaptadores de red. En el apartado correspondiente al adaptador LAN, se visualiza la dirección IP requerida y, una vez obtenida, se ingresa al navegador para acceder al panel de configuración del router.

Figura 5.

Interfaz del router Linksys WRT54G



Nota: Interfaz router Linksys WRT54G la cual se llevará a cabo la configuración

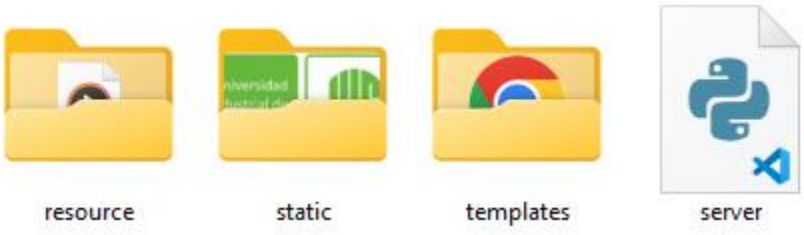
Dentro de la interfaz Linksys WRT54Gse accede al panel de administración. En la sección de “Basic Setup”, se configura la red LAN ajustando la dirección IP y definiendo el rango de la red. En la pestaña “Wireless”, se cambia el nombre de la red (SSID), en este caso a DADWIFI, seguido de la pestaña "Wireless Security", donde se establece la contraseña DADWIFI2024 con el protocolo de seguridad WPA2, con el objetivo de garantizar la privacidad de la red.

4.2 Implementación del Servidor.

Para la implementación del servidor central, se precisa generar archivos y carpetas, los cuales son esenciales para su óptimo funcionamiento. Por estas razones ha sido creada la carpeta Servidor_RPI en el ambiente base, al interior del cual se encuentra subcarpetas organizadas como scripts (para los archivos de Python), templates (para las vistas en HTML), y resource (para almacenar los recursos multimedia y otros archivos necesarios), Esta carpeta se puede apreciar en el Apéndice B.

Figura 6.

Carpeta Servidor_RPI



4.2.1 Estructura de Carpetas para el Servidor e Implementación

La carpeta Servidor_RPI se organiza en varios subdirectorios esenciales para el funcionamiento del servidor. Entre ellos, se encuentra la carpeta templates, que contiene el archivo index.html, el cual define la interfaz web donde los usuarios pueden seleccionar y controlar las

canciones de manera intuitiva. También hay una carpeta resource, que almacena las canciones en formato MP3.

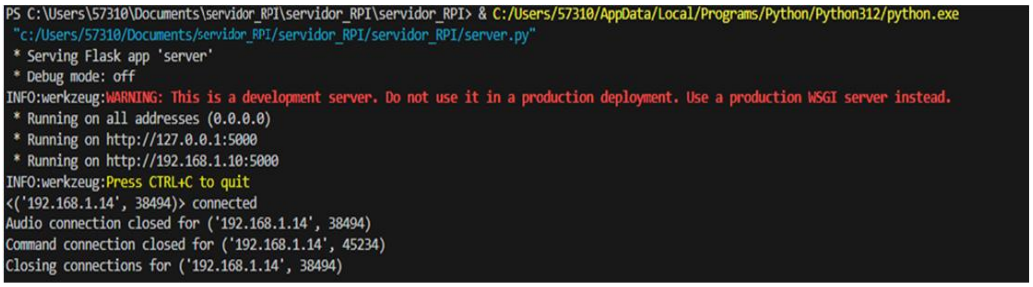
El código de Python ubicado en la carpeta principal es el encargado de la operación del servidor central. Este código integra librerías que aseguran el manejo de la transmisión de audio y la comunicación con los clientes. Las librerías más destacas son PyAudio, que cumple la funcionalidad del control y reproducción de audio; Flask, facilita la construcción de la interfaz web, operando las solicitudes de los clientes; y Pydub, que junto a FFmpeg (integrada en Pydub), permite convertir y procesar archivos de audio en diversos formatos. La Implementación de estas librerías es fundamental para llevar a cabo la transmisión de audio desde el servidor a múltiples clientes y facilitar el control a través de la interfaz web. Estas librerías se encuentran en un enlace que conduce a GitHub y están detalladas en los apéndices C, D, E y F.

En el código están establecidas dos conexiones mediante sockets para la transmisión a cada cliente: el puerto 5544 para la transmisión de audio y el puerto 5533 para el envío de comandos. Al establecer conexión con el cliente, se crea un hilo encargado de gestionar la conexión, asignando la clase client, que almacena la información relevante, como el estado en que se encuentra (pausa o detención) y la posición de la reproducción.

Para poner en marcha todo el sistema, es necesario ejecutar el programa en Python a través de la consola. Esta indicará que el servidor ya se encuentra en funcionamiento, señalando que la interfaz puede ser accedida por el navegador y es accesible en el puerto 5000. Una vez que el servidor esté activo, se debe ejecutar el archivo de Python correspondiente a los clientes para conectarse, y aparecerá en la consola la dirección IP. Estos también aparecerán en la interfaz web, permitiendo así la interacción intuitiva entre servidor y cliente.

Figura 7.

Consola del Servidor ejecutada



Nota: Ejecución del servidor, despliegue de la interfaz web y conexiones de clientes conectados por medio de la dirección IP.

4.3. Implementación del Cliente

El código en Python implementado en los clientes de audio distribuido permite el enlazamiento al servidor central para la recepción y transmisión de audio. Se establecen conexiones mediante sockets TCP: audio_socket y command_socket, siendo esenciales para que el cliente se conecte al servidor para recibir comandos de control. El uso de estos sockets permite la comunicación bidireccional entre el cliente y el servidor.

La configuración de la reproducción de audio, se utiliza la librería PyAudio, se definen parámetros como el formato de audio (FORMAT), el número de canales (CHANNELS) y la tasa de muestreo (RATE). Los datos de audio recibidos se reproducen a través del objeto stream, encargado de transmitirlos a la bocina. Al mismo tiempo, los clientes también procesan los comandos enviados por el servidor (como pausar, detener y ajustar el volumen).

El código cuenta con un bucle principal donde el cliente recibe continuamente los datos de audio provenientes del audio_socket. Cada paquete recibido se descomprime con struct. unpack() y se almacena temporalmente en un buffer. Una vez que se acumula una cantidad suficiente de

datos, estos se ajustan multiplicando cada muestra por el valor de volumen (volume) para controlar la intensidad de la reproducción. Para asegura un cierre limpio de la aplicación y evitar errores en las futuras conexiones, el flujo de audio se detiene.

Este código es fundamental en el diseño de sistemas de audio distribuidos, ya que permite recibir y controlar el audio transmitido desde un servidor central. La implementación de sockets y la gestión adecuada de la recepción y ejecución de comandos ofrecen a los usuarios una experiencia intuitiva para controlar la reproducción (volumen, pausa, y detener) desde la interfaz web del servidor.

Es necesario tener en cuenta que, para la conexión efectiva, se debe configurar la dirección IP del servidor en el código en Python del cliente, además de ejecutar el programa. Todos estos diferentes documentos y el código fuente se pueden encontrar en el anexo correspondiente o en el enlace de GitHub proporcionado en el Apéndice G.

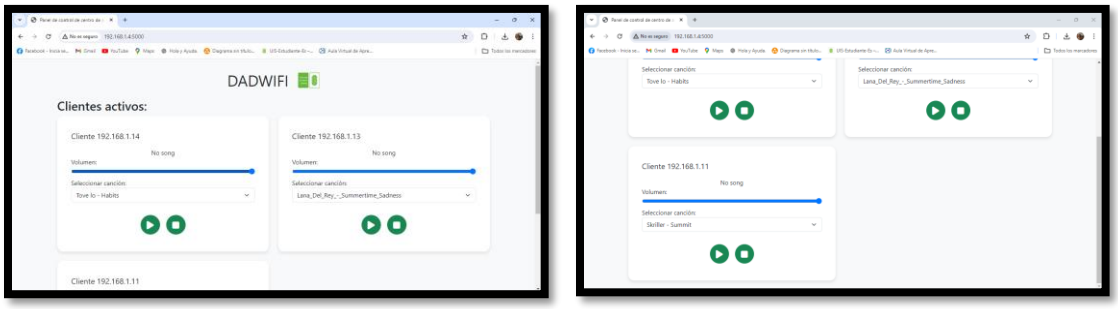
4.4. Interfaz Web

Con el fin de brindar un entorno de control visual e intuitivo, para la interacción con los clientes en el sistema de audio. La interfaz se localiza en el servidor, se accede por medio de la dirección IP del servidor en el navegador. Es muy importante que el servidor se haya ejecutado, seguidamente los clientes.

Mediante los clientes se van conectado se van presentado en la interfaz. Cada vez que un cliente se ejecuta y se establece conexión con el servidor, va apareciendo automáticamente en la lista de “Clientes Activos” se identifican con su respectiva dirección IP. La construcción de este panel de control incluye las funciones ya personalizadas, desde un menú desplegable se selecciona el audio en formato MP3, permitiendo que cada cliente tenga un control de sonido independiente.

Figura 8.

Interfaz web



Nota: Interfaz web intuitiva para el manejo de funcionalidades (reproducción, pausa, detener y ajuste en el volumen)

4.5 Interfaz del aplicativo móvil.

Con el propósito de desarrollar la aplicación móvil, se recurrió a Flutter, el cual consiste en un framework cuya función es la de permite generar nuevas aplicaciones en múltiples plataformas utilizando el lenguaje de programación Dart. La carácteristía primordial de Flutter es su capacidad para recolectar aplicaciones en Android, iOS y la web, lo cual redunda en una mayor facilidad en la portabilidad del sistema. La carpeta donde se encuentra el archivo de la aplicación móvil se puede evidenciar en el Apéndice H.

Ahora bien, referente al diseño de dicha aplicación, el flujo inicia con una pantalla inicial cuya denominación es Splash Screen; allí es posible visualizar el nombre más el ícono del proyecto; además, dicha pantalla tiene como función la presentación y carga, brindando una vista preliminar de la aplicación.

Una vez cargado el Splash Screen, en la pantalla se visualiza la primera configuración en la cual se solicita al usuario ingresar la dirección IP del servidor; dicha pantalla muestra un cuadro de texto cuyo fin es aptar la dirección IP. Además de un botón de confirmación cuya función es

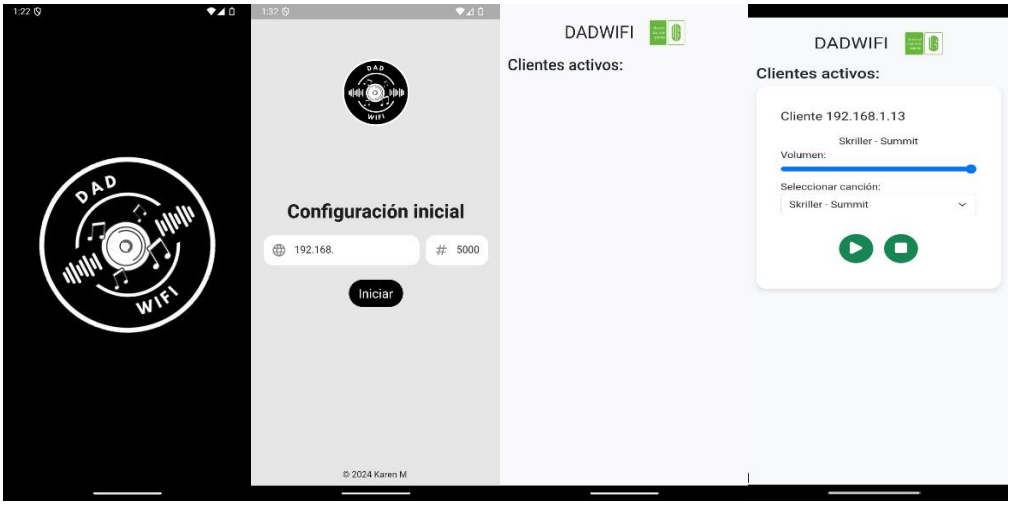
validar la información; cuando ya se ha ingresada la IP correcta, ya se establece la conexión al servicio web del servidor.

Para terminar el proceso, la interfaz carga la página principal donde se observa a los clientes que se hallen activos en ese momento; dichos clientes se presentan en forma de lista, identificados por sus respectivas direcciones IP y las condiciones de cada una de sus conexiones, en forma separada. dirección IP y estado de conexión. Es pertinente mencionar que la interfaz permite la interacción básica la cual permite seleccionar cada cliente con el fin de realizar los ajustes y/o los cambios requeridos para una buena calidad en la reproducción de audio, pudiendo manipular variables como pausa, volumen e identificación de pistas a seleccionar.

En términos generales, el grueso del diseño de la aplicación se concentra en dos aspectos clave: funcionalidad y simplicidad; estos dos aspectos son fundamentales cuando un usuario busca, de manera irreflexiva, un escenario seguro en términos de administración confiable de un sistema de audio distribuido.

Figura 9.

Interfaz móvil



Nota: Capturas del aplicativo móvil primero mostrando su logotipo, seguidamente darle la pauta de la ruta de la dirección IP lo que conduce a la interfaz donde aparecen los clientes activos.

5. Verificaciones De Rendimiento Y Fiabilidad

En esta etapa del proceso, se comprobará el rendimiento y la fiabilidad del sistema, lo cual nos garantizará un funcionamiento adecuado a la hora de implementar el producto realizado.

Como primera etapa se hace una supervisión funcional, la cual consiste en verificar la sincronización y la estabilidad del sistema en tiempo real mediante una inspección visual y auditiva por parte del usuario. Si esta fase tiene éxito se procede al aspecto teórico en donde se consigue el dato exacto del ancho de banda de la red construida con el objetivo de verificar el rendimiento del sistema y por último se hace una prueba de fiabilidad enviando datos de prueba con el fin de verificar caídas del sistema.

5.1 Pruebas de Sincronización y Estabilidad de Módulos de Audio Distribuido

Las pruebas realizadas evidencian la fiabilidad del sistema, supervisando la transmisión de audio en los diferentes módulos de manera simultánea. Se instruye el funcionamiento mediante un video en un enlace de YouTube, ver Apéndice I, donde de manera detallada, se logra evidenciar el correcto funcionamiento e implementación del sistema.

En el video se muestra la ejecución del servidor central, que en este caso está en el computador, seguida de la ejecución de los clientes. A través de la consola del servidor, se habilita la interfaz web, que permite acceder a un panel de control visualmente amigable y facilita el control del sistema.

La reproducción del audio en cada uno de los módulos se evidencia de acuerdo con las necesidades de control, lo cual se muestra en el video, ajustando el volumen de manera individual

y comprobando las funcionalidades de pausa y detención. Además, se monitorea la estabilidad de la conexión para asegurar un rendimiento óptimo.

5.2 Pruebas de rendimiento

El rendimiento consiste en la verificación de la tasa de transferencia máxima del canal permitido por la red construida, Esta máxima transferencia es conocida como ancho de banda del canal. El tipo de audio utilizado requiere para su transmisión un ancho de banda mínimo de 128Kbps. La medida tomada experimentalmente debe garantizar que la red supere el ancho de banda requerido por los 3 módulos en cumplimiento de las metas del proyecto.

El programa D-ITG, de uso libre, permite medir el ancho de banda de un canal da datos en bits por segundo mediante la implementación de un servidor que generará datos hacia un receptor. La información generada viaja desde el servidor hasta el receptor y regresará al servidor el valor de la medida.

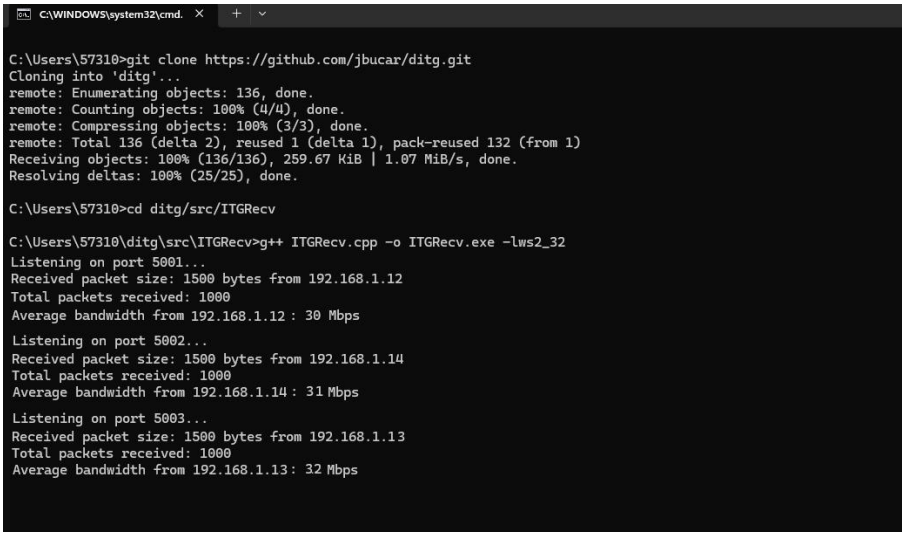
El equipo servidor utilizado será mismo computador utilizado como servidor de audio, este equipo tiene una tarjeta de red con el estándar 802.11AC el cual teóricamente nos puede generar una red de entre 400Mbps y 800Mbps según la frecuencia de conexión. Se tomará como equipo receptor la Raspberry PI la cual cuenta con la misma tarjeta 802.11AC. Las pruebas se realizaron con un router Linksys WTR54G el cual ofrece una red de 54Mbps.

Dado lo anterior se espera tener una red de 54Mbps dependiendo de la calidad del router lo cual indica que el sistema debería soportar 54Mpos / 128Kbps= 216 módulos de audio, lo cual supera exageradamente la capacidad requerida por los 3 módulos de audio usados en el proyecto.

Después de realizar las medidas con el software D-ITG se obtiene un ancho de banda real de 33Mbps lo cual nos garantiza el funcionamiento de 132 módulos.

Figura 10.

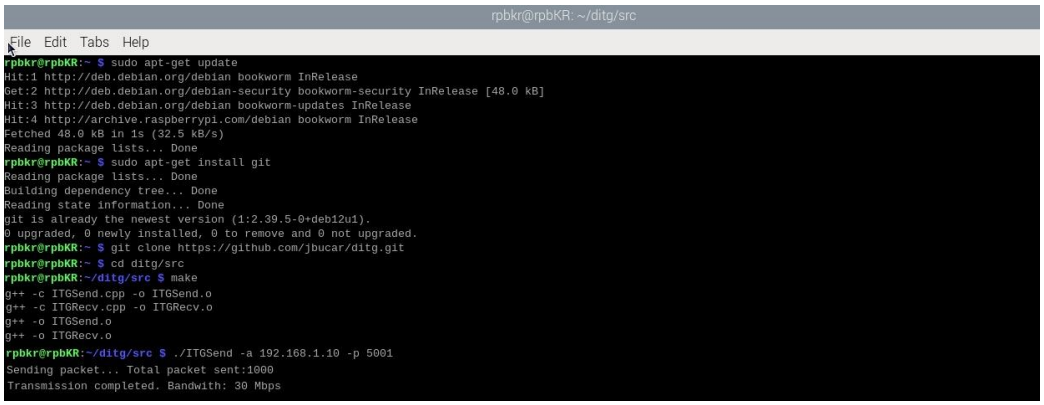
Consola del Servidor, ejecución del programa D-ITG.



Nota: Instalación del programa por medio de GitHub y ejecución del programa en los puertos 5001, 5002, 5003, para la medición del ancho de banda.

Figura 11.

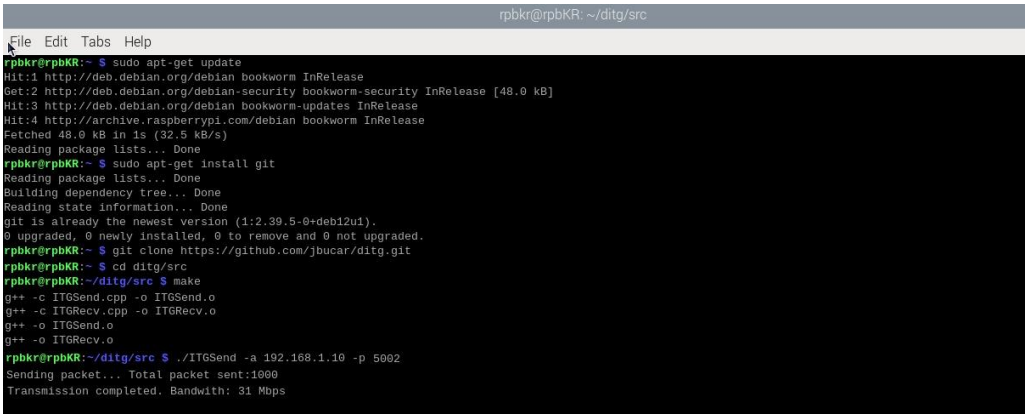
Consola del Cliente 1, Raspberry Pi, ejecución D-ITG.



Nota: Descarga del programa, ejecución y especificación del puerto 5001, medición del ancho de banda 30 Mbps.

Figura 22.

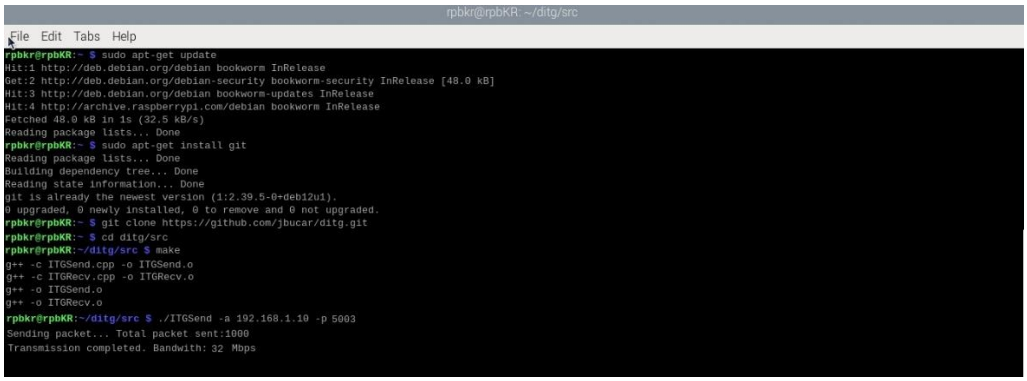
Consola del Cliente 2, Raspberry Pi, ejecución D-ITG.



Nota: Descarga del programa, ejecución y especificación del puerto 5002, medición del ancho de banda 31 Mbps.

Figura 33.

Consola del Cliente 3, Raspberry Pi, ejecución D-ITG.



Nota: Descarga del programa, ejecución y especificación del puerto 5003, medición del ancho de banda 32 Mbps.

5.3 Pruebas de fiabilidad.

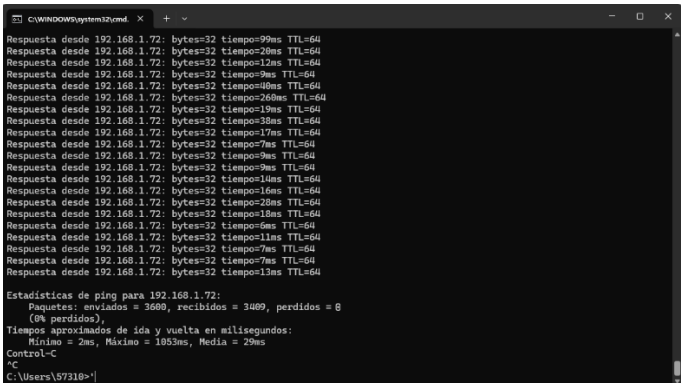
Para comprobar fiabilidad del sistema se utilizó el comando ping para comprobar durante 1 hora la conectividad entre el servidor de audio y cada una de las Raspberrys PI.

Como resultado se obtuvo:

Con la primera Raspberry se enviaron 64 paquetes y no falló ninguno.

Figura 4.

Ping en la Raspberry 1

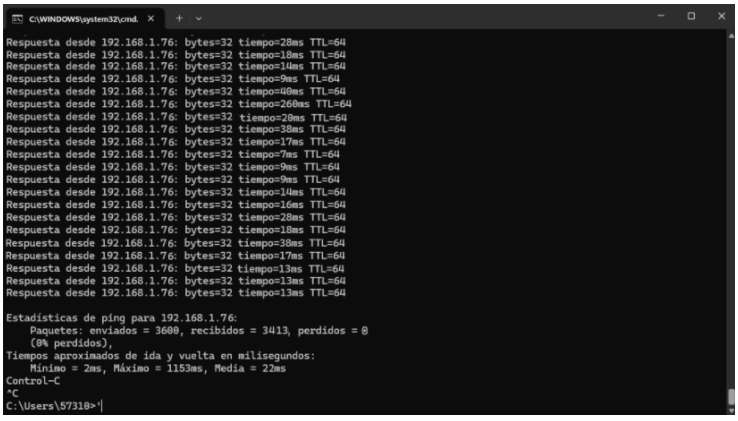


Nota: Ping mediante la terminal cmd, comprobante de conectividad Raspberry 2.

Con la segunda Raspberry se enviaron 64 paquetes y no falló ninguno.

Figura 5.

Ping en la Raspberry 2

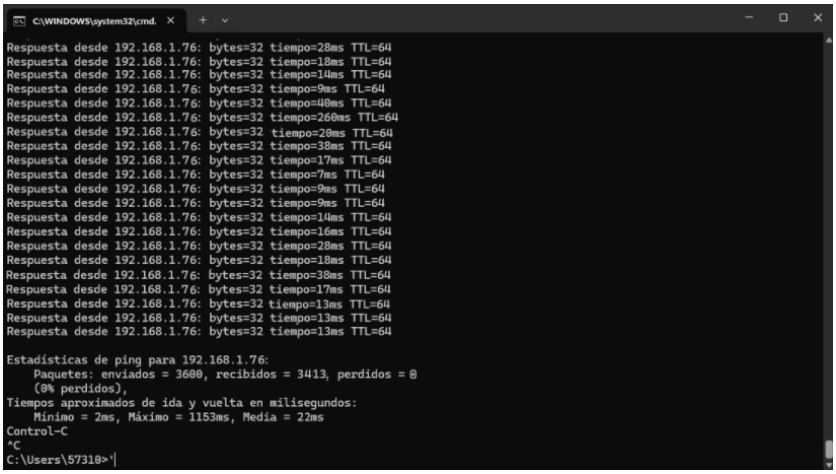


Nota: Ping mediante la terminal cmd, comprobante de conectividad Raspberry 2.

Con la tercera Raspberry se enviaron 64 paquetes y no falló ninguno.

Figura 6.

Ping en la Raspberry 3



Nota: Ping mediante la terminal cmd, comprobante de conectividad Raspberry 3.

Con este proceso se puede comprobar que la fiabilidad del sistema es del 100%pi

6. Costo Total del Sistema de Audio

Con el fin de realizar un análisis de los costos, se tuvieron en cuenta los componentes previamente seleccionados en la sección 3.3. Se utilizaron tres Raspberry PI 4, con un costo de 250,000 pesos colombianos cada una, sumando un total de 750,000 COP. El router Linksys se obtuvo por un costo de 30,000 COP, desempeñando un papel crucial en la creación de una red local eficiente. Por otro lado, se adquirieron tres bocinas Maxell por un total de 90,000 COP, además de los cables jack macho-macho, siendo tres en total por un costo de 15,000 pesos. El costo total del sistema es de 870,000 COP.

7. Conclusiones

Al momento de implementar se evidencia que la optimización de recursos en el diseño por los sistemas embebidos tiene limitaciones de hardware, ya que la capacidad de procesamiento no es tan convencional como la de un computador que cuenta con un sistema más robusto.

El uso topología de la red juega un papel importante para que la configuración de los demás dispositivos no se vea afectada, garantizando que con una red bien construida se evite la interferencia. Y así manteniendo la conectividad continua, en el caso de que un dispositivo presente fallas, no se verá afectada la red completa ni los demás dispositivos.

En el desarrollo de la programación en Python, se concluye que el código desarrollado para un cliente se puede replicar en los demás clientes, facilitando la escalabilidad en las funciones.

Al momento de ajustar los parámetros de *CHUNK* y *FORMAT* en la programación, se evidenciaron cambios en la calidad del sonido, por lo que se ajustaron para lograr una calidad óptima. Es importante que el servidor y los clientes mantengan los mismos valores para asegurar la coherencia en la transmisión de audio

8. Recomendaciones

Al implementar un sistema de audio distribuido mediante las Raspberry PI, se recomienda tener en cuenta los factores que pueden afectar la red y el rendimiento del sistema. A diferencia de un computador, que no es tan limitado y cuenta con procesadores más potentes, de mayor capacidad de memoria y almacenamiento, así como una mejor gestión de recursos, donde se requiere más cuidado en la programación del sistema.

Para un rendimiento óptimo en la programación en Python, se recomienda el uso de librerías livianas que no recarguen el almacenamiento, especialmente en las Raspberry PI, dado que este dispositivo cuenta con recursos limitados.

Referencias Bibliográficas

Baldeón Guillama, R. C. (jun-2019). Diseño e implementación de redes Wi-Fi seguras [Tesis de licenciatura, Universidad Oberta de Catalunya (UOC)]. Recuperado de http://hdl.handle.net/10609/97826

Ganssle, J. (2008). *The art of designing embedded systems*. Newnes.

Heath, S. (2002). *Embedded systems design*. Newnes.

Upton, E., & Halfacree, G. (2014). *Raspberry PI user guide*. Wiley.

Sutherland, I., Haddon, S., & Williams, S. (2016). *Prototyping with Raspberry PI: A practical guide*. MIT Press.

McCormick, R. (2017). *The impact of Raspberry PI on technology education*. Journal of Educational Technology, 23(4), 45-58.

Stallings, W. (2020). *Data and computer communications* (11.a ed.). Pearson.

Kurose, J. F., & Ross, K. W. (2021). *Computer networking: A top-down approach* (8.a ed.). Pearson.

Kumar, R., & Singh, A. (2023). An overview of WIFI technologies: Challenges and advancements. *International Journal of Computer Applications*, 175(5), 1-8. https://doi.org/10.5120/ijca2023923240

Sharma, P., Gupta, V., & Kumar, S. (2022). Wireless local area network technologies: A review. *Journal of Network and Computer Applications*, 197, 103380. https://doi.org/10.1016/j.jnca.2021.103380

Apéndices

Apéndice A. *Documentación oficial de la Raspberry PI en GitHub*

https://www.raspberrypi.com/documentation/computers/getting-started.html#set-up-your-raspberry-pi

Apéndice B. *Carpeta Servidor_RPI en GitHub.*

https://github.com/Karen01-20/servidor_RPI

Apéndice C. *Repositorio de la librería de PyAudio en GitHub.*

https://github.com/topics/pyaudio

Apéndice D. *Repositorio de la librería de PyAudio en GitHub.*

https://github.com/pallets/flask

Apéndice E. *Repositorio de la librería de Pydub en GitHub.*

https://github.com/jiaaro/pydub

Apéndice F.*Repositorio de la librería de FFmpeg en GiyHub.*

https://github.com/FFmpeg/FFmpeg

Apéndice G. *Carpeta Cliente_RPI en GitHub.*

https://github.com/Karen01-20/cliente_RPI

Apéndice H.. *Repositorio de la carpeta para la creación de la app móvil.*

https://github.com/Karen01-20/dadWIFI

Apéndice I. *Video ilustrativo de verificación en link de YouTube.*

https://youtu.be/R9bpmZQLPDI?si=OWfpU-9LI0A3fRQ4