CARACTERIZACIÓN E IMPLEMENTACIÓN DE SISTEMA DE SONIDO AMBIENTAL PARA LA EMISORA RADIO UNIVERSIDAD SURCOLOMBIANA A TRAVÉS DE RASPBERRY PI, MONITOREADO DESDE UNA APLICACIÓN MÓVIL EN LAS SEDES DE LA PLATA, GARZÓN Y PITALITO DE LA USCO.

AUTORES:

LUISA FERNANDA MÁRQUEZ PUENTES

JUAN CARLOS SILVA GUTIÉRREZ

UNIVERSIDAD SURCOLOMBIANA

FACULTAD DE INGENIERÍA

PROGRAMA DE INGENIERÍA ELECTRÓNICA

NEIVA-HUILA

2017

CARACTERIZACIÓN E IMPLEMENTACIÓN DE SISTEMA DE SONIDO AMBIENTAL PARA LA EMISORA RADIO UNIVERSIDAD SURCOLOMBIANA A TRAVÉS DE RASPBERRY PI, MONITOREADO DESDE UNA APLICACIÓN MÓVIL EN LAS SEDES DE LA PLATA, GARZÓN Y PITALITO DE LA USCO.

LUISA FERNANDA MÁRQUEZ PUENTES

Cód. 20111100612

JUAN CARLOS SILVA GUTIÉRREZ

Cód. 20111103687

Proyecto de grado presentado para optar el título de

INGENIERO ELECTRÓNICO

Director:

JOSÉ DE JESÚS SALGADO PATRÓN

Msc. Ingeniero Electrónico

UNIVERSIDAD SURCOLOMBIANA

FACULTAD DE INGENIERÍA

PROGRAMA DE INGENIERÍA ELECTRÓNICA

NEIVA-HUILA

2017

Nota de Aceptación:

_____

_____

_____

_____

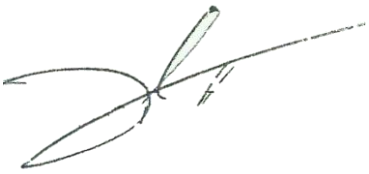
_____

_____

Firma Director de Proyecto

_____

Firma Primer Jurado



_____

Firma Segundo Jurado

Neiva, 27 de noviembre de 2017.

DEDICATORIA

Luisa Fernanda Márquez Puentes

Dedico este trabajo principalmente a mi familia: mis padres, hermanos y sobrinos por ser el motor fundamental en mi vida. Por sus enseñanzas, colaboración y amor infinito en cada paso dado.

A mis amigos que me apoyaron en todo momento y que hoy comparten mi felicidad de haber cumplido un sueño.

Y a todas las personas que conocí y ayudaron en esta parte de mi proyecto de vida.

Juan Carlos Silva Gutiérrez

A lo largo de todo este camino para obtener mi título profesional, quiero dedicárselo con cariño a todos quienes fueron fundamentales para que esto hubiese sido posible:

A Dios, ya que sin Él esto jamás habría sido posible.

A mis padres, Víctor Silva y Luz Dary Gutiérrez quienes fueron los que me dieron el don de la vida y a mi hermano Fernando Silva, en fin, a mi familia, porque sin ellos y su apoyo incondicional, habría sido prácticamente imposible que yo hubiese salido adelante con mi vida pese a muchas dificultades que pasaran.

A mis familiares y amigos que siempre creyeron en mí.

Al Ing. José de Jesús Salgado, quien fue mi director de tesis y quien fue parte fundamental para que esto fuera posible para obtener mi título de profesional.

A todos los profesores de Ingeniería Electrónica, a la secretaria de programa Sonia y a los encargados de los laboratorios por haber sido parte esencial a lo largo de estos 6 años.

AGRADECIMIENTOS

Luisa Fernanda Márquez Puentes

Agradezco inicialmente a Dios, por haberme guiado durante estos años de estudio en donde conocí y aprendí de muchas personas valiosas.
A mis padres María y Ricardo, por su amor, apoyo incondicional y entrega que me forjaron como persona durante mi carrera. A mis hermanos Diego y Óscar, quienes me inspiran a ser mejor cada día sin importar la distancia.
A Juan por su amor y comprensión en todo este proceso y al resto de familia, amigos y compañeros de estudio que me alentaron para culminar con éxito esta etapa de mi vida.

A nuestro director de tesis, el Ing. José Salgado Patrón, por su empeño para que este proyecto se diera. Al director de la emisora institucional, Óscar Forero Mosquera y a su grupo de trabajo por su gestión y colaboración durante todo el proceso. Finalmente, a todo el grupo de trabajo del programa de Ingeniería Electrónica, desde docentes, secretaria hasta laboratoristas, por su continua enseñanza, apoyo y servicio.

Juan Carlos Silva Gutiérrez

Primero que todo agradecer a Dios por haberme dado la oportunidad de vivir, de tener salud, de apoyarme y darme todas sus fuerzas y bendiciones a lo largo de todo este tiempo para que yo pudiese culminar mis estudios.

A mi familia: mi papá, a mi mamá, y a mi hermano, quienes fueron los pilares y el motor de mi vida, por su cariño y amor desde el principio hasta el final, por su lucha constante, por su apoyo incondicional pese a todas las dificultades que se presentaban y por todos los sacrificios que tuvieron que pasar para que yo lograra obtener mi título profesional.

A mis familiares por su lucha y apoyo constante a la distancia, a todos mis compañeros de la universidad quienes fueron un gran apoyo a lo largo de estos 6 años quienes siempre estuvieron en las buenas y en las malas.

A mi director de tesis, Ing. José de Jesús Salgado por su constante apoyo en todo momento, al director de la emisora de la USCO, Oscar Forero por su paciencia en el proyecto, a los encargados de los laboratorios: don Carlos y don Eduardo y a todos los profesores de la facultad de Ingeniería Electrónica y demás docentes por ser haber sido parte de mi formación como persona y como profesional.

CONTENIDO

Pág.

LISTA DE TABLAS

Pág.

LISTA DE FIGURAS

Pág.

LISTA DE ANEXOS

Pág.

GLOSARIO

ARM: arquitectura de 32 bits usada en computadores con sistema de instrucciones reducidos.

API: (Application Programming Interface), interfaz de programación de aplicaciones.

ANDROID: sistema operativo basado en Linux.

CSS: lenguaje de hojas de estilos creado para controlar el aspecto de páginas web.

HTML: lenguaje de marcas de hipertexto para la elaboración de páginas web.

JAVASCRIPT: lenguaje de programación de uso principalmente para desarrollo de páginas web dinámicas.

PYTHON: lenguaje de programación de alto nivel, interpretado y multipropósito.

RASPBERRY PI: mini ordenador sin periféricos guiado a la enseñanza de programación.

RASPBIAN: sistema operativo Linux para Raspberry basado en Debian.

SBC: (Single Board Computer), computador de placa reducida.

SCRIPT: conjunto de órdenes guardadas en un archivo de texto, que es ejecutado por lotes o línea a línea, en tiempo real por un intérprete.

SDK: (Software Development Kit), grupo de herramientas que permiten la programación de aplicaciones móviles.

SOC: (System on a chip), tipo de dispositivos que integra en un solo chip los diferentes componentes de un sistema.

RESUMEN

Este proyecto hace parte de la ampliación de cobertura de la emisora institucional a partir de la implementación del sistema de sonido ambiental, propuesto para los distintos sectores de las sedes de Garzón, Pitalito y la Plata de la Universidad Surcolombiana.

El sistema propuesto está compuesto de dos partes:

- La programación para trasmisión y recepción de datos, en este caso, el streaming de la emisora; realizada en la tarjeta programable Raspberry Pi 3, manejando el lenguaje de programación Python y la conexión a la base de datos en tiempo real: Firebase. Monitoreado a través de la aplicación móvil diseñada en Android 4.2 y la aplicación web estática realizada con HTML, JavaScript y CSS.

- La implementación física del sistema de sonido, con módulos independientes que contienen fuentes de poder que van desde los 12VDC a los 24VDC, amplificadores de baja (8W), media (50W) y alta potencia (100W), dependiendo el espacio a sonorizar, bafles ambientales y la tarjeta Raspberry Pi 3. Cada módulo cuenta con una caja ajustada a las necesidades eléctricas y se dejaron instaladas según la cobertura de red wifi y conexión eléctrica del sector en cada sede.

El resultado final del sistema de sonido ambiental ofrece un sistema robusto dependiente de una buena conexión a internet de manera inalámbrica para proporcionar un servicio de calidad en tiempo real, el cual transmite el streaming de la emisora institucional logrando crear escenarios de comunicación entre la comunidad estudiantil y directivos de las sedes con el resto de la universidad.

Palabras claves: *Transmisión de datos, sonido ambiental, app móvil, tarjeta programable, app web.*

Abstract

This project makes part of the coverage expansion by the institutional radio from the implementation of an environmental sound system, proposed by many sectors of Garzón, Pitalito and La Plata at Surcolombiana University.

The proposed system is composed of two parts:

- The programming for transmission and reception of data, in this case, the streaming of the radio; made on the Raspberry Pi 3 programmable board, handling the Python programming language and the connection to the database

in real time: Firebase. Monitored through the mobile application designed in Android 4.2 and the static web application made with HTML, JavaScript and CSS.

- The physical implementation of the sound system, with independent modules containing power supplies ranging from 12VDC to 24VDC, low (8W), medium (50W) and high power (100W) amplifiers, depending on the space to be sonicated, environmental speakers and Raspberry Pi 3 board. Each module has a box adjusted to the electrical needs and is installed according to the network coverage and electrical connection of the sector in each seat.

The result of the environmental sound system offers a robust system dependent on a good wireless internet connection to provide quality service in real time, which transmits the streaming of the institutional radio, creating communication scenarios between the student community and directors of the seat with the rest of the university.

Key Words: *Data transmission, environmental sound, mobile app, programmable board, web app.*

INTRODUCCIÓN

La humanidad ha estado en constante evolución, cambiando su forma de pensar, de actuar y de estar bien informado de sucesos que ocurren no solo dentro de la comunidad, sino a nivel regional y nacional. Esto ha llevado al hombre a crear nuevas formas de mantener informados a sus semejantes, desarrollando diferentes tecnologías para brindar y narrar acontecimientos a través de diferentes medios de comunicación; hechos que día a día ocurren en las diferentes ciudades, pero que muchas personas cuyas residencias están más apartadas, no tienen la opción de enterarse, ni tampoco de expresar su opinión o sus problemáticas a nivel de comunidad.

A nivel institucional, la Universidad Surcolombiana ha querido promover la emisora oficial ubicada y transmitida en la ciudad de Neiva a sus respectivas sedes situadas en los municipios de La Plata, Garzón y Pitalito a través de un sistema de sonido ambiental, brindando a los más de 2000 estudiantes (cifra estimada de la base de datos), la oportunidad de fortalecer escenarios de comunicación que permitan no solo estar informados de los hechos que ocurren en la región, sino de generar una opinión publica de diversidad cultural.

En un principio la señal originada en la ciudad de Neiva no abarcaba esos municipios del centro y sur del Huila; aunque existían ideas para solucionar la problemática, resultaba un poco costosa y de algunos años de gestión. Sin embargo, gracias al avance de la tecnología y evitando el uso del espectro electromagnético por donde se dispersa la señal de radio, sino mediante la utilización de la señal de streaming disponible que tiene la emisora, dicha señal se programa para ser ejecutada desde un dispositivo programable a cada una de las sedes a través de la red wifi, logrando que toda la comunidad educativa sea testigo de primera mano de los sucesos y noticias que pasan dentro y fuera de la Universidad Surcolombiana.

La solución contempla además el desarrollo de un centro de monitoreo que permita la activación o desactivación del volumen de cada sistema de sonido, de manera general e independiente en cada uno de los módulos ubicados en distintos sectores de las sedes, a través de un dispositivo que todo el mundo tenga a la mano, es decir, desde un teléfono inteligente gracias a una aplicación móvil, logrando una mejor estabilidad y comodidad para acceder al centro de monitoreo cuando se requiera, al momento que desee y por supuesto, de manera económica.

1. MARCO TEÓRICO

1.1 SISTEMAS DE SONIDO AMBIENTAL

Se habla de la música ambiental desde el fenómeno *Muzak* dado a mediados del siglo XX en los Estados Unidos, en donde se ofrecía un tipo de música programada que ayudaba al rendimiento de quién la escuchaba, ya sea como empleado o posible cliente.

En realidad el término proviene de la empresa Muzak Corporation ahora llamada Mood, la cual ofrecía en la época, servicios de música por cable a distintas empresas y que decidió incursionar en el tema con sus investigaciones, dando lugar a la técnica llamada *Estimulación progresiva*, en donde se emitían intervalos cortos de música sin letra (música para ser oída más no escuchada) entre sus empleados, con variación de ritmo e instrumentación, aumentando los índices de productividad al acrecentar su concentración mental y bajar sus niveles de fatiga[1].

Teniendo clara la influencia de la música ambiental en el entorno de la sociedad, los sistemas de sonido ambientales son la estructura base y se definen como dispositivos electroacústicos conectados entre sí, que permiten sonorizar pasiva o activamente cualquier espacio, con una gestión de funciones que están relacionadas con el sonido desde un mando de control ubicado en una o varias instancias.

Dentro de los componentes de una instalación de sonido ambiental se encuentra:

- *Central de Mando:* Es la unidad madre de un sistema de audio, la cual trabaja con la señal a reproducir y alimenta al resto de elementos que componen la instalación de sonido ambiental.

- *Fuente de Audio:* Es el dispositivo o medio que proporciona la señal a reproducir para ser escuchada por los altavoces dispuestos en las zonas. Normalmente pueden ser reproductores de CD/DVD, computadores portátiles, emisoras, radio por internet, etc.

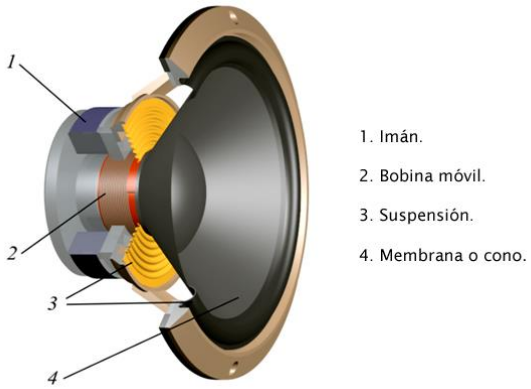
- *Amplificadores de audio*: Dispositivo en el que una pequeña cantidad de energía de entrada es transformada en una gran cantidad de energía de salida. Su función es incrementar eléctricamente señales de audio de bajo nivel de voltaje, y enviarlas a las redes de altavoces con el mínimo de distorsión y el máximo de eficiencia.

---

[1] Funding Universe. *Company Histories [En línea].*

- *Altavoces:* Los altavoces son transductores electroacústicos utilizados para la reproducción de sonido, ellos transforman la energía eléctrica proveniente del aplicador en energía acústica (sonido). Normalmente se colocan en múltiplos de 2 unidades, para que el audio sea en estéreo.

Figura 1. Partes de un altavoz dinámico



Fuente: Cosas de audio.

Propagación del sonido

Para que una onda sonora se propague en un medio, éste debe cumplir como mínimo tres condiciones fundamentales: ser elástico, tener masa e inercia.

Las ondas sonoras no se propagan en el vacío, lo hace en el aire y éste posee características relevantes para la propagación del sonido:

- La propagación es ***lineal***, que quiere decir que diferentes ondas sonoras (sonidos) pueden propagarse por el mismo espacio al mismo tiempo sin afectarse mutuamente.
- Es un medio ***no dispersivo***, por lo que las ondas se propagan a la misma velocidad independientemente de su frecuencia o amplitud.
- Es también un medio ***homogéneo***, de manera que el sonido se propaga esféricamente, es decir, en todas las direcciones, generando lo que se denomina un campo sonoro.

La velocidad de propagación de la onda sonora o velocidad del sonido depende de las características del medio en el que se realiza dicha propagación y no de las características de la onda o de la fuerza que la genera.

Pero la velocidad del sonido sí varía ante los cambios de temperatura del aire, cuanto mayor es la temperatura del aire, mayor es la velocidad de propagación; la

velocidad del sonido en el aire aumenta 0,6 m/s por cada 1º C de aumento en la temperatura[2].

1.2 TRANSMISIÓN DE DATOS

Desde cualquier punto de vista la trasmisión de datos con la mejor fidelidad ha sido parte esencial de las comunicaciones y existen diversas maneras que aseguran el objetivo dependiendo el tipo de señal, medio de transmisión, etc. Para el proyecto en específico se tiene como eje central las placas de desarrollo o tarjetas programables, dichos sistemas permiten la interacción entre datos, recibiendo y visualizando al usuario, haciendo la tarea de trasmisión algo bastante accesible. Además, brindando ventajas sobre otros sistemas gracias a su portabilidad, bajo coste y comprensible programación.

1.2.1 Placas de desarrollo. Ordenador de placa reducida (en inglés: Single Board Computer o SBC) es una computadora completa en un sólo circuito. El diseño se centra en un sólo microprocesador con la RAM, E/S y todas las demás características de un computador funcional en una sola tarjeta.

Muchas de las SBC son tan poderosas que han llegado a tener la capacidad de una tablet o pc de hoy en día. Pero para llegar a ello se recorrió un largo camino que parte desde la llamada “dyna-micro”* en los años 70.

Actualmente, existen dos categorías de SBC en el mercado: los open source y los de propiedad. Difieren entre ellos por la disposición final que se le da al hardware y software que los compone, ya que en los de propiedad se entrega un producto final testeado en anterioridad mientras que en los open source ambas partes se dan al usuario para que interactúe y haga sus propios ajustes.

1.2.2 Herramientas de programación: Software y Lenguajes. La forma más frecuente de software utilizado en SBCs es Linux con numerosas derivaciones como Android, Ubuntu, Fedora, Debian y Arch Linux, así como FreeBSD y Windows CE.

En cuanto a las herramientas de programación / depuración suelen ser gratuitas y de código abierto, como las basadas en el IDE de Eclipse y para los lenguajes,

---

[2] Maggiolo, D. *Propagación del sonido* [En línea].

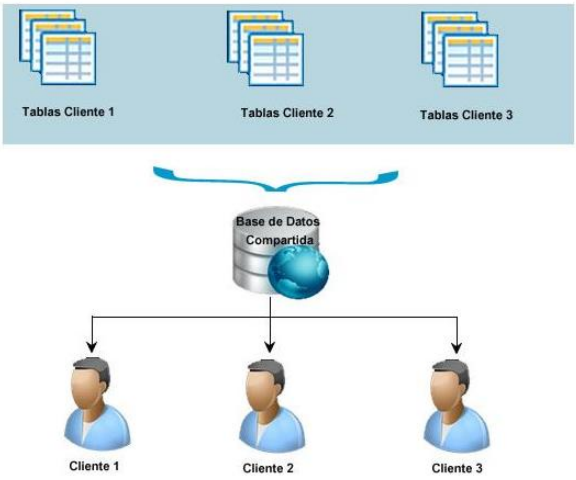
* El "dyna-micro" / "MMD-1" fue el primer ordenador de placa reducida verdadero del mundo, tenía todos los componentes en una sola placa de circuito impreso, incluyendo memoria, E / S, dispositivo de entrada del usuario, y una pantalla.

dependen del fin de la programación, entre ellos los más destacados: Python, JavaScript, C #, C, Ruby, PHP, Java o C ++[3].

1.2.3 Bases de datos. Es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Existen programas denominados sistemas gestores de bases de datos, abreviado SGBD (del inglés database management system o DBMS), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada.

Figura 2. Estructura de una base de datos



Fuente: Google Imágenes.

Existen principalmente dos tipos de bases de datos: SQL y las NoSQL. A parte de estos dos tipos, aparecen las bases de datos híbridas como por ejemplo SQL/NoSQL, bases de datos in-memory o bases de datos as a service.

- *Base de datos relacionales SQL:* Es una base de datos donde todos los datos se almacenan y se accede a ellos por medio de relaciones previamente establecidas.

- *Base de datos NoSQL*: Suelen ser bases de datos mucho más abiertas y flexibles. Permiten adaptarse a necesidades de proyectos, se pueden hacer cambios de los esquemas sin tener que parar las bases de datos.

[3] Ortmeyer, C. *A Brief History of Single Board Computers*. Electronic Desing Uncovered [En línea]. 2014. p. 1-2.

- *Base de datos híbridas:* Como su nombre lo indica, utilizan varios motores de bases de datos para dar cabida a varios modelos NoSQL e incluso algunos relacionales.

1.3 APLICACIONES MÓVILES

En esencia, una aplicación no deja de ser un software. Entendiendo un poco mejor el concepto, se puede decir que las aplicaciones son para los móviles lo que los programas son para los ordenadores de escritorio.

El proceso de diseño y desarrollo de una aplicación abarca desde la concepción de la idea hasta el análisis posterior a su publicación en las tiendas. Desde la perspectiva del diseño y desarrollo se han resumido las fases del proceso en:

Figura 3. Fases de diseño y desarrollo de una aplicación móvil.

Fuente: Cuello, J., & Vittone, J. Diseñando apps para móviles.

Modelo Vista Controlador

Es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control; se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones y sobre

multitud de lenguajes y plataformas de desarrollo dividido en tres componentes distintos:

El Modelo: contiene una representación de los datos que maneja el sistema. El modelo es el responsable de:

- Acceder a la capa de almacenamiento de datos.
- La funcionalidad del sistema.
- Llevar un registro de las vistas y controladores del sistema.

La Vista o interfaz de usuario: compone la información que se envía al cliente y los mecanismos interacción con éste. La vista es el responsable de:

- Recibir datos del modelo y las muestras al usuario.
- Pueden dar el servicio de "Actualización", para que sea invocado por el controlador o por el modelo.

El Controlador, actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno. El controlador es responsable de recibir los eventos de entrada.

En cuanto a la programación se encuentra un lenguaje universal a las aplicaciones móviles: el lenguaje Java.

Figura 4. Logotipo oficial de Java

Fuente: Oracle.

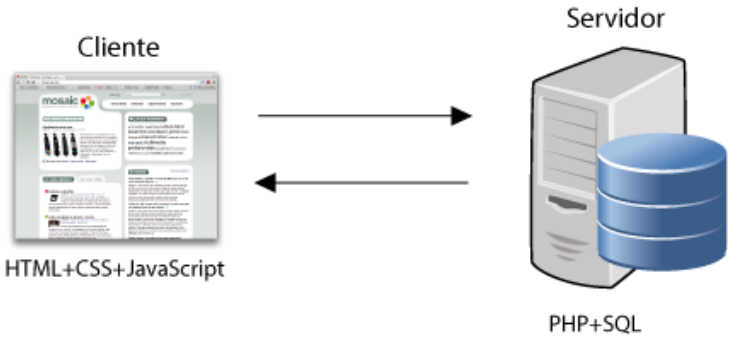
Java es un lenguaje de programación orientado a objetos, cuyo principal objetivo es permitir que una vez creado el programa se pueda ejecutar en cualquier plataforma. Como cualquier lenguaje de programación, el lenguaje Java tiene su propia estructura, reglas de sintaxis y paradigma de programación.

El lenguaje Java es un derivado del lenguaje C, por lo que sus reglas de sintaxis se parecen mucho a C.

1.4 APLICACIONES WEB

En sí, una aplicación web es un tipo especial de aplicación cliente/servidor, donde tanto el cliente (el navegador, explorador o visualizador) como el servidor (el servidor web) y el protocolo mediante el que se comunican (HyperText Transfer Protocol (HTTP)) están estandarizados y no han de ser creados por el programador de aplicaciones.

Figura 5. Esquema básico de una aplicación web



Fuente: Universitat Oberta de Catalunya.

Una aplicación Web consta de varias sub-aplicaciones, como lo son:

- Front-end: un sitio web con acceso público para usuarios finales normales.
- Back end: un sitio web que expone la funcionalidad administrativa para administrar la aplicación. Esto se limita generalmente al personal administrativo.
- Consola: una aplicación que consta de comandos de consola para ejecutar en una ventana de terminal o como trabajos programados para dar soporte a toda la aplicación.
- Web API: proporciona interfaces a terceros para integrarse con la aplicación[4].

La base de programación de las aplicaciones web es el HTML, juntamente con JavaScript y CSS.

HTML

Es un lenguaje que pertenece a la familia de los "lenguajes de marcado" y es utilizado para la elaboración de páginas web. Es un lenguaje de marcado descriptivo que se escribe en forma de etiquetas para definir la estructura de una página web y su contenido como texto, imágenes, entre otros. El estándar HTML lo define la W3C

[4] *Fundamentals: Best MVC Practices.* The Definitive Guide to Yii. Yii PHP Framework [En línea]. 2015.

(World Wide Web Consortium) y actualmente HTML se encuentra en su versión HTML5.

CSS

CSS es el acrónimo de Cascading Style Sheets (es decir, hojas de estilo en cascada). Es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Por ejemplo, CSS abarca cuestiones relativas a fuentes, colores, márgenes, líneas, altura, anchura, imágenes de fondo, posicionamiento avanzado y muchos otros temas.

CSS3 es la última evolución del lenguaje de las Hojas de Estilo en Cascada.

JavaScript

JavaScript es el lenguaje interpretado orientado a objetos desarrollado por Netscape que se utiliza en millones de páginas web y aplicaciones de servidor en todo el mundo.

En pocas palabras, JavaScript es un lenguaje de programación dinámico que soporta construcción de objetos basado en prototipos. La sintaxis básica es similar a Java y C++ con la intención de reducir el número de nuevos conceptos necesarios para aprender el lenguaje.

Figura 6. Logotipos de HTML, JS y CSS.

Fuente: Google Imágenes.

2. DESARROLLO DEL PROYECTO

En este capítulo se describen paso a paso los procesos llevados a cabo para programar e implementar el sistema de sonido ambiental desde cero, partiendo desde la caracterización de cada componente seleccionado hasta el desarrollo del software correspondiente para la transmisión y monitoreo de los datos (streaming[**] de la emisora) desde una aplicación móvil y web, con el objetivo de brindar un sistema robusto, logrando la expansión de la emisora institucional en las demás sedes de la universidad.

2.1 FACTIBILIDAD TECNOLÓGICA

Para el proyecto, se hace necesario evaluar las tecnologías en desarrollo de software existentes en el mercado, teniendo en cuenta prestaciones y costos. Partiendo desde la trasmisión de datos se realiza un estudio comparativo con las placas de desarrollo, los lenguajes de programación y las bases de datos.

2.1.1 Comparativa de placas de desarrollo. Dentro de la variedad existente en el mercado de las SBC, el enfoque ha sido guiado a 4 de las más reconocidas por la comunidad de programadores: Raspberry Pi, BeagleBone, CubieBoard y Odroid. En la Tabla 1 se observa sus características más relevantes y se realiza la comparativa entre ellas para encontrar la que más se ajuste a la necesidad de transmitir el streaming de la emisora monitoreado a través de las aplicaciones móvil y web.

[**] Es un método de transmisión de un archivo de medios en un flujo continuo de datos que pueden ser procesados en la computadora receptora antes de que se haya enviado todo el archivo.

Tabla 1. Placas de desarrollo y sus características

		Placas de desarrollo			
		Raspberry Pi 3 Modelo B	**BeagleBone Black**	**CubieBoard 2 (A20)**	**Odroid-C2**
Descripción		La RasPi 3 es un computador de placa reducida con conectividad Wifi y Bluetooth que puede ser utilizada para varias aplicaciones en desarrollo de software, es la tercera generación de la fundación Raspberry.	BeagleBone Black es una plataforma de desarrollo-comunidad de bajo costo para desarrolladores y aficionados. Arranca Linux en menos de 10 segundos y comienza el desarrollo en menos de 5 minutos con un solo cable USB.	La Cubieboard 2 es un computador de tamaño reducido, que se conecta a un televisor o monitor, a un teclado y a un mouse. Puede ser utilizado para muchas de las tareas como manejo de hojas de cálculo, edición de textos y juegos. También reproduce vídeo de alta definición.	El ODROID-C2 es un computador de placa reducida (SBC) que puede funcionar como un decodificador de cine en casa, una computadora de uso general para navegar por la web, juegos y socializar, un dispositivo de creación de prototipos para el ajuste de hardware, un controlador para domótica, una estación de trabajo para software y mucho más.
Características	CPU	4 × ARM Cortex-A53	ARM Cortex-A8	2 x ARM Cortex-A7	4 x ARM Cortex-A53
	CPU Speed	1.2 GHz	1 GHz	1 GHz	1.5 GHz
	GPU	Broadcom VideoCore IV	PowerVR SGX530	Mali-400MP2	Mali-450MP3
	RAM	SDRAM 1GB	DDR3 512MB	DDR3 1GB	DDR3 2GB
	Conexión de red	Conector Ethernet 10/100 LAN inalámbrica 802.11n Bluetooth 4.1	Conector Ethernet 10/100 Wifi 802.11b/g/n Bluetooth	Conector Ethernet 10/100	10/100/1000 Gigabit Ethernet
	USB 2.0	4 puertos	2 puertos mini (cliente) 2 puertos (servidor)	2 puertos (servidor) 1 puerto mini USB OTG	4 puertos 1 puerto USB OTG
	HDMI	✓	Micro HDMI	✓	✓
	Otras opciones de visualización	RCA Compuesto	-	LVDS, VGA	-
	Audio	HDMI Jack 3.5mm	Micro HDMI	HDMI Jack 3.5mm Spdif-out por pines auxiliares	HDMI
	Puertos GPIO/MISC	27 pines GPIO Bus I²C Bus SPI UART, CSI	66 pines GPIO Bus I²C Bus SPI CAN, UART	96 pines: I²C, SPI, RGB/LVDS UART, IR AudioIn	40 pines: UART, SPI, I²C ADC IR
	OS Soportado	Linux Android Windows 10 IOT Risc OS LibreElec	Linux Android FreeBSD Symbian Windows CE Risc OS	Linux Android	Linux Android
	Tamaño (LxW)	85 x 56 mm	87 x 56 mm	100 x 60mm	85 x 56 mm
	Alimentación	Micro USB 5V@2.5A	Micro USB 5V@2A	Micro USB 5V@2A	Micro USB 5V@2A
	Consumo	700mA (3,5W) - 800 mA (4.0 W)	460mA (2.3W máx.)	800mA (4 W)	800mA (4 W)
Costo		**$ 207.000**	**$ 377.900**	**$ 199.400**	**$ 272.900**

Fuente: Autor.

Los criterios de comparación están guiados hacia: conexión a red wifi, capacidad de procesamiento, OS y costo.

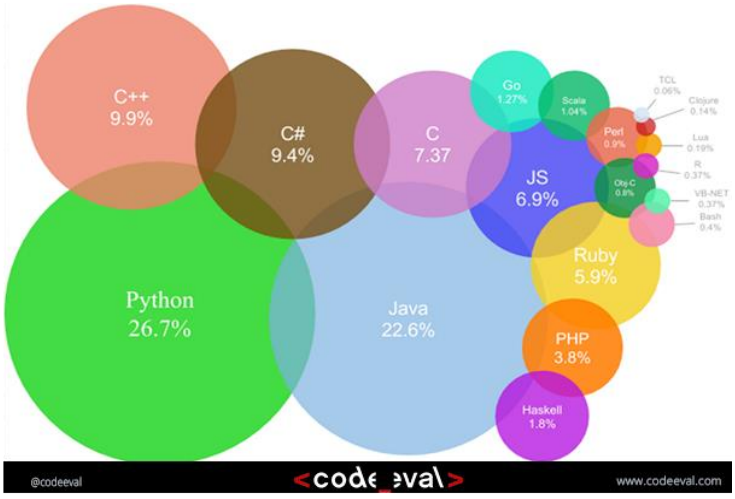
- *Conexión de red:* Las Raspberry Pi 3 y la BeagleBone son las únicas que cuentan con módulo wifi integrado. Para las otras sería necesario adquirir un adaptador de red wifi aparte.
- *Capacidad de procesamiento:* La Raspberry Pi 3 y la Odroid-C2 cuentan con buenos procesadores.
- *OS:* Afortunadamente todas las SBC cuentan con soporte a distribuciones Linux, lo que facilita su programación y permite implementar dentro de ellas cualquier lenguaje de alto nivel.
- *Costo:* La más económica es la CubieBoard 2, pero está limitada en cuanto a conexión inalámbrica y la sigue la Raspberry Pi 3.

Una vez realizada la comparativa se toma la decisión de trabajar con la SBC de la fundación Raspberry debido a sus buenas prestaciones por su módico precio y su respaldo e información disponible para programar, cubriendo las necesidades planteadas para el proyecto.

2.1.2 Comparativa de lenguajes de programación. Existen infinidad de lenguajes de programación según sus prestaciones, nivel de aprendizaje y desarrollo que tenga dentro de la comunidad de desarrolladores.

Según CodeEval, una exclusiva comunidad de más de 69.000 desarrolladores competitivos, desde el año 2013 hasta el 2016, el lenguaje de programación predilecto por miles de desarrolladores ha sido Python, ocupando el primer lugar en esos años, seguido de Java y C++.

Figura 7. Lenguajes de programación populares en el año 2016.



Fuente: CodeEval.

De modo que se toma a Python como el lenguaje central para programar en la placa de desarrollo escogida, con el fin de lograr la trasmisión del streaming de la emisora dentro del sistema de sonido ambiental propuesto.

Python

Figura 8. Logotipo oficial de Python

Fuente: Python.org

Se trata de un lenguaje de programación multiparadigma, esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional. Pues su filosofía de diseño está enfocada en la productividad del programador y legibilidad del código.

Además, es desarrollado bajo una licencia open-source, por lo que puede ser usado en cualquier sistema con total libertad, incluso con fines comerciales.

2.1.3 Comparativa de bases de datos. Para el proyecto, la monitorización de los datos es prioritaria y una vez revisado los pros y contra de estos tipos de base de datos, se opta por tomar una NoSQL o hibrida en último caso. Debido a que se dicho tipo de base de datos favorece a la hora de captura y procesado de eventos con los datos o variables almacenadas.

A través de la página de Base de Conocimientos de los Sistemas de Gestión de Bases de Datos Relacionales y NoSQL (DB-Engines) se han tomado 3 distintas bases de datos a comparar: *MongoDB*, *Amazon DynamoDB* y *Firebase.*

Dentro de la comparación realizada que puede ser observada en: https://db-engines.com/en/system/Amazon+DynamoDB%3BFirebase+Realtime+Database%3BMongoDB , se resalta:

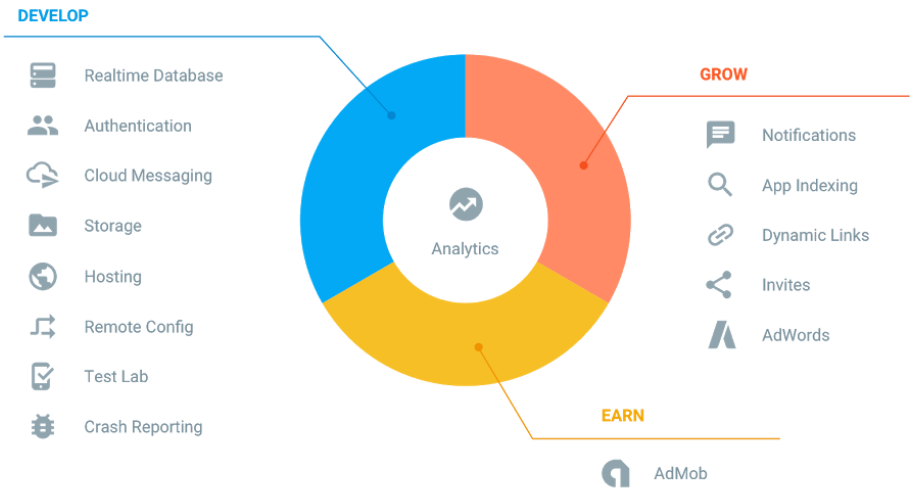
- Todas utilizan autenticación a través de reglas que se pueden definir.
- MongoDB soporte más de 20 lenguajes de programación mientras que Amazon DynamoDB abarca 10 y Firebase apenas 3.
- Todas cuentan con esquema de datos.
- Firebase cuenta con APIs específicas como Android, iOS y Javascript.

Una vez vista la comparación, las 3 bases de datos estarían en condiciones de cumplir con lo que se quiere, pero llama la atención la relación de Firebase con su compañero de Google, Android, evento que es atractivo para lograr compaginar de manera más fácil la base de datos directamente con la App móvil y web.

Firebase

Es una plataforma de Google, cuya principal función es desarrollar y facilitar la creación de aplicaciones de elevada calidad de una forma rápida, con el fin de que se pueda aumentar la base de usuarios y ganar más dinero. La plataforma está subida en la nube y está disponible para diferentes plataformas como iOS, Android y web.

Figura 9. Subproductos de Firebase



Fuente: Firebase

A la hora de gestionar y aprender a usar la plataforma, Firebase ofrece documentación diversa de muy buena calidad, mediante ejemplos, tutoriales y documentación complementaria sobre todos sus usos.

2.2 SITIO DE TRABAJO

Para cada una de las sedes se analizaron y escogieron ciertos puntos en donde se pudiera implementar el sistema de sonido ambiental, buscando las mejores condiciones de trasmisión del sonido, asegurando buena conexión a Internet y sin interrumpir otras actividades dentro de la universidad.

En cuestiones de redes, en la Sede de La Plata, Garzón y Pitalito, según el Centro de Tecnologías de Información y Comunicaciones (CTIC), se cuenta con un canal de acceso a Internet de 50Mbps prestado por la empresa Telefónica Telecom distribuidos mediante routers instalados por cada una de las sedes.

La Plata

La sede de La Plata se encuentra en el Km 1 vía a Fátima. Estructuralmente cuenta con cinco bloques de un piso. Además, cuenta con zona de parqueadero, campo deportivo y buenas zonas verdes.

Figura 10. Mapa satelital de la planta física de la sede de La Plata

Fuente: Google Maps

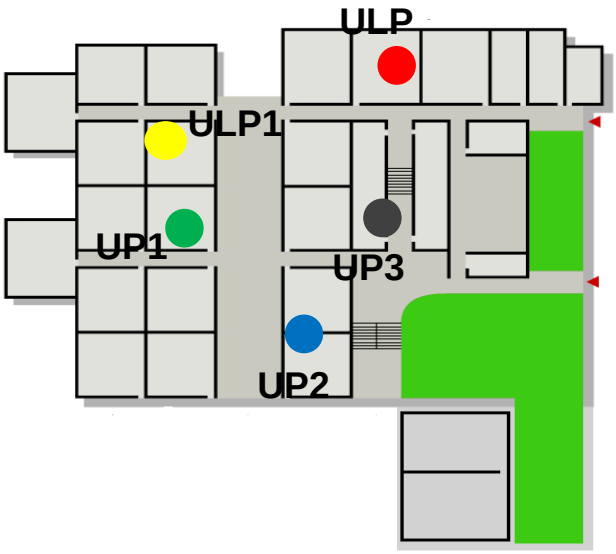
Redes inalámbricas

Dentro de la sede de La Plata, se encuentran instalados cinco (5) routers funcionales con conexión de red abierta de marca Hewlett-Packard (HP).

A continuación, se presenta el listado de las redes encontradas:

- UP1: USCO_LAPLATA1
- UP2: USCO_LAPLATA2
- UP3: USCO_LAPLATA3
- ULP: USCO_LAPLATA
- ULP1: USCOC_LAPLATA

Figura 11. Mapa de la sede con las redes wifi detectadas



Fuente: Universidad Surcolombiana

Garzón

La sede de Garzón se encuentra ubicada en la vía a las Termitas. En su estructura cuenta con tres bloques. A parte cuenta con una cafetería, un campo deportivo techado y zona de parqueadero.

Figura 12. Mapa satelital de la sede de Garzón

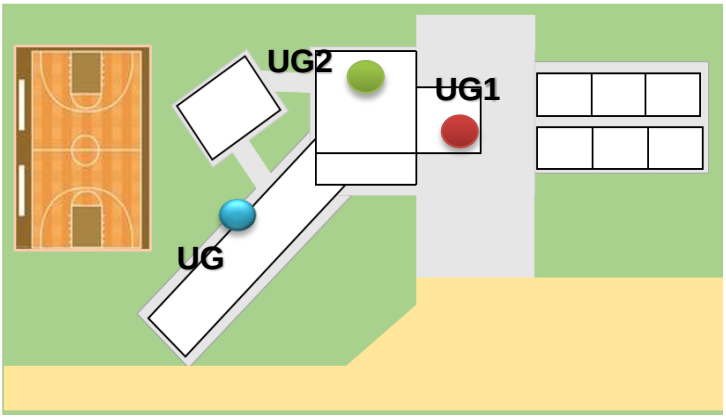
Fuente: Google Maps.

Redes inalámbricas

Dentro de la sede de Garzón, se encuentran instalados tres (3) routers funcionales. Los nombres de las redes identificadas son:

- UG: USCO_GARZON
- UG1: USCO_GARZON1
- UG2: USCO_GARZON2

Figura 13. Mapa de la sede con sus redes wifi identificadas

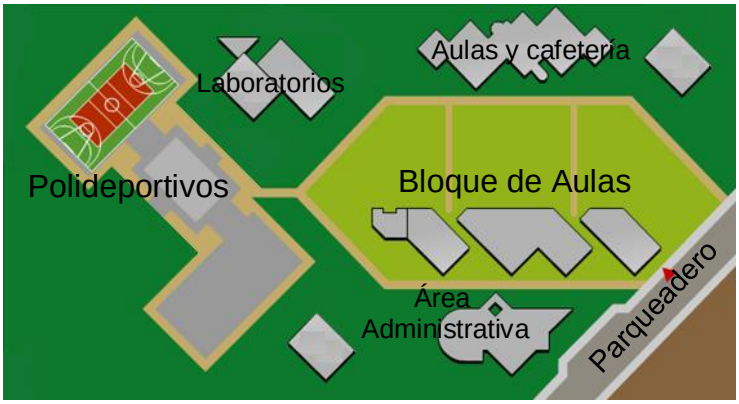


Fuente: Autor.

Pitalito

La sede universitaria de Pitalito se encuentra en el Km 1 vía a la vereda El Macal. Dentro de su estructura física cuenta con cuatro bloques, amplias zonas verdes, campos deportivos y parqueadero.

Figura 14. Mapa de la planta física de la sede de Pitalito



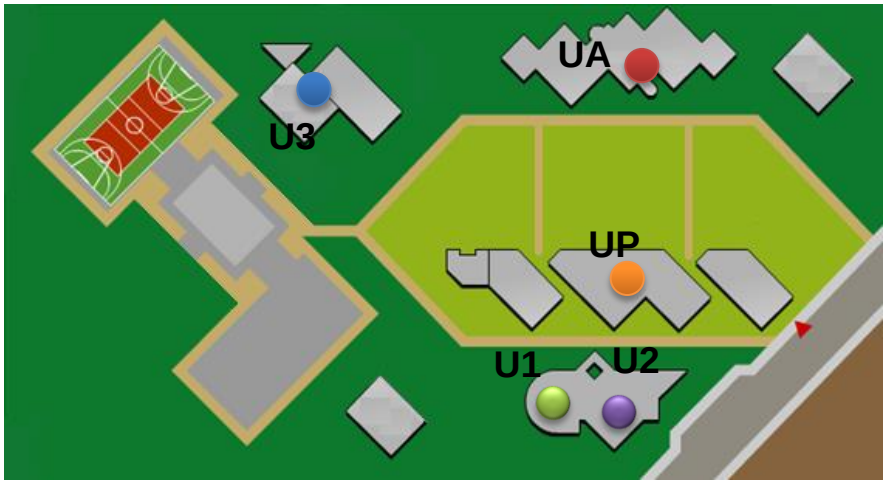
Fuente: Universidad Surcolombiana

Redes Inalámbricas

Dentro de la sede se cuenta con cinco (5) routers de marca Hewlett-Packard (HP), y las redes encontradas son:

- U1: USCO1
- U2: USCO-2
- U3: USCO-3
- UA: USCO-AULA
- UP: USCO-PITALITO

Figura 15. Mapa de la sede con las redes wifi identificadas



Fuente: Universidad Surcolombiana.

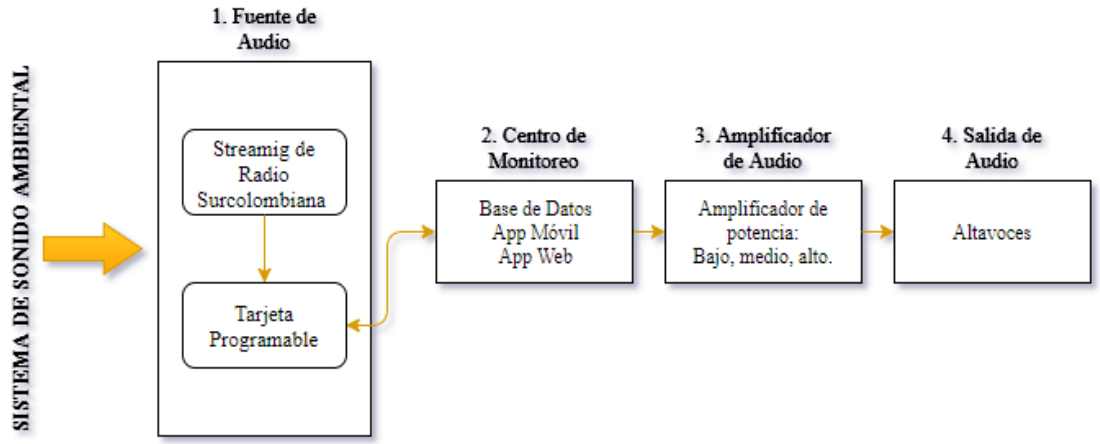
El planteamiento del proyecto contempla la distribución de módulos en varios sectores del sistema de sonido ambiental en cada sede. En cada lugar se tienen características acústicas distintas y por ende se manejan tres tipos de potencia en los amplificadores: baja, media y alta.

- ***Baja potencia:*** Diseñado para trabajar en espacios cerrados, menores a 25 m²; destinado a ser instalado en oficinas.
- ***Media potencia:*** Para usar en lugares parcialmente cerrados, aquellos que tengan un área relativamente grande por donde normalmente hay concurrencia de personas y no se necesita mayor nivel de sonido.
- ***Alta potencia:*** Propuesto para campo abierto, es decir, lugares amplios en donde la estructura es reducida.

2.3 CARACTERIZACIÓN DEL HARDWARE

Para el desarrollo del sistema físico de sonido ambiental, en el diseño del hardware se trabaja desde una descripción estructural de los elementos, en donde se enumeran los componentes del sistema y sus conexiones. Se describen cada uno de los componentes utilizados según la estructura de la Figura 16.

Figura 16. Esquema estructural del sistema de sonido ambiental planteado



Fuente: Autor.

2.3.1 Fuente de audio. Las partes que componen la fuente de audio del sistema vienen de un medio intangible como lo es el streaming de la emisora en Internet y de un medio físico que es la tarjeta programable destinada para lograr la transmisión de la emisora en los distintos puntos de cada sede.

Según los resultados obtenidos en el apartado 2.1.1, la tarjeta programable escogida es la Raspberry Pi 3, por su capacidad de procesamiento, costo, disponibilidad de librerías para programación y wifi integrado.

Raspberry Pi 3: es un computador de placa reducida (SBC) de bajo coste, desarrollado en el Reino Unido por la Fundación Raspberry Pi (Universidad de Cambridge), con el objetivo de estimular la enseñanza de ciencias de la computación.

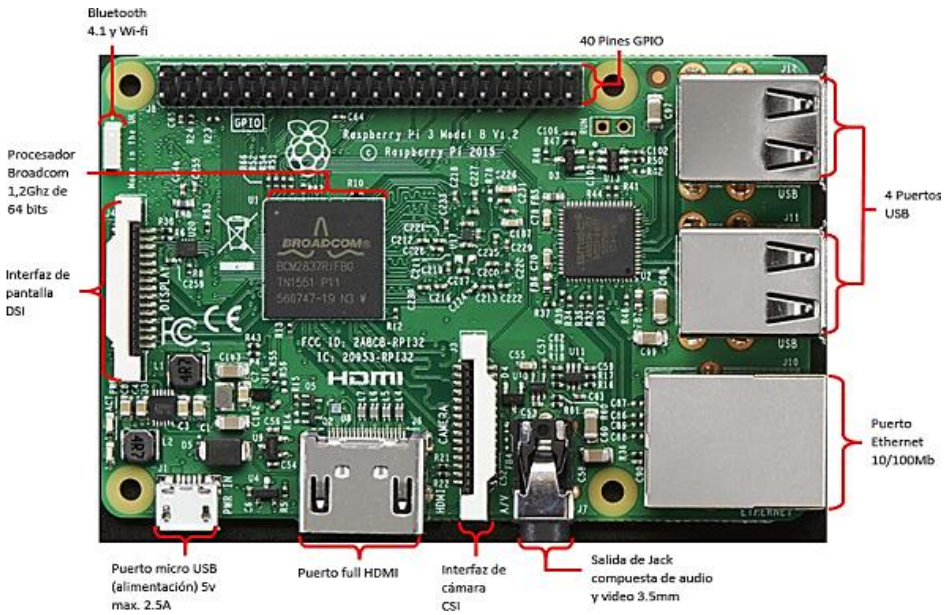
El concepto es el de un ordenador despojado de todos los accesorios que se pueden eliminar sin que afecte al funcionamiento básico.

Figura 17. Tarjeta Raspberry Pi 3

Fuente: Pimoroni

- ✓ *Procesador: Broadcom BCM2837*
- ✓ *CPU: 1,2 GHz*
- ✓ *GPU: Dual Core VideoCore IV*
- ✓ RAM: 1Gb
- ✓ Salida de Audio: Jack de 3.5mm y HDMI
- ✓ Puertos USB 2.0: 4
- ✓ Conectividad: Conector RJ-45, Wifi 802.11n y Bluetooth 4.1

Figura 18. Partes de la Raspberry Pi 3 Model B



Fuente: T-Bem.

La tarjeta Raspberry Pi 3 adquirida para el proyecto, cuenta con caja protectora, disipadores de calor para cada uno de los chips de la tarjeta y ventilador.

Figura 19. Raspberry Pi 3 con caja más ventilador

Fuente: Autor.

Centro de monitoreo. Es la base de datos, la App móvil y la App web. Cuyos elementos serán trabajados desde cualquier dispositivo móvil con sistema operativo Android o computador. Más adelante se mira en detalle el diseño y puesta en funcionamiento del mismo.

2.3.2 Amplificador de audio. Se trabaja con amplificadores del fabricante SURE Electronics, de arquitectura clase D para audio, con un desempeño de fidelidad del sonido suficiente para manejar las potencias requeridas, con valores designados de acuerdo con lo existente en el mercado: 8W, 50W y 100W, respectivamente.

*Baja potencia (8W): e*l AA-AB32231 es una placa amplificadora de audio que emplea el chip TPA3110D2 de TI (Texas Instruments). La eficiencia del chip elimina la necesidad de un disipador de calor externo al reproducir música. Es compatible con la amplificación de dos canales y cada uno de esos canales tiene una potencia de 8W que ayuda a la producción de calidad de sonido de última generación.

Aplicaciones:

- Sistemas de música de fondo.
- Distribución multi-canal.

Tabla 2. Algunas características del AA-AB32231

Características	
Fuente	8V-19V DC
Rta en frecuencia	20Hz-20KHz
Eficiencia	92% @ 4W 8Ω 85% @6W 4Ω

Fuente: Sure Electronics.

Figura 20. Amplificador 2 x 8W

Fuente: Sure Electronics.

Para más información de especificaciones eléctricas y modos de conexión, ver el Anexo A y B respectivamente.

Media Potencia (50W): el AA-AB32174 es un amplificador con doble canal de amplificación de audio y protección termal con ventilación refrigerada. Cuenta con un chip TDA7492 de ST (STMicroelectronics), haciéndolo un amplificador de alta calidad.

Aplicaciones:

- Sistemas de música de fondo.
- Máquinas expendedoras.
- Kioscos interactivos.
- Car Audio.

Figura 21. Amplificador 2 x 50W

Fuente: Sure Electronics

Tabla 3. Algunas características del AA-AB32174

Características	
Potencia	50W
Fuente	14V-27V DC, 6A
Mínima impedancia	4 Ω
Rta en frecuencia	20Hz-20KHz

Fuente: Sure Electronics

Para más información de especificaciones eléctricas y modos de conexión, ver el Anexo C y D respectivamente.

*Alta Potencia (100W): e*s una tarjeta amplificadora de un solo canal de salida de 100W, que utiliza chips TC2000 y TP2050. Soporta cuatro canales de amplificación y puede ser utilizada para parlantes pasivos de 4Ω y 8Ω. Su referencia es la AA-AB32186.

Aplicaciones:

- Kioscos interactivos.
- Car audio.
- Sistema de refuerzo de sonido.

Figura 22. Amplificador 2 x 100W

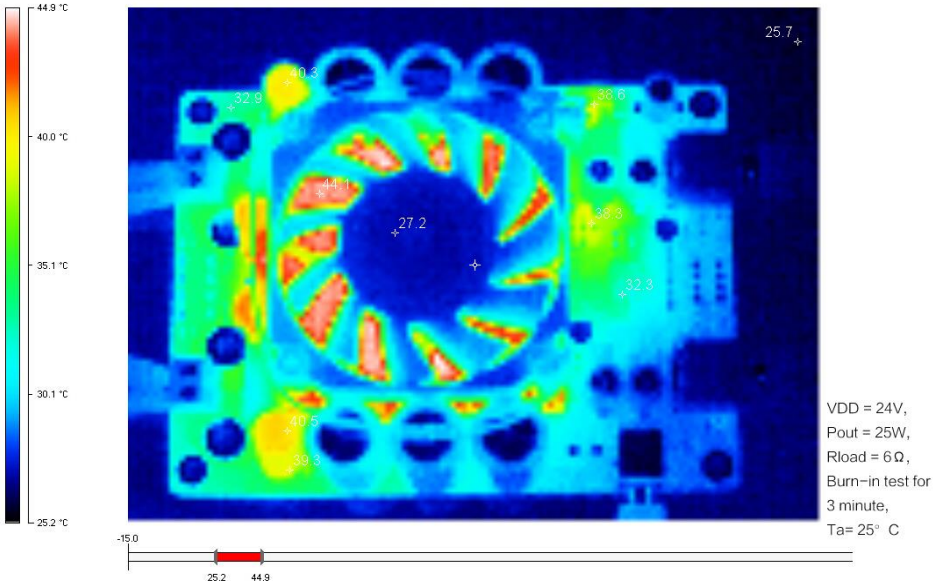
Fuente: Sure Electronics

Tabla 4. Algunas características del AA-AB32186

Características	
Fuente	10-30V DC
Rta en frecuencia	20Hz-20KHz
Eficiencia	95% @ 100W

Fuente: Sure Electronics.

Figura 23. Imagen térmica del amplificador en funcionamiento



Fuente: Sure Electronics

Para más información de especificaciones eléctricas y modos de conexión, ver el Anexo E y F respectivamente.

Ventajas de los amplificadores:

- Varios métodos de cableado facilitan la conexión: Socket RCA y bloque de terminal.
- Excelente disipación de calor que elimina la necesidad de un extra disipador de calor.
- Excelente diseño de los puertos de alimentación que permite la conexión de múltiples tarjetas amplificadoras en serie.

2.3.3 Salida de audio. Dentro de la amplia gama de altavoces existentes en el mercado, para las medias y altas potencias se direcciona hacia aquellos utilizados en exteriores con mayor durabilidad a condiciones del ambiente marca Bose y para la baja potencia se dirige hacia altavoces pequeños genéricos de marca Yamazaki con calidad de sonido adecuada.

Parlantes Bose 151 SE: los altavoces para exteriores ofrecen mayor rendimiento y durabilidad para lograr sonido estéreo completo en una amplia área de audición. Estos altavoces para exteriores Bose cuentan con un diseño de altavoz Articulated Array® dispuestos para ofrecer sonido estéreo completo en un área mucho más amplia que los altavoces para exteriores convencionales. El resultado es un rendimiento de altavoces Stereo Everywhere® que logra una experiencia de audición constante de un punto a otro dentro de la cobertura.

Figura 24. Parlante Bose 151 SE

Fuente: Bose.

Especificaciones técnicas:

- Tres (3) transductores ambientales de 6cm y de rango completo por altavoz.
- Dimensiones: 11,4 cm * 15,2 cm * 31,8 cm

- Compatible con amplificadores o receptores de 10-100W por canal con una potencia de 4-8 Ω.
- Administración de potencia continua de 50W @ 6Ω
- Límites de temperatura: -40°C a 70°C
- Resiste salinidad, niela, sol y condiciones de humedad (relativa).

2.3.4 Fuentes de poder y acondicionamiento. Las fuentes de poder necesarias para alimentar la etapa amplificadora del sistema de sonido ambiental cumplen con los requerimientos eléctricos de cada amplificador.

Amplificador AA-AB32231: Para este amplificador de potencia baja (8W), una vez revisada la hoja de especificaciones eléctricas del anexo A, se trabaja con una fuente NES-35-12 del fabricante Mean Well, con las siguientes características:

Tabla 5. Características de NES-35-12

Características	
Entrada	120-370 V
Salida	12V @3A
Temp. De trabajo	-20°C - +60°C
Dimensiones	99 L*97 W*36 H (mm)

Fuente: Mean Well.

Figura 25. Fuente NES-35-12

Fuente: Mean Well.

Amplificadores AA- AB32174 y AA-AB32186: Para los amplificadores de potencia media y alta, 50W y 100W, respectivamente. Según la hoja de especificaciones eléctricas de los anexos C y E, se trabaja con una misma fuente. La fuente seleccionada es la NES-150-24 del fabricante Mean Well, con las siguientes características:

Tabla 6. Características de NES-150-24

Características	
Entrada	100-240V
Salida	24V @6.5A
Temp. De trabajo	-20°C - +60°C
Dimensiones	199 L*98 W*38 H (mm)

Fuente: Mean Well

Figura 26. Fuente NES-150-24

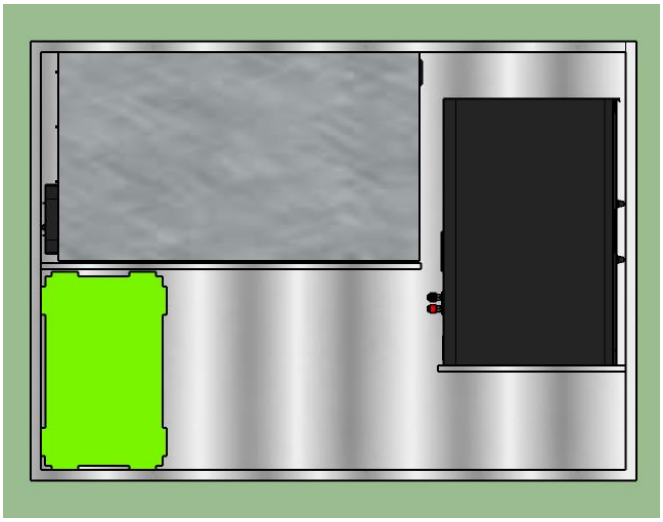
Fuente: Mean Well

Para el acondicionamiento de los equipos electrónicos correspondientes a la tarjeta programable, el amplificador de audio y las fuentes de alimentación, se ha planteado el diseño de una estructura rígida que resguarde los elementos de situaciones de temperatura, humedad y polvo.

Teniendo en cuenta los tamaños de los materiales y con la idea de utilizar una caja de material resistente, se buscan distintos materiales tomando como opciones el aluminio, acero, madera y acrílico.

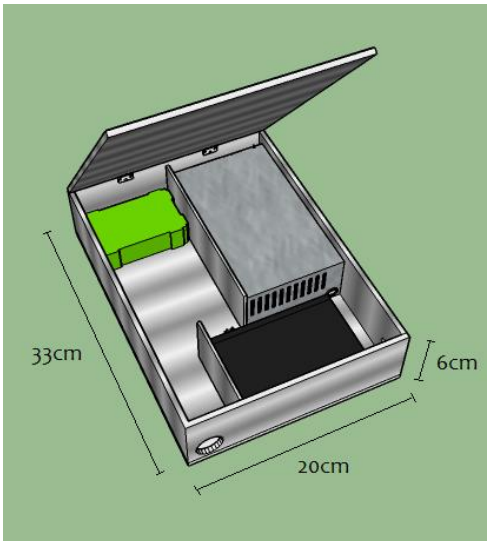
Una vez verificadas las opciones desde el punto de vista económico y de prestaciones, se opta por trabajar con cajas de acrílico.

Figura 27. Vista 2D del diseño de la caja



Fuente: Autor.

Figura 28. Diseño de la caja con medidas



Fuente: Autor.

Cable eléctrico

El cable usado para la conexión de la parte eléctrica es de marca *Centelsa*, el cual es usado normalmente como cordón de servicio liviano para la conexión de aparatos electrónicos, éste posee las siguientes especificaciones:

Tabla 7. Especificaciones cable eléctrico

Voltaje	300v
Calibre	12 AWG
Diámetro	2,46 milímetros
Espesor Aislamiento	0,64 milímetros
Peso Total	85 kilogramos/kilómetro
Resistencia DC a 20°C	5,31 ohm/kilómetro
Temperatura de operación	60°C
Capacidad de corriente	25 A

Fuente: Cables Flexibles-Centelsa

Figura 29. Cable eléctrico 2 x 12 AWG @ 300V

Fuente: Centelsa

Cable de audio

El cable usado para la conexión entre la caja que contiene el amplificador y el parlante es de marca Zuinsai Electronics, el cual usa un tipo de tecnología OFC (Libre de Oxígeno), evitando una oxidación y el deterioro de los cables; el cual posee las siguientes características:

Tabla 8. Especificaciones cable de audio

Área	2,08 milímetros cuadrados
Calibre	14 AWG
Diámetro total	2,32 x 6,56 Milímetros
Espesor Aislamiento	0,64 milímetros
Peso Total	56,6 kilogramos/Kilómetro

Fuente: Zuinsai Electronics

Figura 30. Cable de audio OFC 2 x 14 AWG

Fuente: Google-Imágenes.

Cable de audio para Raspberry Pi 3 hacia etapas amplificadoras

Dentro de las cajas en la sección de amplificación, se utilizan dos tipos de cable de audio: un cable de audio Plug 3,5 mm a 2 Plug RCA estéreo de marca *Steren* para la conexión entre la RasPi 3 y las etapas de 50W y 100W y un cable de audio Plug 3,5 mm a Plug 3,5 mm para la etapa de 8W con la tarjeta.

Figura 31. Cable plug 3.5mm a RCA.

Fuente: Steren.

Figura 32. Cable de audio 3.5mm a 3.5mm

Fuente: Google imágenes

2.4 DISEÑO DE APP MÓVIL Y WEB

El centro de monitoreo del sistema de sonido ambiental planteado en el proyecto es una aplicación móvil en la plataforma Android. Adicionalmente, por petición del grupo de la emisora de la universidad se hizo necesaria diseñar una aplicación web.

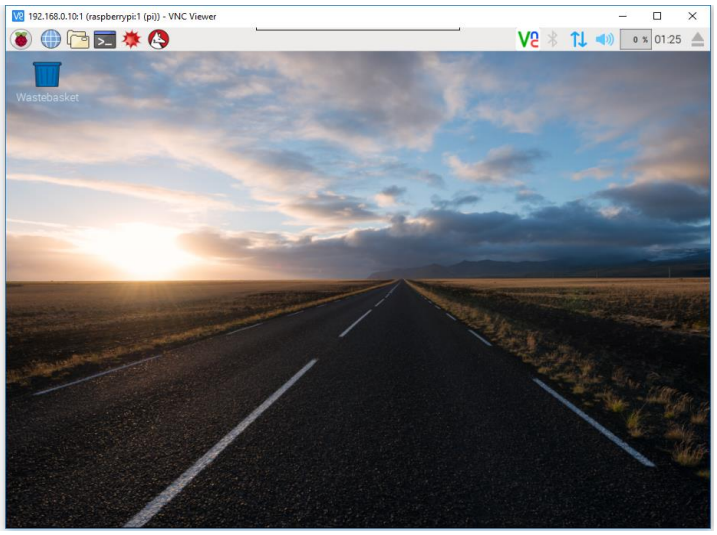
El primer paso es configurar la fuente de audio que en este caso será la tarjeta Raspberry Pi 3, ella tomará la señal del streaming de la emisora. Una vez lista la trasmisión de los datos, lo siguiente será diseñar la app móvil y web como tal.

2.4.1 Programación en Raspberry Pi 3 – Python (IDLE). Para programar en la tarjeta, es esencial tener instalado el sistema operativo para poder acceder y trabajar con Python desde consola. La tarjeta Raspberry Pi 3 adquirida, cuenta con NOOBS V2.0.0 guardado en su micro SD, por lo que se procede a hacer una instalación desatendida del Raspbian OS.

Noobs es una imagen ISO oficial de Raspberry.org que permite gestionar distintos OS para la Raspberry Pi.

Para total información acerca de la instalación del OS se puede encontrar en el anexo G. Una vez lista la instalación el resultado será tener acceso remoto a la tarjeta programable con Raspbian OS.

Figura 33. Escritorio PIXEL de Raspbian OS.



Fuente: Autor.

Como último paso se debe realizar la actualización de la tarjeta con los siguientes comandos a través del terminal:

sudo apt-get update

sudo apt-get upgrade

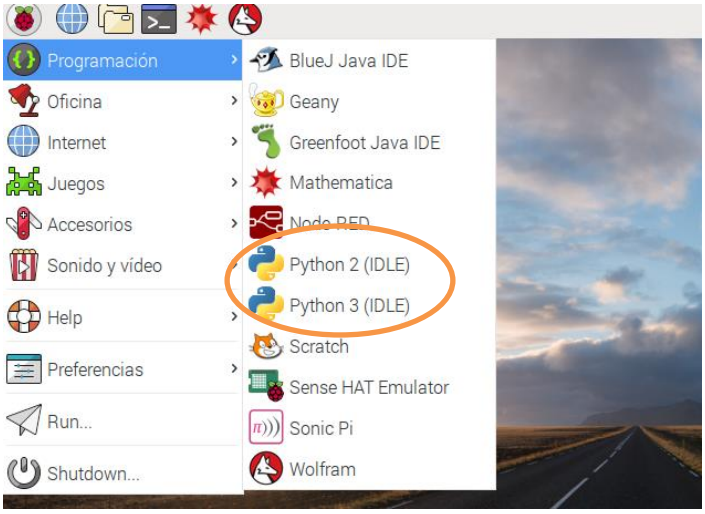
Ahora se empieza a realizar la descarga de módulos o paquetes pertinentes para iniciar a programar en el lenguaje Python.

Python (IDLE)

Raspbian cuenta con dos versiones de Python, la 2.7 y la 3.4, cada una de ellas con IDLE (Integrated DeveLopment Environment for Python), que es el entorno de desarrollo que permite editar y ejecutar los programas de Python.

Debido a la falta de librerías actualizadas en la versión 3, para el proyecto se opta por utilizar la versión 2 (2.7), compatible con todas las herramientas necesarias para llevar a cabalidad la programación planteada.

Figura 34. Ruta de acceso a Python en sus dos versiones.



Fuente: Autor.

Según Marzal y Gracia en su libro *Introducción a la programación con Python*, el contenido del fichero en IDLE es normalmente denominado como *script*[5].

Para el proyecto se busca que el *script* cumpla los siguientes criterios para desempeñar un buen funcionamiento:

- Establecer conexión entre la tarjeta Raspberry Pi y el servidor de la base de datos.

[5] Marzal, A. & Gracia, I. (2003). *Introducción a la programación con Python*. (pp. 59-60).

- De acuerdo con los datos recibidos del *Database*, la tarjeta ha de comportarse según las órdenes dadas, como la activación o desactivación del streaming de la emisora y/o el cambio en el volumen de la Raspberry Pi.
- Enviar a la base de datos el estado en el que se encuentra la tarjeta durante su operación cuando se le requiera, para poder monitorear a gusto todo el sistema de sonido.

Sin embargo, fuera de la programación contenida en el script, hace falta la adición de módulos correspondientes a la base de datos y a la tarjeta de audio de la RasPi para lograr total comunicación entre ellos.

Conexión Python – Firebase

Para enlazar Python con la base de datos creada en Firebase, se necesita descargar e instalar el módulo con las siguientes líneas de comando:

sudo pip install requests==1.1.0

sudo pip install python-firebase

Es importante adicionar en la parte inicial de los scripts a programar, la instrucción de la librería para que se reconozca los comandos específicos del módulo:

from firebase import Firebase

Módulo VLC para Python

Python requiere de una librería especial para la difusión de audio que sea compatible, por lo tanto, se descarga e instala en el módulo VLC, aplicable sólo para la versión de Python 2.7:

sudo pip install python-vlc

Una vez instalado, se agrega la librería para habilitar el uso del reproductor en el script: *Import vlc*

Tarjeta de sonido de la Raspberry Pi 3

La RasPi 3 posee una tarjeta de sonido *bcm2835 Alsa* accesible desde el módulo denominado Alsaaudio. Con el fin de manipular la variable del volumen desde el código de Python y lograr enlazar con la aplicación móvil se descarga e instala desde el terminal de Raspbian:

sudo apt-get install python-alsaaudio

Una vez terminado el proceso de instalación de manera exitosa, al igual que cada herramienta adicional, se escribe la librería en el script para así “llamar” los servicios que posee el módulo de Alsaaudio: *import alsaaudio*

Estructura de los scripts en Python

Al manejar bastantes datos, se decide manejar el código en diferentes bloques de scripts; ya que los datos al estar enlazados de manera independiente funcionan de manera ordenada, trabajando de forma paralela.

Las acciones dispuestas para los distintos scripts son:

- Activación/Desactivación del Streaming de la emisora
- Volumen General de cada sede
- Activación/Desactivación del volumen general
- Volumen Individual de cada sector

Entonces se hace un script principal, que técnicamente, éste lo que hace, es “llamar” los otros scripts secundarios ejecutándolos uno a la vez, según la orden que se le desde la App, evitando represamiento de datos y mala ejecución de las acciones.

Ejecución del script de Python

Se requiere que los script corran una vez se encienda la tarjeta Raspberry. Por lo tanto se hacen ejecutables los scripts y luego se les carga en el arranque del sistema.

- Script ejecutable de Python

chmod +x “nombre del archivo (.py)”

- Edición del fichero de arranque “profile”

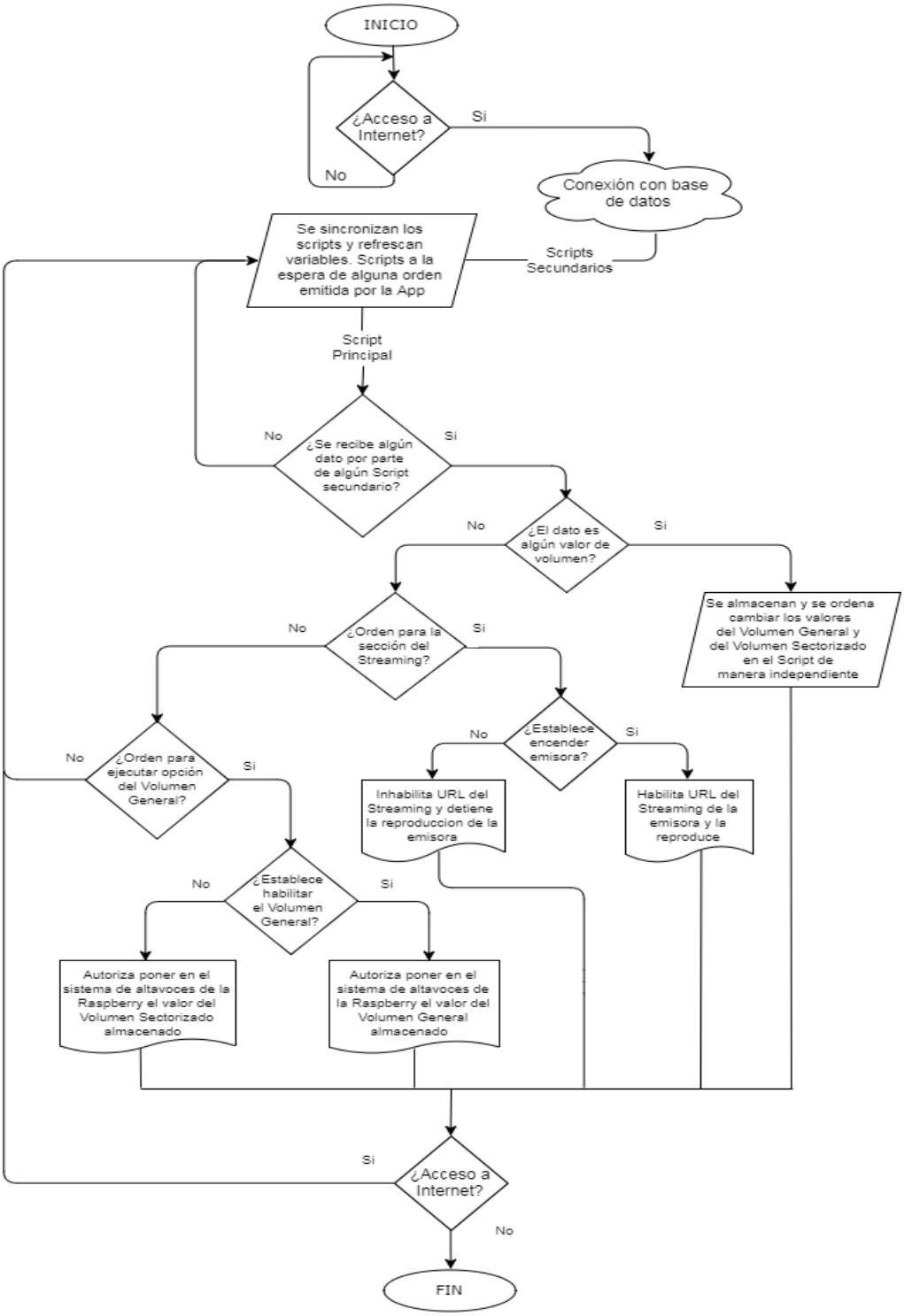
sudo nano /etc/profile

- *Adición de líneas de código en el fichero*

sudo python / (Ubicación del archivo ejecutable) &

sudo python /etc/home/desktop/main.py &

Figura 35. Diagrama de flujo de programación en Python en Raspberry Pi 3



Fuente: Autor.

A lo largo del apartado se hicieron mención a los scripts, ellos son:

- Script principal: Main.py
- Script Secundarios: Conexion.py, Conexion2.py, Conexion3.py y Control.py

Todos estos códigos están guardados dentro de una misma carpeta y los scripts secundarios se encuentran en una subcarpeta, ubicados en la ruta "/home/pi/Desktop/nombre del sector" de la Raspberry Pi y un ejemplo de cada código podrá ser observado con detalle en los anexos H, I, J, K y L respectivamente.

2.4.2 Android Studio. Para el desarrollo de la Aplicación Móvil, se ha optado por utilizar el entorno integrado Android Studio, software oficial de Google para la creación de aplicaciones, siendo exclusivos para la plataforma Android, sistema operativo existente en mayoría de los celulares inteligentes usados alrededor del mundo.

Figura 36. Logo de Android Studio

Fuente: Android Studio

Para empezar a crear la aplicación móvil, ha de tenerse en cuenta la práctica del MVC (Modelo-vista-controlador) vista en anterioridad, aquella técnica que organiza el código de manera que sea reutilizable y concreto. Por ello se ha planteado realizar una App que maneje diversas ventanas para abarcar en totalidad las funciones del sistema de sonido.

Entre las vistas planteadas se tiene la de *inicio de sesión*, la *lista de las sedes* y los *sectores de cada sede*; por ende, varían las herramientas a utilizar, ya sea el adicionamiento de botones, los editores de texto, las imágenes, todo eso con el fin de brindar funcionalidad total a la aplicación móvil y un aspecto agradable.

- **Inicio de sesión:** Es la primera vista para el usuario una vez se ha iniciado la App móvil, ofreciendo un formulario de ingreso con usuario y contraseña, con modo offline para salvaguardar la inadecuada activación de alguna función.

Figura 37. Interfaz de inicio de sesión en la aplicación móvil

Fuente: Autor.

- **Menú de selección de sedes:** Se encuentran las 3 sedes de la universidad en donde se van a tener los sistemas de sonido ambiental. Las opciones son: Sede La Plata, Sede Garzón y Sede Pitalito.

Figura 38. Menú principal de las sedes

Fuente: Autor.

- **Centro de monitoreo sectorizado:** En esta vista se localizan los sectores de cada sede con sus controles de volumen respectivos, permitiendo, de acuerdo

con la velocidad de la red de internet, el seguimiento de cada variable en tiempo real y la correcta ejecución de acuerdo con la orden dada.

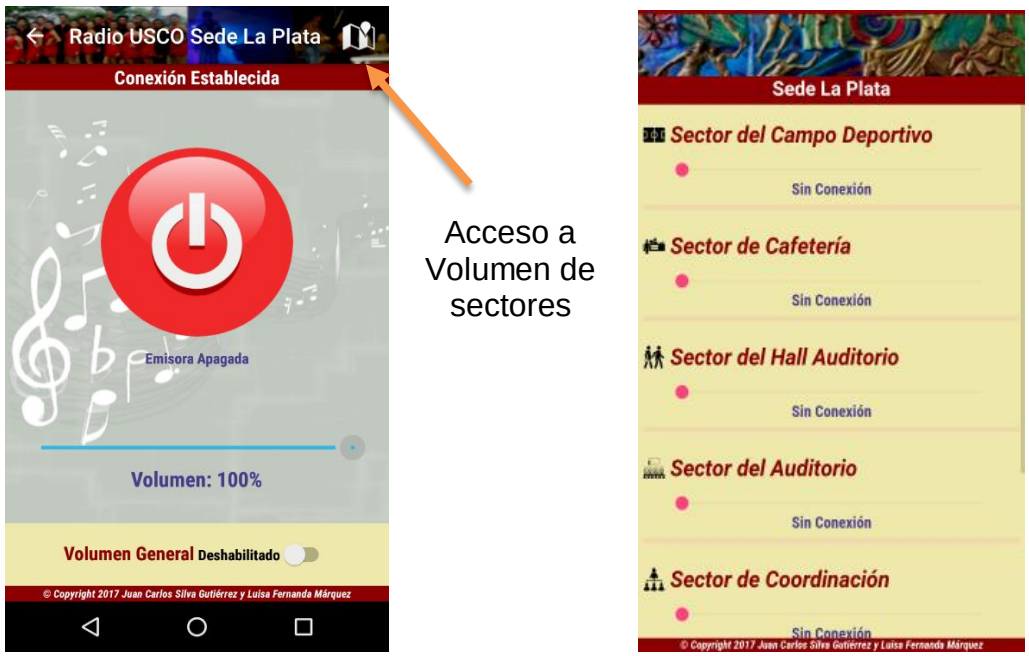
Según lo requerido, se dispone de dos centros de monitoreo, uno general y otro específico a los sectores de cada sede. El centro principal maneja un activador on/off encargado de la reproducción del streaming de la emisora con su respectiva barra de volumen, aplicable a todos los sectores de la sede.

En caso de necesitar sectores de manera independiente, se habilita o deshabilita la función del centro principal y se permite el acceso al modo específico “Volumen de Sectores”

Figura 39. Botón de Act/Desact de volumen general

Fuente: Autor.

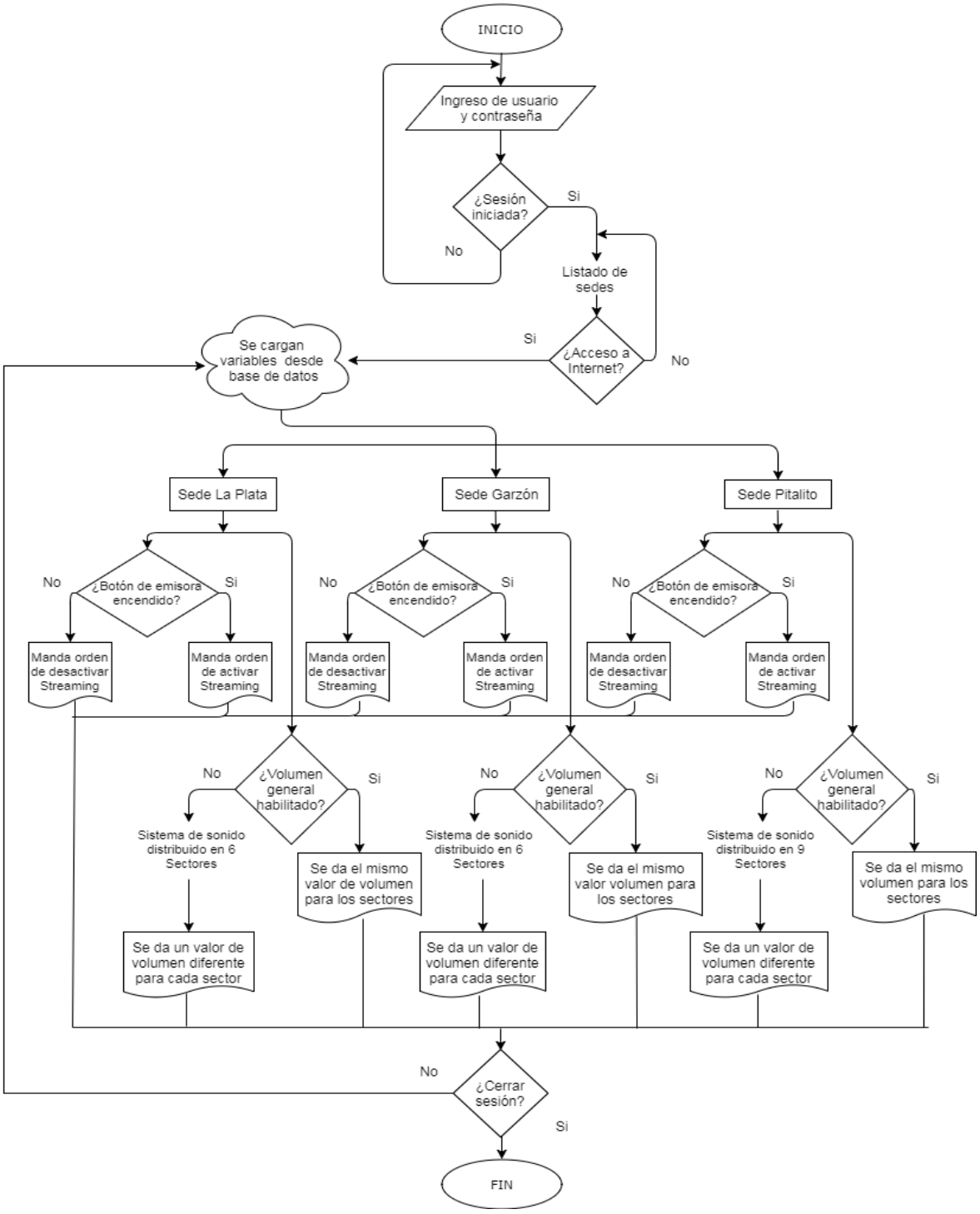
Figura 40. a. Interfaz de monitoreo general, b. Monitoreo sectorizado



a. *b.*

Fuente: Autor.

Figura 41. Diagrama de flujo de la aplicación móvil

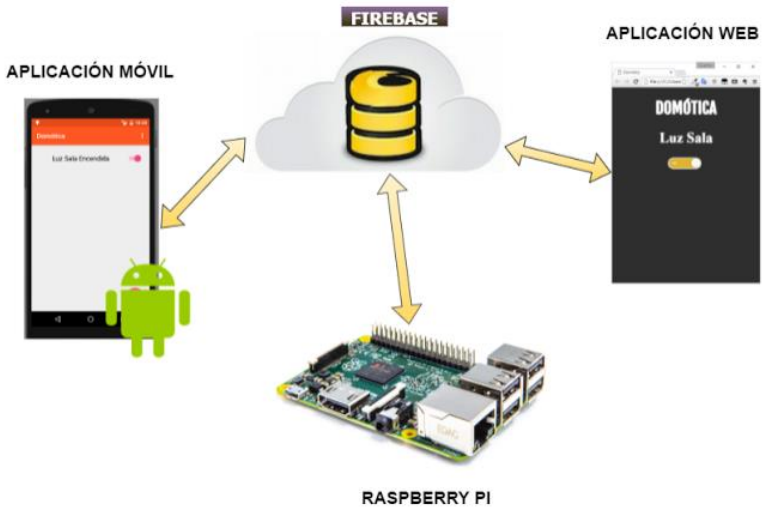


Fuente: Autor.

Finalmente se diseña una App móvil para dispositivos Android a partir de la versión 4.2 (Jelly Bean).

2.4.3 Firebase. Es una plataforma de desarrollo para aplicaciones que incluye varios servicios y el de mayor interés es la base de datos en tiempo real que posee, convirtiendo a Firebase en el mediador ideal que se necesita para comunicar la App móvil y web con la tarjeta Raspberry Pi 3.

Figura 42. Conexión Firebase, aplicativos y Rasberry Pi



Fuente: Blog Jefferson Rivera.

La ventaja de la base de datos radica en su rápida interacción entre variable y variable que brinda un correcto funcionamiento y estabilidad en un entorno dinámico y además está enlazada con el entorno de Android Studio.

Desarrollo de la base de datos en Firebase

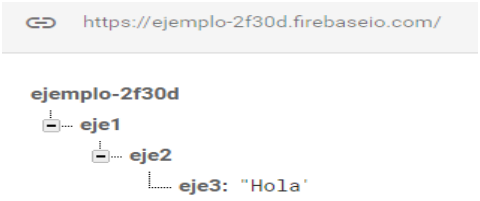
Una vez dentro de la consola principal de Firebase, se ingresa a **Database** con el fin de crear la base de datos que aloja las variables mediante una estructura de texto ligero JSON[***], agregándole jerarquía y orden a las instrucciones conservadas.

Algo que caracteriza a la base de datos en tiempo real de Firebase es que los datos se almacenan como objetos JSON en forma de árbol, que a diferencia de la base

[***] Es un texto escrito con la notación de objetos de JavaScript. Es una sintaxis para almacenar e intercambiar datos.

de datos SQL, no existen tablas ni registros. Cuando se agregan datos al árbol JSON, éstos se convierten en un nodo en la estructura JSON existente.

Figura 43. Ejemplo de estructura de base de datos

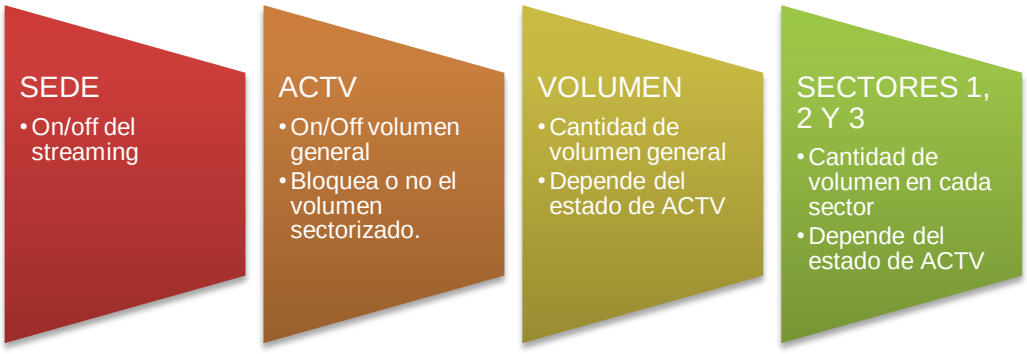


Fuente: Autor.

Para hacer eso posible, Firebase otorga para cada proyecto creado una dirección web única, la cual, representa la ubicación en donde los datos se guiarán para alojarse en la base de datos y dependiendo los datos que se envíen a dicha dirección, el fichero JSON en forma de árbol comenzará a organizar la información de forma jerárquica dejando a disposición el uso de obtener una lectura o hacer una publicación de las variables aseguradas en el *Database* de forma remota.

La estructura de la base de datos realizada para el proyecto se encuentra organizada de la siguiente manera:

Figura 44. Estructura de datos realizada en Firebase.



Fuente: Autor.

De esa manera, anexando la dirección correspondiente a la programación *java* de cada variable, se logra total manejo de los datos para el uso de lectura y escritura de los mismos a partir de la aplicación móvil y web.

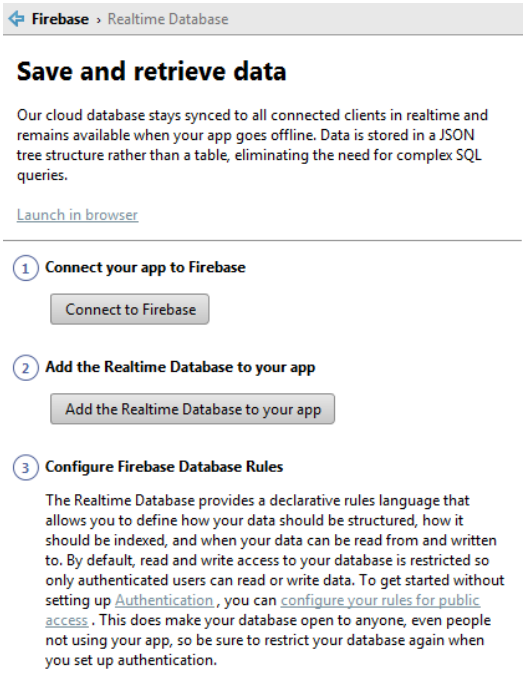
Figura 45. Visualización de estructura final en el Realtime Database.

Fuente: Autor.

Conexión Firebase – Android Studio

Dentro de Android Studio se presenta el módulo de Firebase en donde se escoge el servicio a utilizar y guiados a través de pasos adjuntos se logra finalmente el enlace entre la aplicación móvil y la base de datos.

Figura 46. Pasos para conexión de Android Studio con Firebase



Fuente: Autor.

Cabe resaltar que finalizando las instrucciones se encuentra las reglas de autorización para lectura y escritura dentro de la base de datos, dicho paso es indispensable y se hace directamente desde la consola de Firebase.

Figura 47. Reglas de autorización para lectura y escritura de datos en Firebase

Fuente: Autor.

Hasta este punto se ha logrado el diseño de la App móvil y su respectiva conexión con la base de datos en tiempo real de Firebase, en consecuencia, resta el diseño de la App web y su respectiva conexión con los elementos anteriores.

Aplicación web

Recordando la estructura planteada en el apartado 1.4, el enfoque en el diseño de la aplicación web va hacia el back-end y el front-end.

Por una parte, Firebase es el back-end necesario, debido a su estructura de fácil manejo y su portabilidad de una App a la otra. De otro lado, el front-end se realiza bajo el estándar de los aplicativos webs: HTML y CSS para darle forma y JS para la funcionalidad de la misma App web y para ello se utiliza el editor de texto: Brackets.

2.4.4 Editor de texto: Brackets. Es un proyecto de código abierto que está escrito en JavaScript, HTML y CSS. Tiene características únicas como la Edición rápida y la Vista previa dinámica, bastante útiles a la hora de programar.

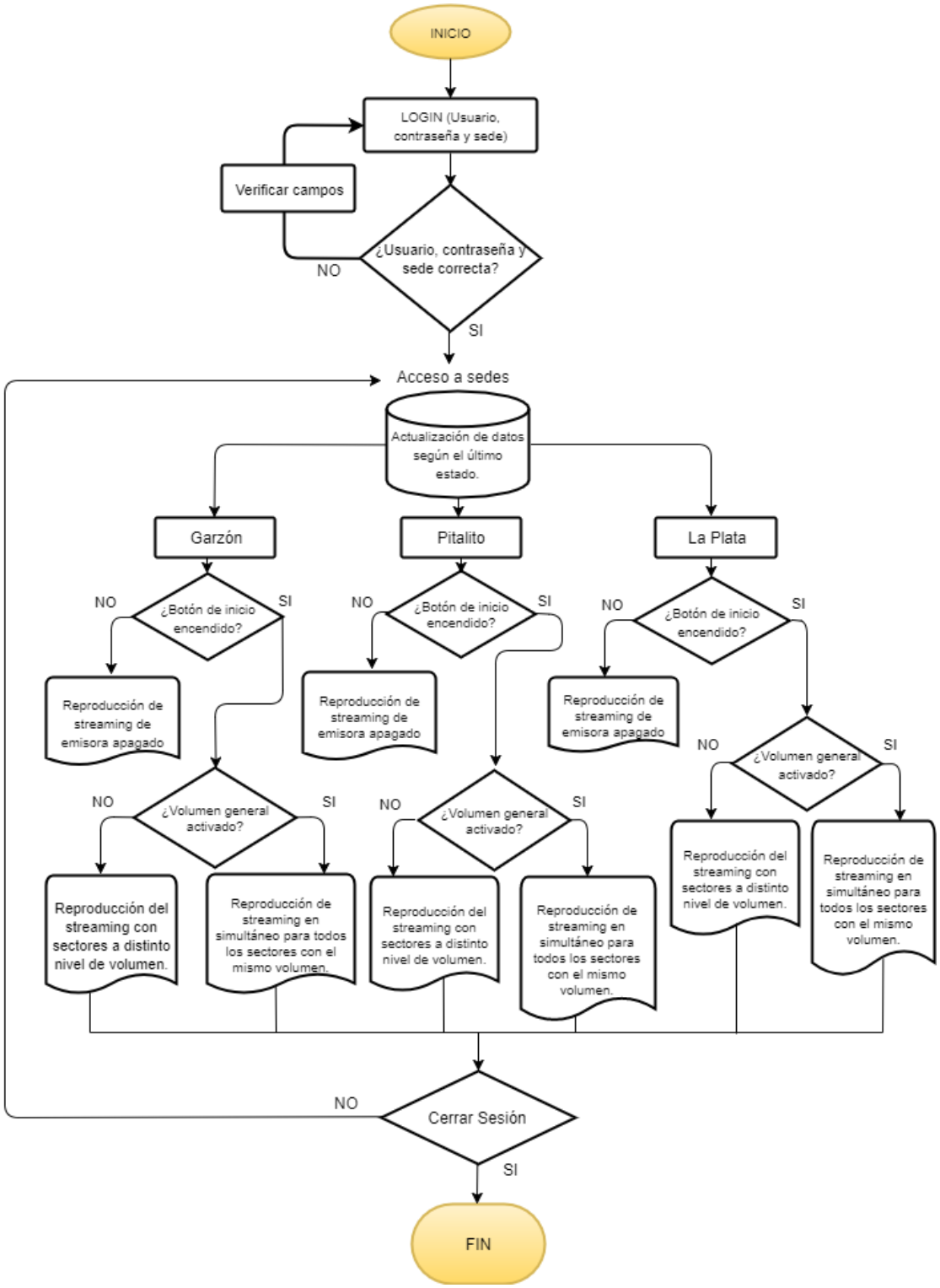
Se encuentra disponible en http://brackets.io/ en su versión 1.10. Una vez instalado, se pueden encontrar varios lenguajes de programación ya que cuenta con alrededor de 40. En esta ocasión se enfoca en los mencionados en anterioridad.

Figura 48. Interfaz principal de Brackets

Fuente: Autor.

En este editor se empieza a dar forma al aplicativo web, partiendo del siguiente diagrama de flujo propuesto para la programación.

Figura 49. Diagrama de flujo de la aplicación web



Fuente: Autor.

Para empezar a escribir código en Brackets, se establece una carpeta en donde se ubicarán los archivos.

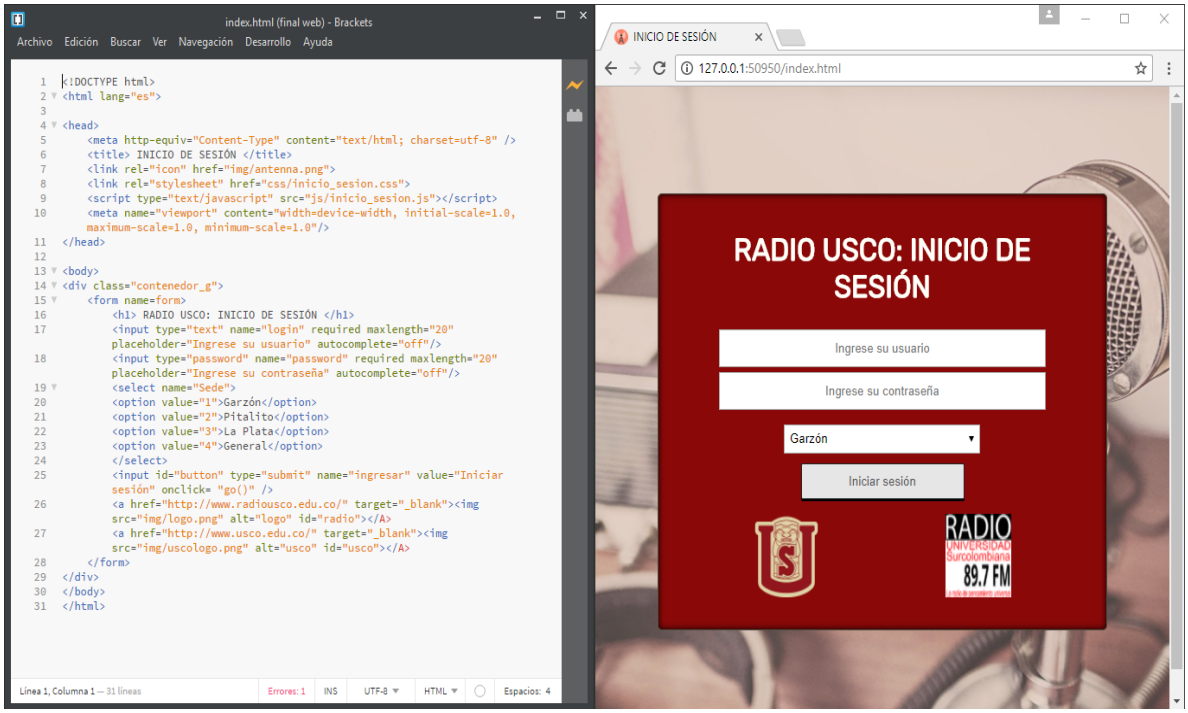
Teniendo en cuenta el diagrama de flujo planteado, la aplicación web se divide en:

- **Página de inicio**
- **Página de sede con sectores**

A nivel de código se establece un orden, la parte del HTML con su respectivo CSS para decoración-estilos y la parte JS encargada del control de acciones a realizar durante el funcionamiento del aplicativo.

Para iniciar se realiza la página de inicio que contendrá un formulario HTML, capaz de solicitar los datos de ingreso al usuario, permitiendo o negando el ingreso al monitoreo de las variables de volumen en cada sector de cada sede a partir de su configuración en un archivo de JS llamado *inicio_sesion.js.*

Figura 50. Vista dinámica de la página de inicio



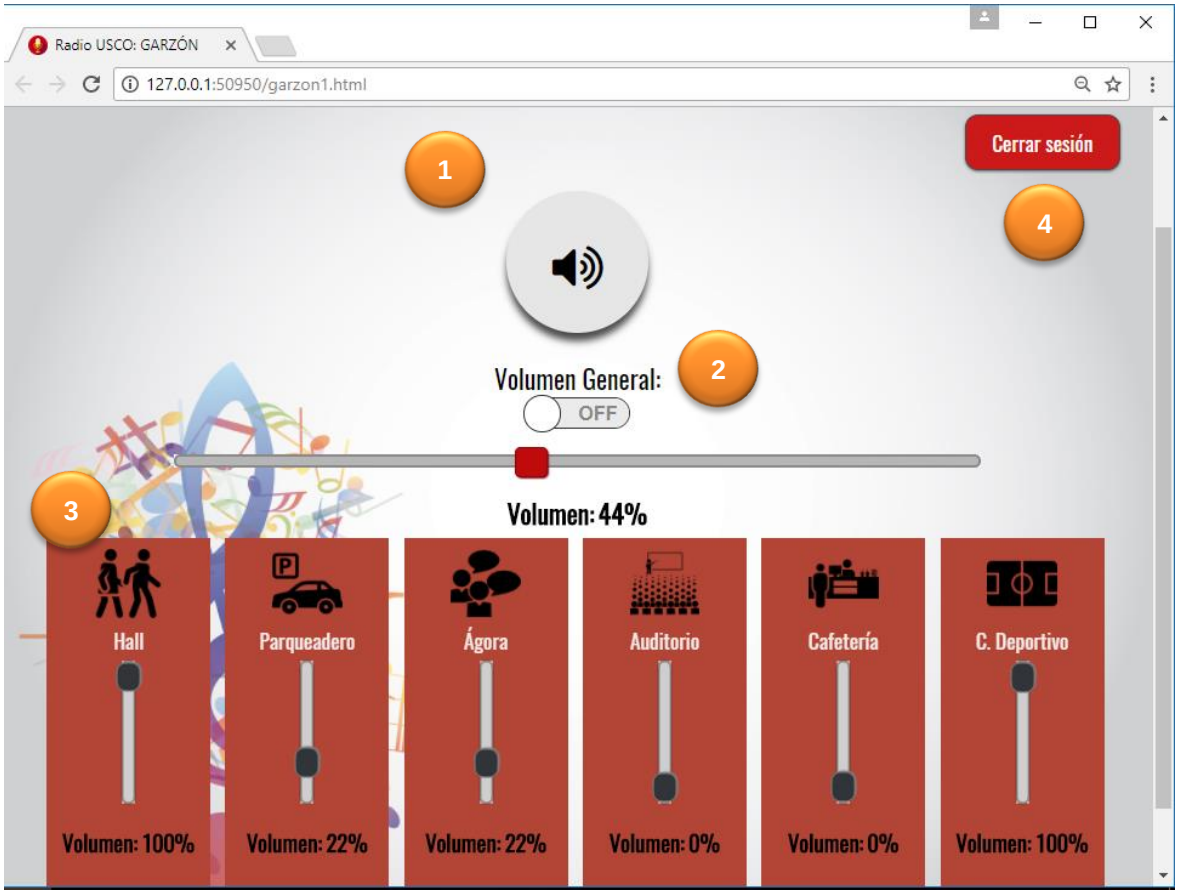
Fuente: Autor.

En caso de ingreso erróneo de credenciales se muestra al usuario una ventana emergente de notificación.

Una vez se ha ingresado de manera exitosa al aplicativo, se despliega el menú de la sede con cuatro partes:

1. Botón on /off de la trasmisión
2. Modo de volumen general
3. Modo de volumen sectorizado
4. Botón de Cerrar sesión

Figura 51. Página del menú de las sedes



Fuente: Autor.

1. Botón on /off de la trasmisión: Botón ligado directamente con la reproducción o no del streaming de la emisora. A través de la conexión con Firebase, en donde modifica la variable de reproducción al establecer un true o false.

2. Modo de volumen general: Se tienen dos maneras de monitoreo del sistema de sonido ambiental para cada sede, de manera general e individual. En este caso como su nombre lo indica, se tiene un manejo equitativo de todas las variables,

siempre y cuando el objeto flipswitch se encuentre activado. Enviando el mismo valor de volumen para ellas.

3. Modo de volumen sectorizado: Una vez se ha desactivado el flipswitch encargado del volumen general, se da acceso a indicar valores distintos para cada sector a gusto del usuario.

4. Botón de Cerrar sesión: Este botón permite concluir el monitoreo del sistema de sonido, en caso de no querer trabajar más en el aplicativo. Cabe resaltar que así se cierre la App Web, los datos almacenados en la misma son enviados a la base de datos, quien maneja todo y por ende no altera el funcionamiento.

Firebase y Brackets

Toda la anterior interacción del aplicativo web va acompañada del back-end proporcionado por Firebase. Para realizar monitoreo de cada variable, es necesario tener permiso para escribir y leer de la base de datos, como ya se vio con la App móvil.

Esto se logra a través de las librerías que brinda Firebase para su conexión con JS.

Figura 52. Código para importar librería de Firebase

```
<!-- importamos la libreria de firebase cliente para javascript -->
<script src='https://cdn.firebase.com/js/client/2.2.1/firebase.js'></script>
<!-- Logica de comunicación con firebase  -->
```

Fuente: Autor.

Una vez se tienen las librerías al igual que en la App móvil, se llega a las variables a través de las url dispuestas para cada una. A partir de allí queda disponible para que la programación lea y escriba datos necesarios para que todo funcione como debe ser.

Figura 53. Código para acceder a una variable en Firebase

```
var ref = new Firebase("https://uscoemisora.firebaseio.com/sede/garzon/");
```

Fuente: Autor.

Finalmente, la App web se dejará alojada de manera estática en cada uno de los computadores de cada sede.

3. IMPLEMENTACIÓN DEL SISTEMA

A lo largo de este capítulo se verá el proceso de implementación del sistema de sonido ambiental para cada una de las sedes. Conociendo la unión del software y hardware realizada. Apoyados en la experiencia del técnico de sonido Alex Q, quien estuvo presente en cada sitio brindando la asesoría correspondiente a distribución e instalación de los parlantes de acuerdo a la cobertura de sonido planteada.

3.1 MODELO FINAL

Una vez listo el hardware necesario para el sistema de sonido, así como el diseño de cada una de las aplicaciones, se procede a montar el sistema de sonido ambiental. Cada sistema en cada sede tiene a su vez determinado número de módulos. Dichos módulos cuentan con una caja realizada en acrílico de 0,4cm de grosor, que guarda en su interior la fuente de alimentación, el amplificador de audio, la tarjeta Raspberry Pi 3, los cables de conexión eléctrica y de audio respectivos. Además de contar con su parlante de manera externa a la caja.

Como ya se había mencionado, teniendo en cuenta los requerimientos sonoros y espacio de las sedes, se diseñaron tres tipos de módulos para cada sistema de sonido de acuerdo con la potencia.

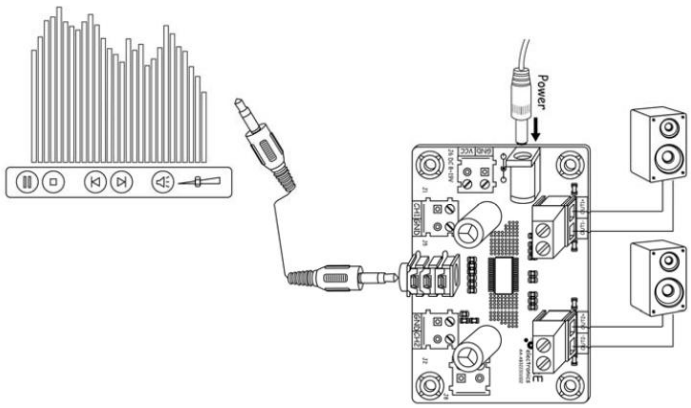
- Baja potencia (8W)
- Media potencia (50W)
- Alta potencia (100W)

Cada uno de los sistemas lleva la estructura planteada en la figura 16.

3.1.1 Baja potencia. Dicho módulo cuenta con:

- ✓ Fuente de alimentación de 12V @ 3A
- ✓ Etapa amplificadora referencia AA-AB32231
- ✓ Tarjeta Raspberry Pi 3 en caja protectora
- ✓ Adaptador para RasPi 3 – 5V @ 3A
- ✓ Cable de audio Plug 3,5 mm a Plug 3,5 mm
- ✓ Cable eléctrico 2 x 12 AWG a 300 V
- ✓ Cable de Audio OFC 2 x 14 AWG

Figura 54. Esquema de conexión para entradas y salidas del amplificador

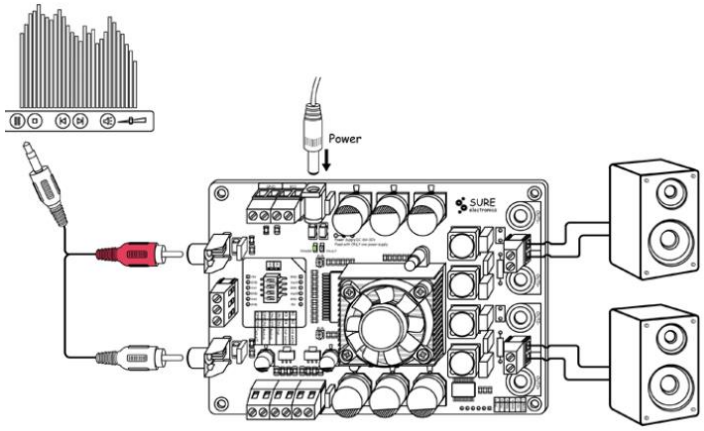


Fuente: Sure Electronics.

3.1.2 Media y alta potencia. Son módulos que requieren la misma fuente de alimentación para sus distintas etapas amplificadoras. Aptos para espacios semi abiertos y abiertos respectivamente. Dentro de la caja en acrílico contiene:

- ✓ Fuente de alimentación de 24V @ 6,2A
- ✓ Amplificador de audio referencia AA-AB32174 o AA-AB32186
- ✓ Tarjeta RasPi 3 en caja protectora
- ✓ Adaptador para RasPi 3 – 5V @ 3A
- ✓ Cable eléctrico 2 x 12 AWG a 300 V
- ✓ Cable de Audio OFC 2 x 14 AWG
- ✓ Cable de audio Plug 3.5 mm a Plug RCA

Figura 55. Esquema de conexión para amplificador de media y alta potencia



Fuente: Sure Electronics.

Para estos módulos es importante conocer la intensidad de sonido que puedan emitir los parlantes sin llegar a ser una fuente de ruido en el ambiente. Por ello se calcula teóricamente la Intensidad de sonido en Decibelios mediante las siguientes fórmulas:

$$I = \frac{P}{A} \quad \rightarrow \quad \frac{P}{4\pi r^2} \; {}^{W}/_{m^2}$$

$$\beta dB \; = \; 10log\frac{I}{I_0} \; dB$$

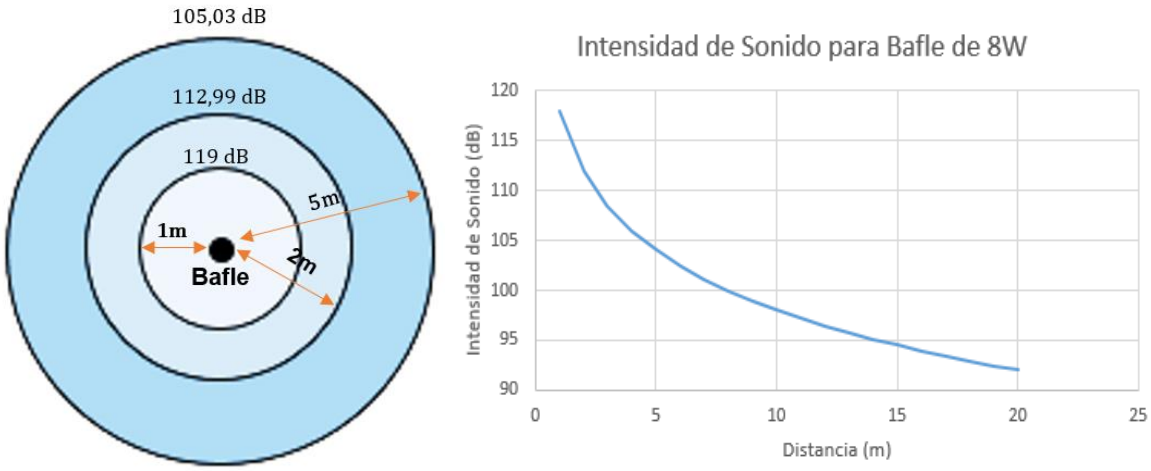
I_0 Se refiere al umbral del sonido que puede llegar a impresionar el oído humano, que en este caso es de $10^{-12} \; {}^{W}/_{m^2}$

Tabla 9. Intensidad de Sonido del Bafle de 8W

Distancia (metros)	1 m	2 m	3 m	5 m
Intensidad de sonido (dB)	118,04 dB	112,02 dB	108,49 dB	104,06 dB

Fuente: Autor

Figura 56. Frente de Onda y Mapa de distribución para 8W



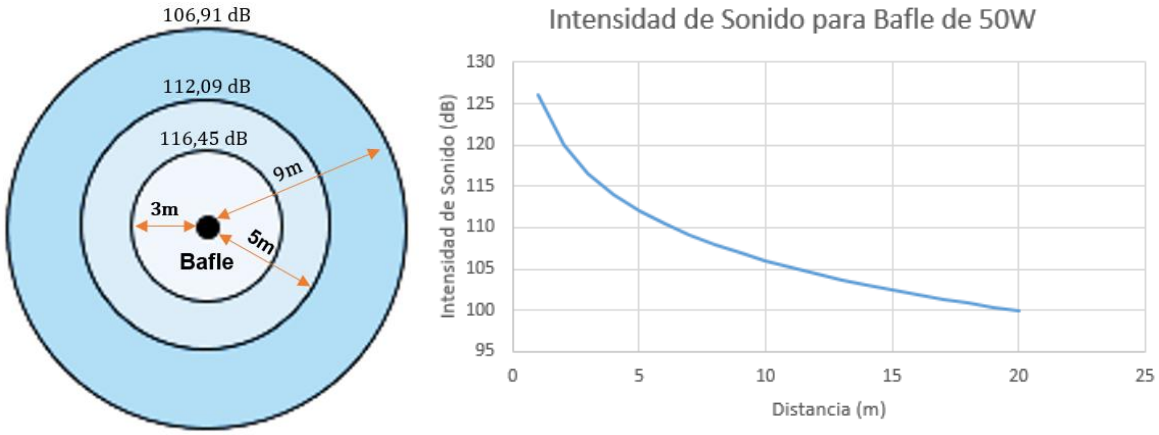
Fuente: Autor

Tabla 10. Intensidad de Sonido del Bafle de 50W

Distancia (metros)	3 m	5 m	7 m	9 m
Intensidad de sonido (dB)	116,45 dB	112,09 dB	109,09 dB	106,91 dB

Fuente: Autor

Figura 57. Frente de Onda y Mapa de distribución para 50W



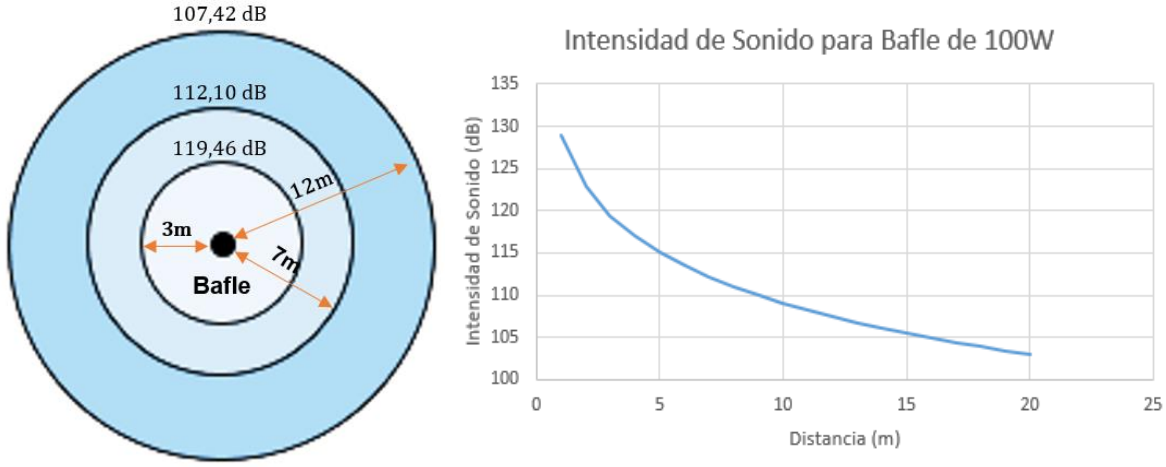
Fuente: Autor

Tabla 11. Intensidad de Sonido del Bafle de 100W

Distancia (metros)	3 m	7 m	9 m	12 m
Intensidad de sonido (dB)	119,46 dB	112,10 dB	109,92 dB	107,42 dB

Fuente: Autor

Figura 58. Frente de Onda y Mapa de distribución para 100W



Fuente: Autor

Al final se entrega tres tipos de cajas (Módulos) con las especificaciones en anterior mencionadas.

Figura 59. Montaje de módulo

Fuente: Autor

Figura 60. Vista interior del módulo terminado

Fuente: Autor

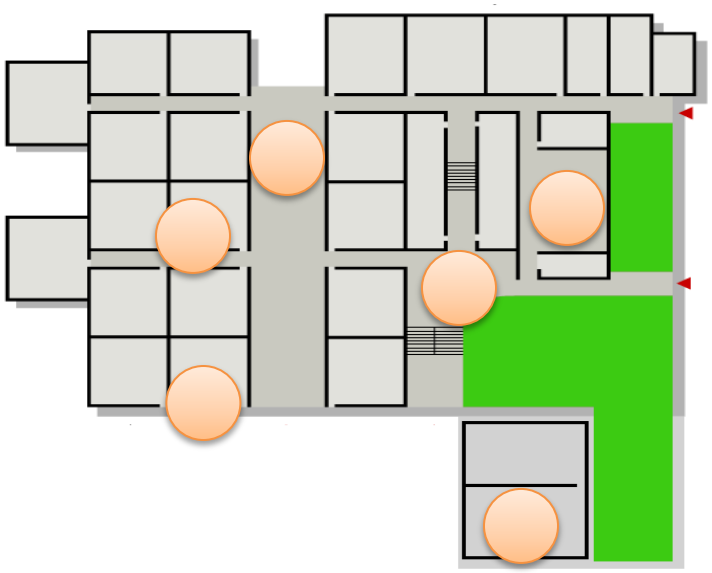
3.2 SECTORES

3.2.1 Sede La Plata. A pesar de contar con una planta física relativamente pequeña da lugar para poder instalar seis módulos. De esa manera, observando la cobertura de red de Internet y el mejor lugar para la instalación del módulo con su parlante, se distribuyen a lo largo de la sede.

Los sectores seleccionados son:

- ✓ Parqueadero
- ✓ Cafetería
- ✓ Hall del Auditorio
- ✓ Auditorio
- ✓ Oficina de Coordinación
- ✓ Corredor Principal

Figura 61. Mapa de la sede con la ubicación de los sectores



Fuente: Universidad Surcolombiana

Para los sectores en donde se encuentra la oficina de Coordinación y del Auditorio, se dispuso de módulos de baja potencia (8W), ha de aclararse que se decidió instalar baja potencia dentro del auditorio, debido a que es un recinto cerrado de tamaño reducido, cuya infraestructura hacía que el sonido generara eco, imposibilitando la instalación de un módulo con mayor potencia.

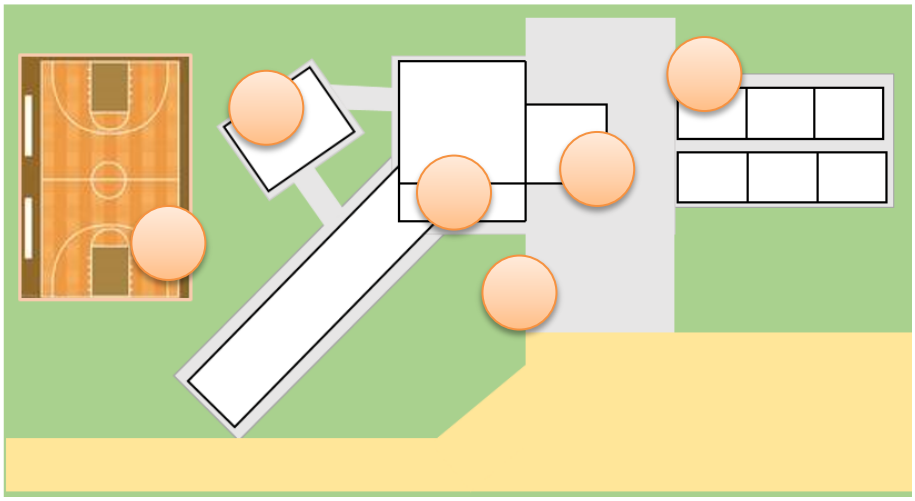
En el caso del Hall del Auditorio y de la Cafetería, se dispuso de módulos de media potencia, es decir, de 50W, al ser sectores semiabiertos. Finalizando con la

instalación, se ubican módulos de alta potencia (100W) en los sectores del Parqueadero y del Corredor Principal por ser zonas abiertas.

3.2.2 Sede Garzón. Estructuralmente es mediana y en su mayoría son aulas de clase, por ende, se decide instalar 6 módulos de audio, los sectores seleccionados son:

✓ Campo Deportivo
✓ Cafetería
✓ Parqueadero
✓ Ágora
✓ Auditorio
✓ Pasillo Secretaría

Figura 62. Mapa de la sede con los sectores escogidos



Fuente: Autor.

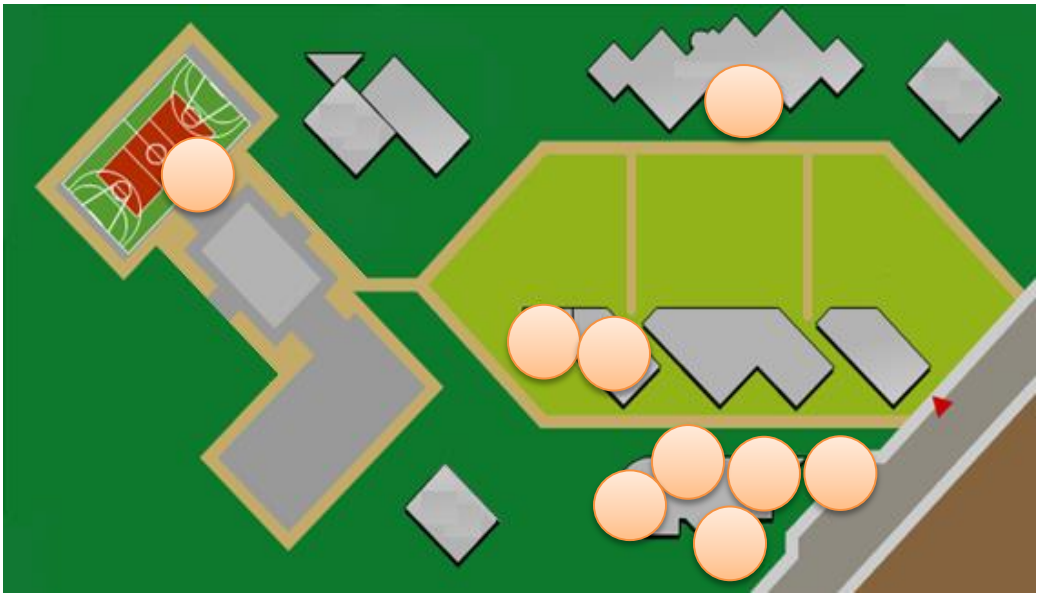
En total se instalaron tres (3) módulos de potencia media (50W) en sectores donde la concurrencia de gente era menor que en un espacio abierto. Los módulos de potencia alta (100W) fueron instalados en la zona del campo deportivo, el parqueadero y ágora donde el espacio abierto requería de la potencia planteada para lograr una mejor cobertura sonora sin tener problemas con la inteligibilidad.

3.2.3 Sede Pitalito. Se decide instalar nueve (9) módulos, distribuyendo pensando en la calidad del sonido de la trasmisión de la emisora, la no afectación de las clases y la tranquilidad del personal de la zona administrativa.

✓ Oficina de Coordinación
✓ Pasillo de Administración
✓ Oficina de Secretaría

✓ Oficina de Biblioteca
✓ Hall de Baños del primer piso
✓ Hall de Baños del segundo piso
✓ Campo Deportivo cubierto
✓ Cafetería-Restaurante
✓ Parqueadero

Figura 63. Mapa de la sede con los sectores



Fuente: Universidad Surcolombiana

Debido a que dentro del bloque del área administrativa es un recinto prácticamente cerrado, se dispuso cuatro (4) módulos de baja potencia (8W).

Para los sectores semiabiertos, como lo son los pasillos de baños y la cafetería, se situaron módulos de media potencia (50W), mientras que los sectores abiertos como lo es el campo deportivo y el parqueadero, se instalaron los bafles Bose con módulos de alta potencia (100W). Cabe resaltar que, en la zona del campo deportivo debido al área de trasmisión, se dispuso de un par de bafles Bose para mejor reproducción del streaming, evitando pérdida de sonido.

4. RESULTADOS

Lo resultados obtenidos son gracias al apoyo de la universidad para el desarrollo del sistema de sonido para la emisora institucional en sus sedes. Los recursos utilizados en este proyecto son en un 90% de la Universidad Surcolombiana, el 10% restante fue asumido y se vio reflejado en los materiales de conexiones eléctricas e instalación, el técnico de sonido y la manutención en los municipios durante la semana de instalación de los sistemas.

4.1 SISTEMA INSTALADO

En el proyecto se instalaron un total de 21 módulos de sonido distribuidos de la siguiente manera:

Tabla 12. Distribución de módulos en las sedes.

	P. Baja	P. Media	P. Alta
Sede La Plata	2	2	2
Sede Pitalito	4	3	2
Sede Garzón		4	2

Fuente: Autor.

Para todos los sectores nombrados en donde se implementaron los módulos, se trató de que cada uno de éstos estuviera lo más alejado posible uno del otro, sobre todo en los sectores de campo abierto y semiabierto, ya que debido a su potencia generada se estaba evitando alguna perturbación debido a las ondas estacionarias que pudiesen aparecer, creando alguna interferencia o ruido indeseado producto de la cercanía y de la posición en donde pudiesen estar los bafles, es por eso que en la mayoría de sectores se hizo la instalación en las esquinas o rincones de los lugares seleccionados para que la onda sonora del streaming no realizara un rebote con el objeto en frente creando una reflexión del sonido generando un eco indeseable, empobreciendo la calidad de la reproducción.

Finalmente, cada sistema se dejó correctamente instalado y funcional, sujeto a la calidad de la señal de internet vía wifi. A continuación, se muestran evidencias de algunos sectores instalados en cada sede.

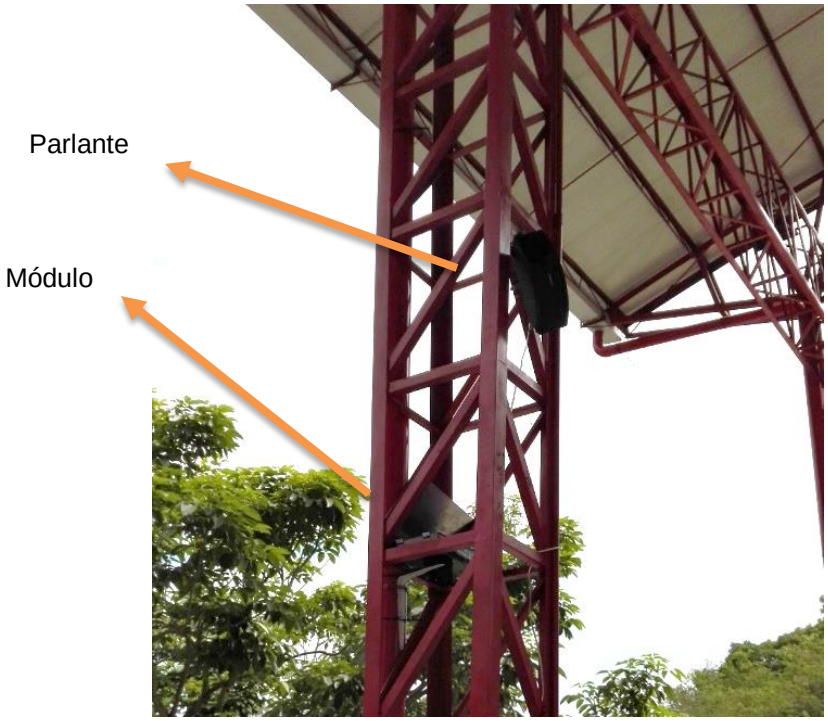
SEDE LA PLATA

Figura 64. Modulo en los sectores del pasillo, auditorio y cafetería.

Fuente: Óscar Forero M.

SEDE GARZÓN

Figura 65. Módulo en los sectores de cancha, parqueadero y pasillo.



Fuente: Autor

Figura 66. Módulo en los sectores de cafetería, auditorio y ágora.

Fuente: Autor

SEDE PITALITO

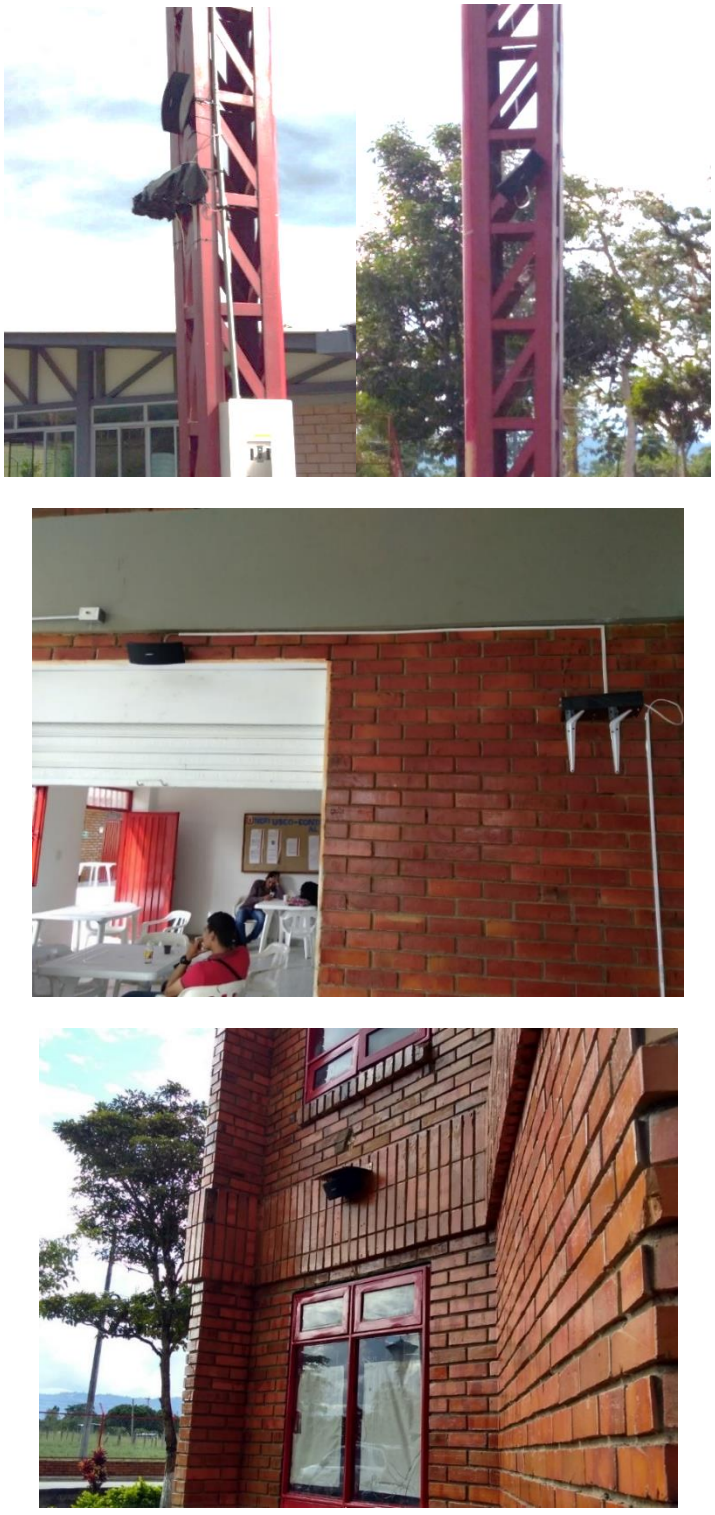
Figura 67. Módulo en los sectores de coordinación, pasillo y secretaría

Fuente: Autor

Figura 68. Módulo en sectores de oficina y hall en 1er y 2do piso

Fuente: Autor

Figura 69. Módulos en sectores de la cancha y cafetería



Fuente: Autor.

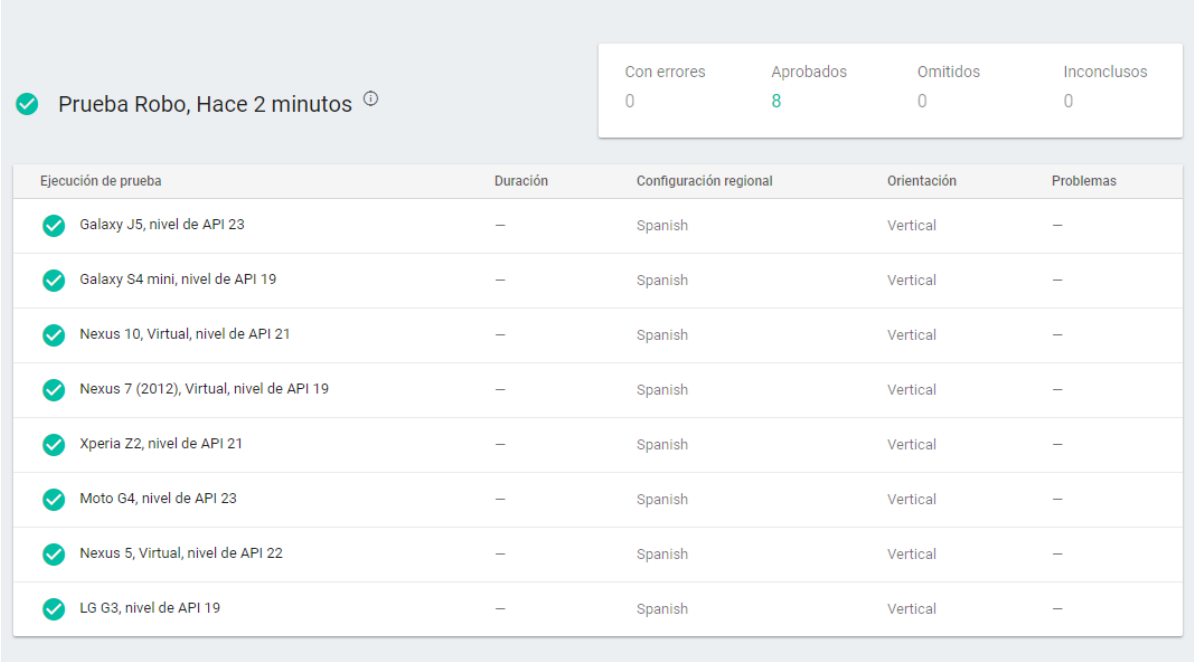
4.2 EVALUACIÓN DE DESEMPEÑO

App Móvil

Para la App móvil la plataforma de Firebase ofrece el servicio de “Laboratorio de pruebas (Test Lab)”, en donde se pueden probar las App realizadas en distintos equipos virtuales, con el objetivo de garantizar la calidad de la App, Firebase Test Lab for Android proporciona dispositivos físicos y virtuales para ejecutar pruebas que simulan entornos de uso reales. Brindando informes de prueba con resultados acerca de registros, capturas de pantalla e información útil a la hora de corregir cualquier problema que se presente y hacer una App móvil más estable.

Dentro de la prueba realizada se probaron 5 equipos reales y 3 virtuales con APIs distintas, para poder conocer si existía algún problema de compatibilidad o algo más.

Figura 70. Resultado de Test Lab en Firebase



Fuente: Autor.

App web

Dentro de la misma plataforma de productos de desarrollo de software que ofrece Google, se encuentra PageSpeed Insight, un producto encargado de medir el rendimiento de las páginas para dispositivos móviles y para ordenadores. Obtiene la URL dos veces, una vez con un agente de usuario para móviles, y otra con un agente de usuario para ordenadores.

La puntuación de PageSpeed va de 0 a 100 puntos. Cuanta más alta sea la puntuación, mejor. PageSpeed Insights mide cómo la página puede mejorar el rendimiento de los siguientes factores:

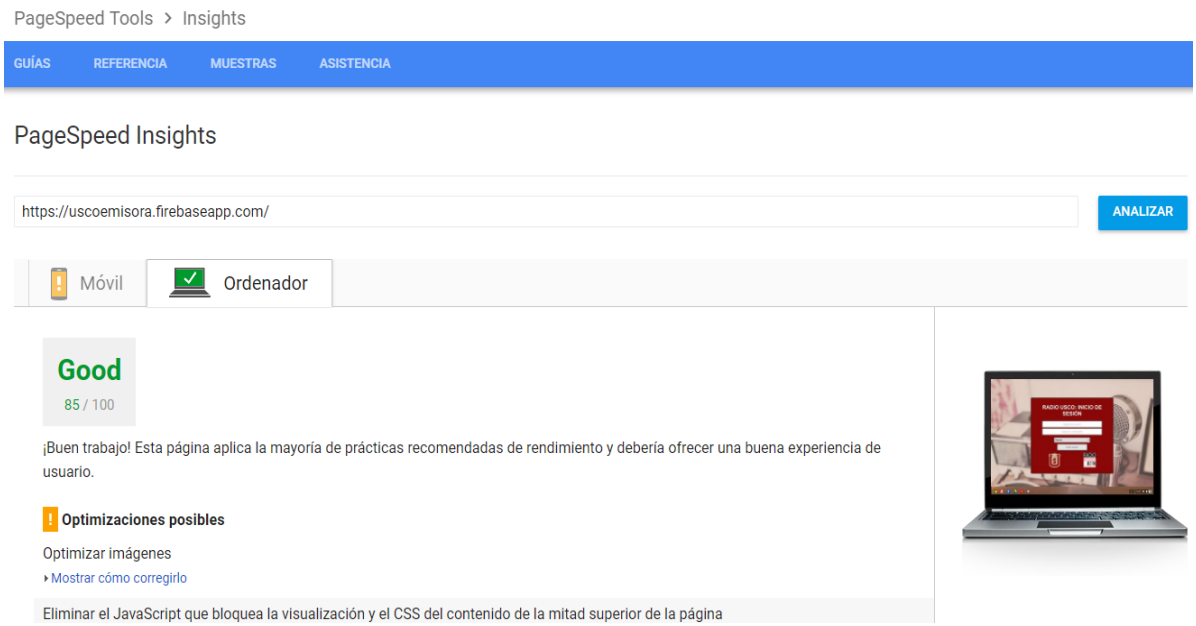
Tiempo de carga en la mitad superior de la página: tiempo transcurrido desde el momento en que un usuario solicita una página nueva hasta que el navegador muestra el contenido de la mitad superior.

Tiempo de carga completa de la página: tiempo transcurrido desde el momento en que un usuario solicita una página nueva hasta que se muestra completamente en el navegador.

Sin embargo, dado que el rendimiento de una conexión de red varía considerablemente, PageSpeed Insights solo tiene en cuenta los aspectos de rendimiento ajenos a la red: la configuración del servidor, la estructura HTML de la página y el uso de recursos externos como imágenes, JavaScript y CSS. No obstante, el rendimiento absoluto de la página dependerá de la conexión de red del usuario.

Para poder testear la App web dentro de esta plataforma, fue necesario utilizar la función “Hosting” de Firebase, arrojando el aplicativo en la siguiente dirección: https://uscoemisora.firebaseapp.com/

Figura 71. Prueba ejecutada en PageSpeed Insights



Fuente: Autor.

De modo específico, se realizaron pruebas a nivel de verificador de código para HTML y CSS en la página oficial de W3C[****]: https://validator.w3.org/ y https://jigsaw.w3.org/css-validator/ , respectivamente. Arrojando certificados de validez.

Figura 72. Validación para archivo css realizada

Fuente: Autor.

4.3 ANÁLISIS DE RESULTADOS

En cada una de las pruebas realizadas a los aplicativos para el sistema de sonido ambiental, se dieron buenos resultados. En el caso de la App móvil, con la prueba realizada en la plataforma de Firebase, se da un parte de seguridad al poderse ejecutar la app en varios dispositivos reales y virtuales con distintas características sin obtener problema alguno. Dicho resultado avala un buen funcionamiento y programación de la app. Existen otros programas que evalúan el desempeño de una app móvil, pero por motivos de costo se desistieron de ellos.

Mientras tanto, para la App web, los resultados hablan de un aplicativo amigable que ofrece una buena experiencia del usuario. Si bien la página de test arroja unas correcciones correspondientes a optimización de recursos almacenados en caché y abreviaciones en código css en archivos secundarios, es un buen puntaje el obtenido. La validación aportada por la W3C ayuda a comprobar que la programación realizada para el aplicativo es el adecuado y que cumple con las reglas según el estándar fijado.

[****] El World Wide Web Consortium (W3C) es una comunidad internacional que desarrolla estándares que aseguran el crecimiento de la Web a largo plazo.

En cuanto al sistema de sonido ambiental en general, se realizaron pruebas de funcionamiento que consistían en prender y apagar distintos sectores. Los problemas encontrados fueron relacionados con la velocidad del internet de la red wifi de las sedes, lo que ocasionaba retardo en la señal de más de 5min o incluso su no funcionamiento.

En las sedes de La Plata y Garzón, el servicio de internet es un poco más variable y dependiendo el flujo de usuarios, así mismo varía el funcionamiento del sistema. En la mayoría de puntos instalados se presenta el mismo inconveniente debido a la ubicación de los módulos en las sedes, sobre todo en las zonas alejadas de los router. Por ende, se realizó un oficio a rectoría informando de la situación con el aval de los coordinadores de cada sede, con el objetivo de tratar de gestionar una mejor solución al tema del servicio de internet y así poder garantizar el pleno funcionamiento de cada sistema de sonido ambiental en las sedes. (Véase Anexo M).

5. CONCLUSIONES

Las etapas de amplificación, trasmisión, difusión y acondicionamiento del sistema de sonido ambiental cuentan con elementos seleccionados específicamente con el objetivo de brindar los mejores resultados para la trasmisión del streaming. Cada elemento fue previamente comparado y escogido por cuestiones de calidad, prestaciones y costo.

La placa de desarrollo escogida-Raspberry Pi 3- fue programada en uno de los lenguajes con mayor auge en la actualidad, siendo Python un lenguaje potente, rápido, accesible en otras plataformas, fácil de aprender y con una gran comunidad open-source.

Se logra diseñar una App móvil sólida y amigable para el monitoreo del sistema de sonido ambiental. Realizada en API 17, versión 4.2 Jelly Bean de Android. Se escoge dicha versión ya que la mayoría de dispositivos móviles cuentan mínimamente con esa. Además, se realiza una App web a petición de los directivos de la emisora institucional.

Se implementan 21 módulos de sonido ambiental distribuidos en las tres (3) sedes de la universidad, de acuerdo con su potencia y área de aplicación. Fueron probadas en su totalidad y su capacidad de respuesta al monitoreo queda sujeta a la velocidad del servicio de internet de las sedes.

Se tiene un sistema completamente escalable que puede ser utilizado bien sea para conectar más altavoces o más amplificadores, lo que se requiera en caso de querer mayor cobertura de sonido.

El sistema de sonido ambiental realizado para cada sede cumple estructuralmente a cabalidad los requisitos para ser un sistema de sonido robusto con materiales que proporcionan un óptimo rendimiento.

6. RECOMENDACIONES

Según el Centro de Tecnologías de Información y Comunicaciones (CTIC), se cuenta con un canal de acceso a Internet de 50Mbps comprado por un año a la empresa Telefónica Telecom distribuidos mediante dispositivos de conectividad red wifi (Router) instalados por todas y cada una de las sedes, sin embargo se requiere mejorar el servicio de red inalámbrica en cada una de las sedes ya sea aumentando el ancho de banda proporcionado o creando redes privadas de uso exclusivo en cada sede, ya que debido a las demoras en conexión e inestabilidad de las mismas, el sistema de sonido ambiental no logra reproducir el streaming de la emisora adecuadamente.

La SBC adquirida – Raspberry Pi3- posee un defecto de fábrica en su salida de audio de 3.5mm, por ende, al reproducir cualquier archivo de audio se alcanza a escuchar un leve pitido, por ello se recomienda adaptarle una tarjeta de sonido USB disponible en el mercado para mejorar la calidad del sonido. Otra opción es realizar la conexión de salida de audio a través del puerto HDMI, consiguiendo adaptadores con dicho propósito.

Para mejorar la calidad de audio de los sistemas, se recomienda adquirir más parlantes para los módulos de potencia media y alta, instalados en espacios abiertos y semiabiertos; ya que debido a la cantidad de sectores y parlantes disponibles, se instalaron solo uno por sector. Si bien dicho arreglo funciona, cuando el nivel de ruido del entorno aumente, la intensidad del sonido saliente no será suficiente, aumentando la ininteligibilidad de la transmisión.

Se recomienda a largo plazo, cambiar la cajilla que contiene los materiales del sistema por un material más duradero, ya que, si bien el material escogido cumplía con requerimientos básicos, su uso a largo tiempo y exposición a condiciones ambientes variantes deterioran el mismo.

REFERENCIAS BIBLIOGRÁFICAS

Acerca de JavaScript [En línea]. *Mozilla Developer Network*. 2015. [Consultado: 13 junio 2017]. Disponible en: https://developer.mozilla.org/es/docs/Web/JavaScript/Acerca_de_JavaScript

Benjarano, M. *Tutorial 5 - Conexión remota al Raspberry Pi usando SSH (Secure Shell) [En línea]*. Frambuesa Pi Colombia - Raspberry Pi en español. 2013. [Consultado: 4 mayo 2017]. Disponible en: http://www.frambuesapi.co/2013/09/25/tutorial-5-conexion-remota-al-raspberry-pi-usando-ssh/

Cuello, J., & Vittone, J. *Diseñando apps para móviles [En línea]*. [Consultado: 15 de julio 2017]. Disponible en: http://appdesignbook.com/es/

Docs: VNC Connect and Raspberry Pi. RealVNC [En línea]. (2017). Realvnc.com. [Consultado: 4 abril 2017]. Disponible en: https://www.realvnc.com/docs/raspberry-pi.html

Fundamentals: Best MVC Practices. The Definitive Guide to Yii. Yii PHP Framework [En línea]. 2015. Yiiframework.com. [Consultado: 12 Junio 2017]. Disponible en: http://www.yiiframework.com/doc/guide/1.1/en/basics.best-practices

Funding Universe. *Company Histories [En línea].* [Consultado: 09 de mayo 2017]. Disponible en: http://www.fundinguniverse.com/company-histories/muzak-inc-history/

Gribnitz. *Dyna-Micro Single Board Computers*. VCF West Forum [En línea]. 2017. [Consultado: 10 de agosto 2017]. Disponible en: http://www.vcfed.org/forum/showthread.php?57918-Dyna-Micro-Single-Board-Computers

JSON Introduction [En línea]. W3schools.com. [Consultado: 12 junio 2017]. Disponible en: https://www.w3schools.com/js/js_json_intro.asp

Maggiolo, D. Propagación del sonido [En línea]. [Consultado: 23 de agosto 2017]. Disponible en: http://www.eumus.edu.uy/docentes/maggiolo/acuapu/prp.html

Martí, J. *Las músicas invisibles: la música ambiental como objeto de reflexión.* Trans. En Revista Transcultural de Música *[En línea],* 2002. [Consultado: 10 de mayo 2017]. Disponible en: http://www.redalyc.org/articulo.oa?id=82200611.

Marzal, A. & Gracia, I. (2003). *Introducción a la programación con Python*. (pp. 59-60).

Moreno, G. & Valdez, G. (2015). *Análisis, diseño e implementación de una aplicación móvil para el monitoreo en tiempo real de CCTV para dispositivos Android, haciendo uso de la red celular. Monitoreo remoto.* Ecuador, (pp. 25-29).

Newark, C. *History of the development kit*. *Embedded Computing Desing [En línea]*. 2015. [Consultado: 10 de junio 2017]. Disponible en: http://www.embedded-computing.com/embedded-computing-design/history-of-the-development-kit

NoSQL vs SQL: Principales diferencias y cuándo elegir cada una de ellas [En línea]. (2015). PandoraFMS. [Consultado: 13 de julio 2017]. Disponible en: https://blog.pandorafms.org/es/nosql-vs-sql-diferencias-y-cuando-elegir-cada-una/

Ortmeyer, C. *A Brief History of Single Board Computers*. Electronic Desing Uncovered [En línea]. 2014. p. 1-2. [Consultado: 12 de junio 2017]. Disponible en: http://www.newark.com/wcsstore/ExtendedSitesCatalogAssetStore/cms/asset/pdf/americas/common/NE14-ElectronicDesignUncovered-Dec14.pdf

Powers, N. *Configuración desatendida para su Raspberry Pi 3 [En línea]*. Arrow. 2016. [Consultado: 4 mayo 2017]. Disponible en: https://www.arrow.com/es-mx/research-and-events/articles/headless-setup-for-your-raspberry-pi-3

W3C España. *Sobre el W3C [En línea].* [Consultado: 24 de agosto 2017]. Disponible en: http://www.w3c.es/Consorcio/

¿Qué es el streaming? [En línea]. Streaming Colombia. [Consultado: 10 de abril 2017]. Disponible en: https://www.streamingcolombia.co/blog/qu%C3%A9-es-streaming/.

ANEXOS

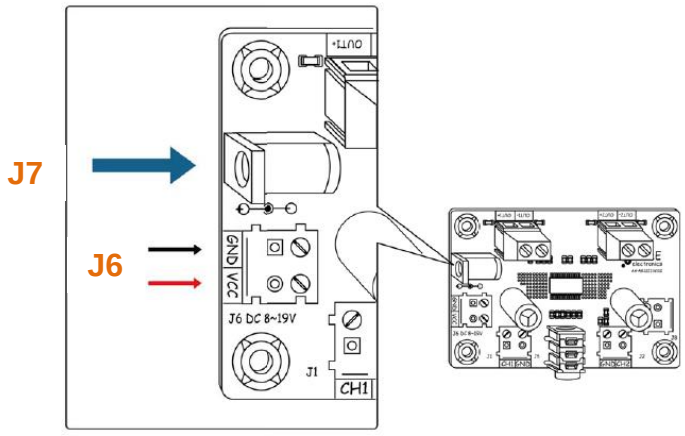
Anexo A. Especificaciones eléctricas del AA-AB32231 (8W).

Todos los parámetros son probados a TA=20°C, 12VDC, f=1KHz, entrada sinusoidal, RL = 4Ω, Ganancia= 20dB.

Parameter	Condition	Min.	Typ.	Max.
Supply Voltage	-	8V	12V	19V
Input Sensitivity	Gain=26dB	-	0.283V	-
S/N Ratio	Maximum output at THD+ N < 1%, f = 1 kHz, Gain=20db, A-weighted	-	102dB	-
THD+N	Vcc = 16 V, f = 1 kHz, Po = 7.5 W (half-power)		0.1%	-
	RL = 8 Ω, f = 1 kHz, Po = 5 W (half-power)		0.06%	-
Efficiency	Vcc=12V, RL=8Ω, Pout=5W	-	86%	-
	Vcc=12V, RL=8Ω, Pout=10W		89%	-
Input Impedance	-	-	30K ohm	-
Gain	-		26dB	
Output Power	THD+ N= 10%, f = 1 kHz, Vcc = 16 V	-	15W	-
	THD+N = 10%, f = 1 kHz; Vcc = 13 V	-	10W	-
Frequency Response	-	-	20HZ-20KHz (±3dB)	-
Load	-	3.2 ohm	4 ohm	-
Operating Temperature	-	-40℃	20℃	65℃
Storage Temperature	-	-60℃	20℃	105℃
Thermal Shutdown	-	-	150℃	-

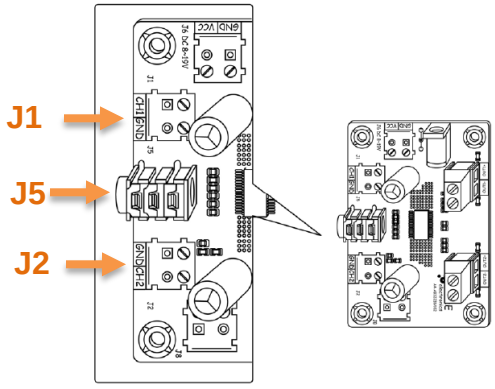
Anexo B. Conexiones para amplificador 8W

1. Fuente de poder



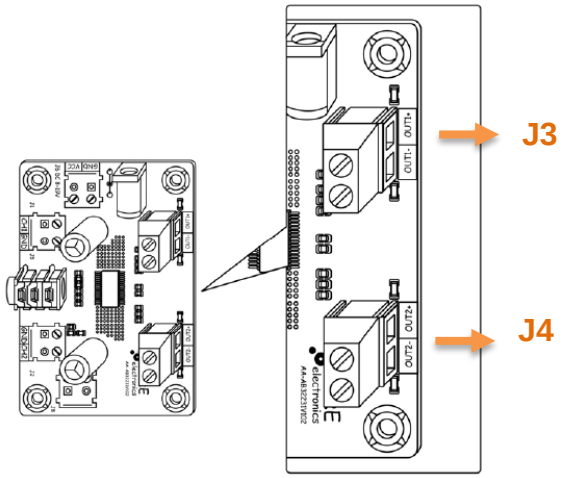
Conector		Descripción
Jack	J7	Fuente de voltaje de 8V-19V DC
Bloque de terminal	J6	VCC GND

2. Entradas:



Conector		Descripción
Jack 3.5mm de audio	J5	Entrada Estéreo
Bloques de terminal	J1	CH1 GND
	J2	CH2 GND

3. Salidas:



Conector	
Bloques de terminal	J3, J4

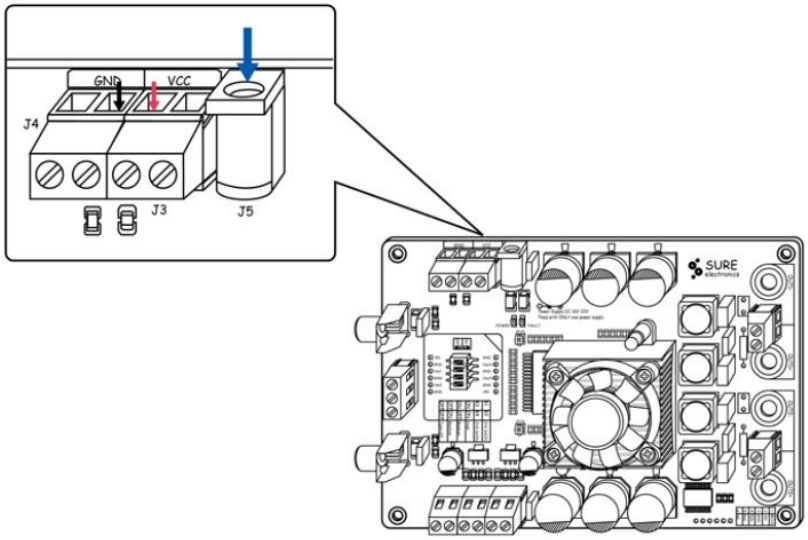
Anexo C. Especificaciones eléctricas del AA-AB32174 (50W)

Parameter	Condition	Min.	Typ.	Max.
Supply Voltage	AA-AB32189	14V	36V	40V
	AA-AB32174		24V	27V
	AA-AB32165		16V	19V
Quiescent Current （Powered by 19V）	FAN ON,STBY Disable	-	90mA	-
	FAN ON,STBY Enable	-	50mA	-
	FAN OFF,STBY Disable	-	40 mA	
	FAN OFF,STBY Enable		10 mA	
Input Sensitivity （AA-AB32174）	25.6dB	-	910mV	-
	31.6dB		450mV	
	35.1dB		300mV	
	37.6dB		220mV	

Gain(SW1 Setting)	K1 ON, K2 ON	24.6	25.6	26.6
	K1 ON, K2 OFF	30.6	31.6	32.6
	K1 OFF, K2 ON	34.1	35.1	36.1
	K1 OFF, K2 OFF	36.6	37.6	38.6
Frequency Range	-	20Hz to 20KHz（±3dB）		
Efficiency	Both channels output rating power.	-	>90%	-
Input Impedance	-	48K ohm	60K ohm	-
Load	-	-	6 ohm	-
Operating Temperature	-	0℃	20℃	50℃

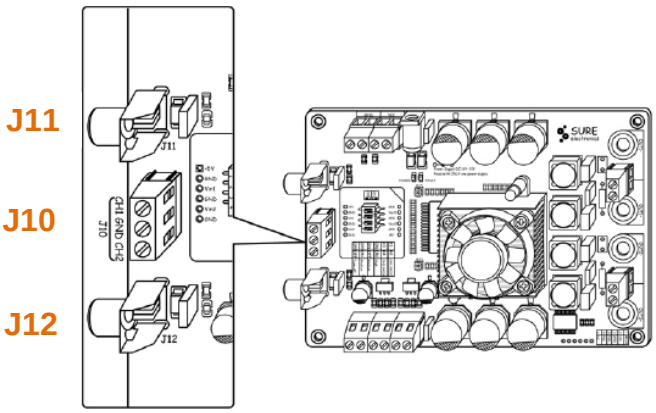
Anexo D. Conexiones del amplificador 50W

1. Fuente de poder:



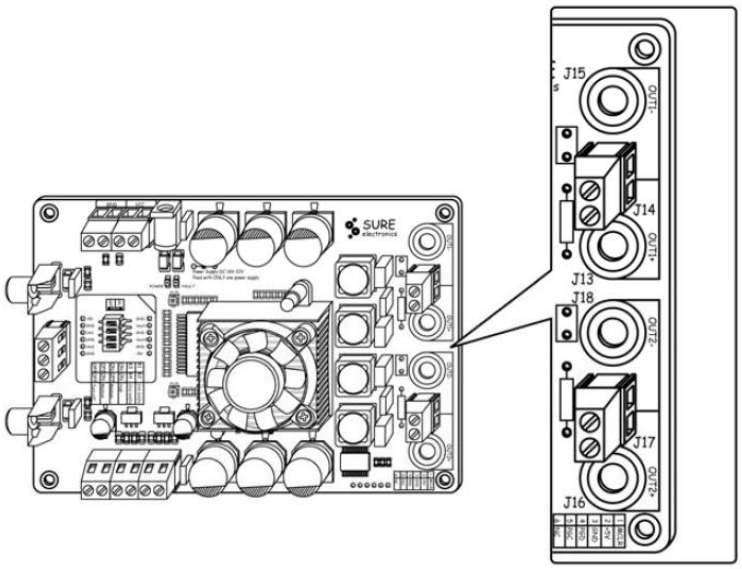
Conector		Descripción
Jack	J5	Enchufe de fuente de voltaje DC
Bloque de terminal	J4	VCC GND

2. Entradas:



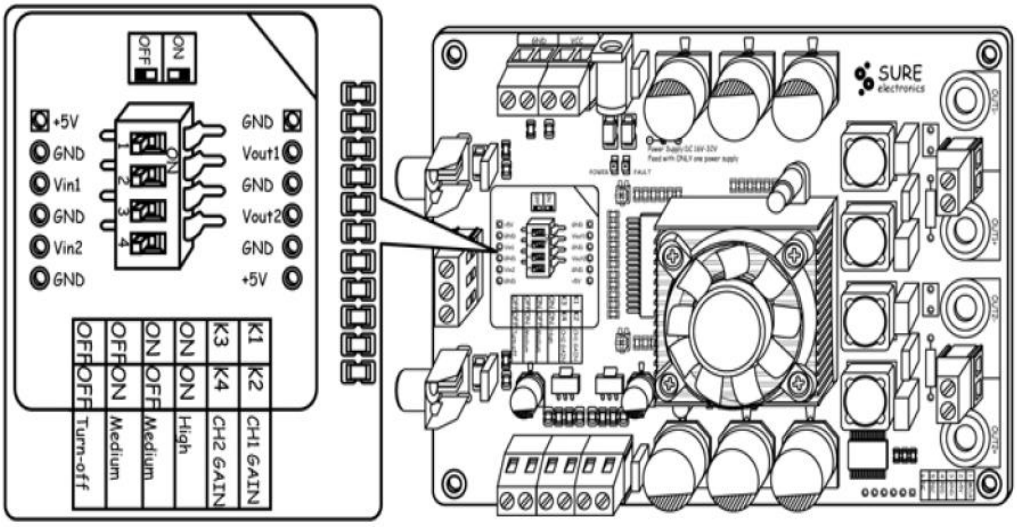
Conector		Descripción
Conector RCA	J11/J12	Entrada Canal 1/Canal 2
Bloques de terminal	J10	CH1 GND
	J10	CH2 GND

3. Salida:



Conector	
Conector RCA	J15, J13, J18, J16
Bloques de terminal	J14, J17

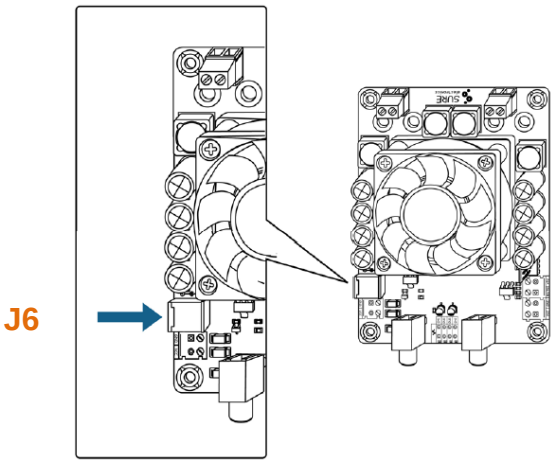
4. Control de volumen:



Anexo E. Especificaciones eléctricas del AA-AB32186 (100W)

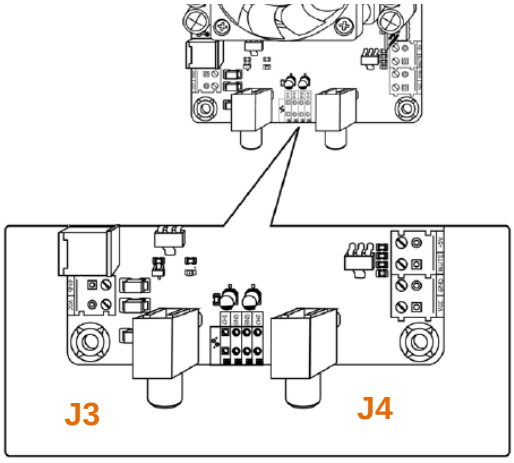
Parameter	Condition	Min.	Typ.	Max.
Supply Voltage	-	10V	27V	30V
Input Sensitivity	Pout=100W	-	520mV	-
Quiescent Power (W)	Vmute=0V VCC=30V	-	6.6	-
Standby Power (W)	Vmute=5V VCC=30V		3.6	
Efficiency	Pout=100W	-	95%	-
Input Impedance	-	-	10K ohm	-
Gain（dB）			32	
Output Power (W)	RL=4 ohm VCC=30V	-	1%THD+N 77.5 10%THD+N 105	-
Frequency Response (dB)	VCC=30V	-3	0.26	3
Load	-	3.2 ohm	4 ohm	-
Operating Temperature	-	0℃	20℃	70℃
Storage Temperature	-	-20℃	20℃	105℃
Thermal Shutdown	-	-	150℃	-

Anexo F. Conexiones del amplificador 100W



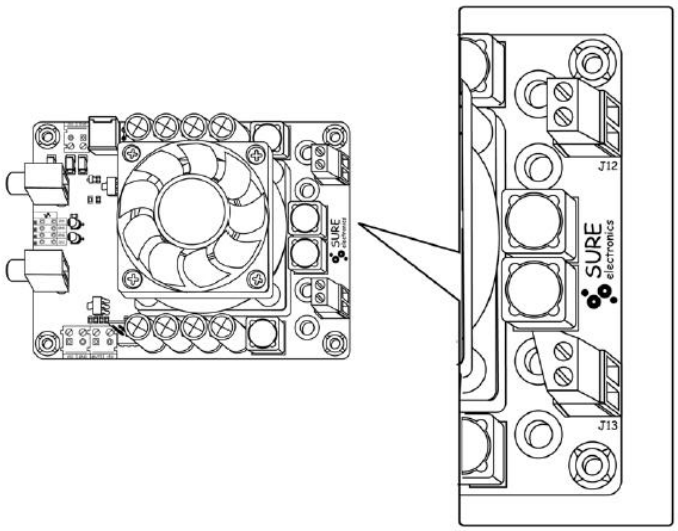
Conector		Descripción
Jack	J6	Enchufe de entrada de fuente de poder

2. Entradas:



Conector	
RCA	J3, J4

3. Salidas:



Conector		Descripción
Bloques de terminal	J12	Salida del canal 1
	J13	Salida del canal 2

Anexo G. Instalación de Raspbian OS en Raspberry Pi 3.

Para la instalación del Rasbian OS se ha elegido utilizar el gestor de NOOBS versión 2.0.0. Se configura una instalación silenciosa, editando el archivo *recover.cmdline* en el directorio raíz de la tarjeta micro SD, agregando *silentinstall*:

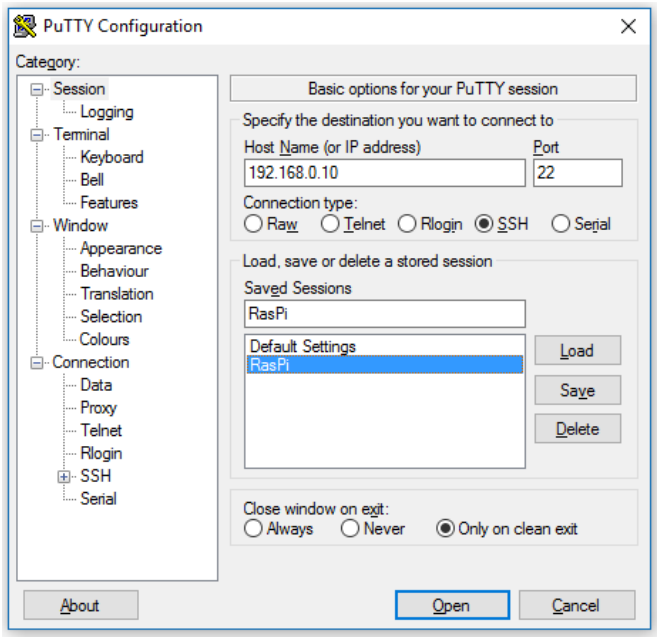
runinstaller quiet ramdisk_size=32768 root=/dev/ram0 init=/init vt.cur_default=1 elevator=deadline silentinstall

Teniendo en cuenta que no se dispone de un monitor o tv con entradas HDMI, se procede a realizar una conexión remota usando SSH.

Es necesario instalar el cliente SSH en Windows, en este caso *PuTTY*, el cual se puede descargar a través de: http://www.putty.org/.

En la interfaz de PuTTY se configura el acceso a la tarjeta, lo único importante es conocer la dirección IP asignada para la Raspberry Pi 3 dentro de la red.

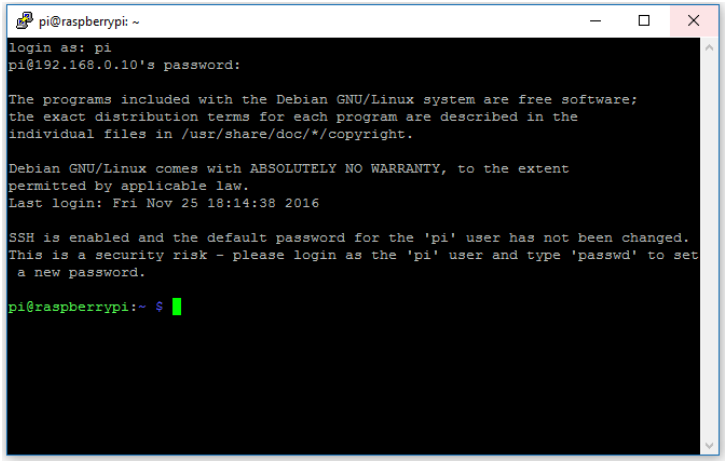
Figura 73. Configuración en Putty



Fuente: Autor.

Si la conexión es exitosa, aparecerá una ventana en línea de comandos pidiendo el nombre del usuario y la contraseña, por defecto el usuario es *pi* y la contraseña es *raspberry*.

Figura 74. Terminal a través de conexión SSH



Fuente: Autor.

Ahora se está conectado a la Raspberry Pi, tenga en cuenta que esta es una conexión ASCII, es decir Putty no ha sido configurado para activar el modo gráfico

en modo remoto, para esto es necesario descargar otros programas en el PC. Por ello, para activar el modo gráfico se utiliza VNC.

Conexión remota al Raspberry Pi usando VNC

Se utiliza *VNC* (acrónimo en inglés de *Virtual Network Computing*) que es una aplicación cliente-servidor. Una donde el servidor VNC envía "fotos" del escritorio al computador remoto, varias veces por segundo.

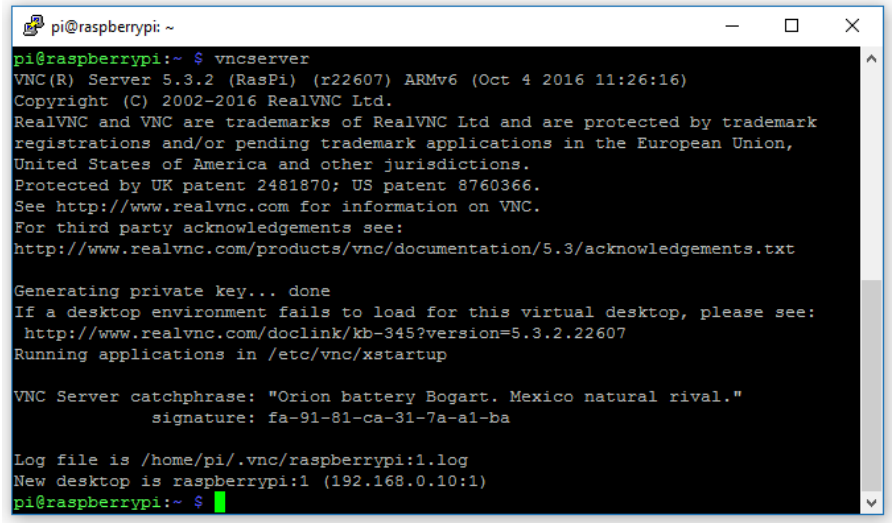
Para instalar el servidor VNC en la tarjeta Raspberry Pi 3, es necesario acceder al terminal e ingresar el comando:

sudo apt-get install realvnc-vnc-server

Una vez ha sido instalado, es necesario iniciar el servicio, con el comando:

vncserver

Figura 75. Creación de sesión VNC

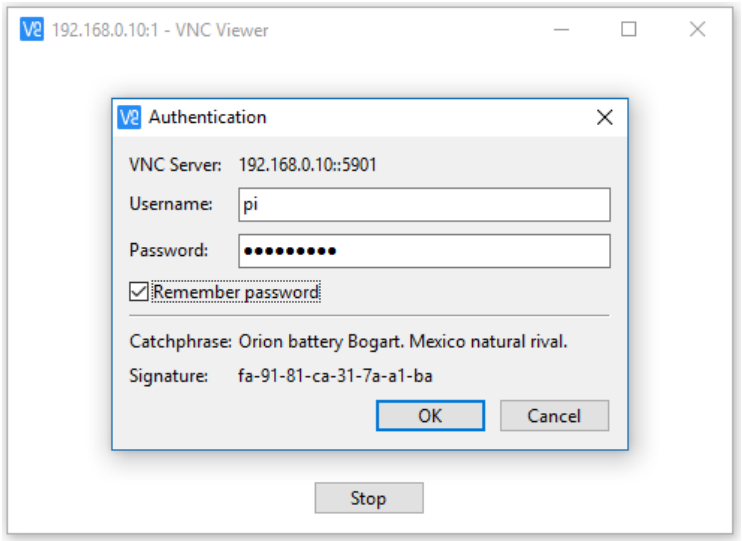


Fuente: Autor.

Luego se descarga la versión de VNC Viewer para pc, desde donde se maneja remotamente la RasPi. Disponible en la página oficial: https://www.realvnc.com/download/vnc/

A continuación, desde la computadora, se ingresa la dirección establecida para acceder al escritorio remoto y se ingresan la clave de ingreso previamente asignada.

Figura 76. Datos de login en VNC Viewer



Fuente: Autor.

Una vez creada la sesión, se abre la ventana de VNC donde se muestra el entorno de escritorio PIXEL de la versión de Raspbian.

Anexo H. Script principal: Main.py

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
# Se llama las librerías y los Scripts Secundarios
import vlc
import alsaaudio
import sys
import signal
import time
from clases.Conexion import Conexion
from clases.Conexion2 import Conexion2
from clases.Conexion3 import Conexion3
from clases.Control import Control
from firebase import firebase

def psede(respues):
    global res
    res = respues

# Se recibe el valor del Script Secundario para el Volumen Sectorizado
```

```
def procesa3(respuesta3):
    global volumen2
    global m2

    volumen2=int(float(respuesta3))

    print (respuesta3)
    m2 = alsaaudio.Mixer('PCM')

    while res == True:
        m2.setvolume(volumen2)
        break

# Se recibe el valor del Script Secundario para el Volumen General
def procesa2(respuesta2):
    global volumen
    global m

    volumen=int(float(respuesta2))

    print (respuesta2)
    m = alsaaudio.Mixer('PCM')

    while res == True:
        m.setvolume(volumen)
        break

# Se recibe la orden del Script Secundario para habilitar o inhabilitar el valor del
Volumen General
def procesa4(respuesta4):
    print (respuesta4)
    global resp
    resp = respuesta4

# Si se decide habilitar el Volumen General, en los altavoces se utilizará el valor del
Volumen General
    if respuesta4 == (True):
        m = alsaaudio.Mixer('PCM')
        while res == True:
            m.setvolume(volumen)
            print (volumen)
            break

# Si se decide inhabilitar el Volumen General, en los altavoces se utilizará el valor
del Volumen Sectorizado
```

```
    if respuesta4 == (False):
        m2 = alsaaudio.Mixer('PCM')
        while res == True:
            m2.setvolume(volumen2)
            print (volumen2)
            break

# Se recibe el valor del Script Secundario para habilitar o inhabilitar el Streaming
def procesa(respuesta):

    if respuesta == (False):
        m0 = alsaaudio.Mixer('PCM')
        m0.setvolume(0)
        print (respuesta)

    if respuesta == (True):

        if resp == (True):
            m = alsaaudio.Mixer('PCM')
            m.setvolume(volumen)
            print (volumen)

        if resp == (False):
            m2 = alsaaudio.Mixer('PCM')
            m2.setvolume(volumen2)
            print (volumen2)

    sys.stdout.flush()
    sys.stdout.flush()
    sys.stdout.flush()
    sys.stdout.flush()
    sys.stdout.flush()

def em():

    while True:

        # URL en donde se encuentra alojado el Streaming de la emisora

        p = vlc.MediaPlayer("http://54.187.68.216:8080/")
        p.play()

        break
    sys.stdout.flush()
```

```
try:
        print ("Inicio")

        a = Conexion(psede)
        x = Conexion3(procesa3)
        w = Conexion2(procesa2)
        y = Control(procesa4)
        t = Conexion(procesa)
        z = em()

        a.daemon=True
        x.daemon=True
        w.daemon=True
        y.daemon=True
        t.daemon=True

        a.start()
        x.start()
        w.start()
        y.start()
        t.start()

        signal.pause()
except (KeyboardInterrupt, SystemExit):
        raise
        print ("Salida")
```

Anexo I. Script Secundario: Conexión.py

```
# -*- coding: utf-8 -*-
from firebase import firebase
import threading
import time

# Conexión con la base de datos para habilitar o inhabilitar el Streaming
class Conexion(threading.Thread):

   def __init__(self, cb):
      threading.Thread.__init__(self)
      self.callback = cb
      self.fire  =  firebase.FirebaseApplication('https://uscoemisora.firebaseio.com/',
None)
      self.ultimo_estado = self.fire.get('/sede/laplata/lp', None)
```

```
        self.callback(self.ultimo_estado)

# Refresca o actualiza la conexión si se detecta un cambio en la orden
    def run(self):
                E = []
                E.append(self.ultimo_estado)
                i = 0

                while True:
                        estado_actual = self.fire.get('/sede/laplata/lp', None)
                        E.append(estado_actual)

                        if E[i] != E[-1]:
                                self.callback(estado_actual)

                        del E[0]
                        i = i+i
                        time.sleep(0.1)
```

Anexo J. Script secundario: Conexion2.py

```
# -*- coding: utf-8 -*-
from firebase import firebase
import threading
import time

# Conexión con la base de datos para recibir el valor del Volumen General
class Conexion2(threading.Thread):

    def __init__(self, cb):
        threading.Thread.__init__(self)
        self.callback = cb
        self.fire  =  firebase.FirebaseApplication('https://uscoemisora.firebaseio.com/',
None)
        self.ultimo_estado = self.fire.get('/volumen/laplata/lp', None)
        self.callback(self.ultimo_estado)

# Refresca o actualiza la conexión si se detecta un cambio en el valor
    def run(self):
                E = []
                E.append(self.ultimo_estado)
                i = 0

                while True:
```

```
                    estado_actual = self.fire.get('/volumen/laplata/lp', None)
                    E.append(estado_actual)

                    if E[i] != E[-1]:
                            self.callback(estado_actual)

                    del E[0]
                    i = i+i
                    time.sleep(0.2)
```

Anexo K. Script secundario: Conexion3.py

```
# -*- coding: utf-8 -*-
from firebase import firebase
import threading
import time

# Conexión con la base de datos para recibir el valor del Volumen Sectorizado
class Conexion3(threading.Thread):

    def __init__(self, cb):
        threading.Thread.__init__(self)
        self.callback = cb
        self.fire  =  firebase.FirebaseApplication('https://uscoemisora.firebaseio.com/',
None)
        self.ultimo_estado = self.fire.get('/sectores1/laplata1/lp1', None)
        self.callback(self.ultimo_estado)

# Refresca o actualiza la conexión si se detecta un cambio en el valor
    def run(self):
            E = []
            E.append(self.ultimo_estado)
            i = 0

            while True:
                    estado_actual = self.fire.get('/sectores1/laplata1/lp1', None)
                    E.append(estado_actual)

                    if E[i] != E[-1]:
                            self.callback(estado_actual)

                    del E[0]
                    i = i+i
```

```
                time.sleep(0.2)
```

Anexo L. Script secundario: Control.py

```
# -*- coding: utf-8 -*-
from firebase import firebase
import threading
import time

# Conexión con la base de datos para habilitar o inhabilitar la asignación del Volumen General
class Control(threading.Thread):

    def __init__(self, cb):
        threading.Thread.__init__(self)
        self.callback = cb
        self.fire  =  firebase.FirebaseApplication('https://uscoemisora.firebaseio.com/', None)
        self.ultimo_estado = self.fire.get('/actv/laplata/lp', None)
        self.callback(self.ultimo_estado)

# Refresca o actualiza la conexión si se detecta un cambio en la orden
    def run(self):
            E = []
            E.append(self.ultimo_estado)
            i = 0

            while True:
                estado_actual = self.fire.get('/actv/laplata/lp', None)
                E.append(estado_actual)

                if E[i] !=E[-1]:
                    self.callback(estado_actual)

                del E[0]
                i = i+i
                time.sleep(0.2)
```

Anexo M. Copia de carta enviada a Rectoría.

UNIVERSIDAD SURCOLOMBIANA USCO
NIT. 891.180.084-2
RECIBIDO 23 OCT 2017
Fecha:
Hora: 2:19 p.m
Nº radicado: 016160
Recibido: Juan Ramirez
Folios: 4

Neiva, 23 de Octubre de 2017.

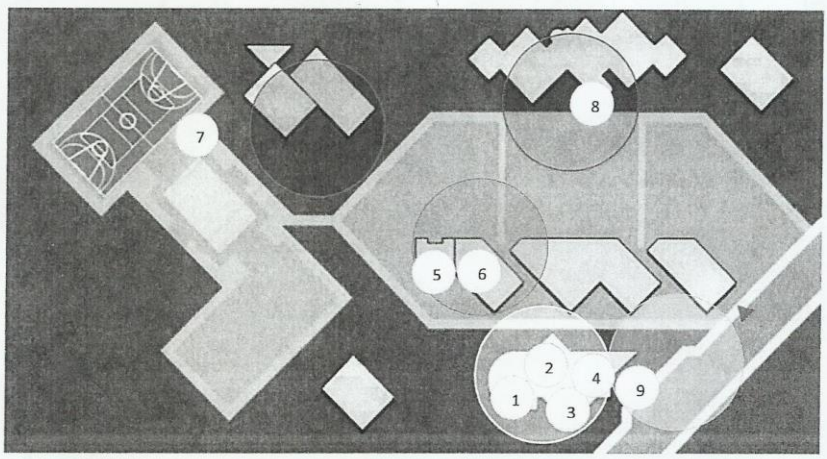
Doctor:
PEDRO LEÓN REYES GASPAR
Rector Universidad Surcolombiana
Ciudad.

Asunto. Informe sobre el proyecto de sistema de retransmisión de la emisora institucional en las sedes de la universidad.

Respetuoso saludo,

Como es de su conocimiento, la emisora institucional en compañía del pregrado de ingeniería electrónica de la universidad llevó a cabo la instalación de los sistemas de sonido ambiental en cada una de las sedes en la pasada semana comprendida entre los días 31 de Julio al 04 de agosto del presente año. Dicha instalación fue realizada en compañía del director de la emisora, el Sr Óscar Iván Forero, quien estuvo presente durante todo el proceso.

Los días 31 de Julio y 1 de agosto se realizó la instalación de 9 módulos en la sede de Pitalito, distribuidos de la siguiente manera:



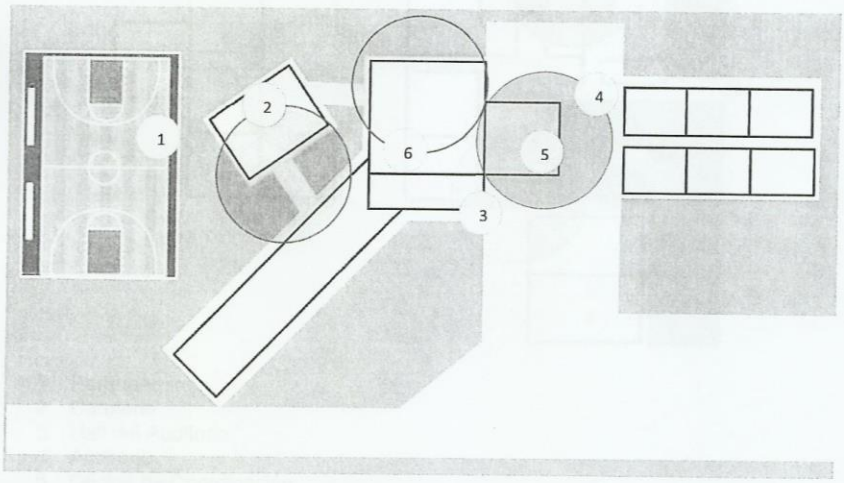
1. Oficina de coordinación
2. Pasillo de servicio médico
3. Oficina de secretaría
4. Oficina de biblioteca
5. Pasillo del baño - Primer piso
6. Pasillo del baño- Segundo piso
7. Campo deportivo cubierto
8. Restaurante

9. Parqueadero

Dentro de las pruebas de funcionamiento realizadas, nos encontramos con sectores que no servían en su totalidad debido a la intensidad de señal de internet recibida, pues cada módulo del sistema depende directamente de la calidad del servicio de internet wifi. En los sectores como el del campo deportivo la señal es baja y muy intermitente, mientras que en los módulos ubicados en los bloques, éstos presentan variaciones en la señal, de acuerdo al uso que se le esté dando a la red wifi disponible para cada sector. Lo anterior no permite el correcto funcionamiento de todo el sistema dispuesto.

La entrega de los módulos instalados se hizo al Sr *Óscar Andrés Torres,* (Auxiliar Administrativo), con quien se dejó el respectivo software de monitoreo para el sistema de sonido.

En la sede de Garzón se instalaron 6 módulos del sistema de sonido los días 02 y parte del 03 de agosto, distribuidos de la siguiente manera:



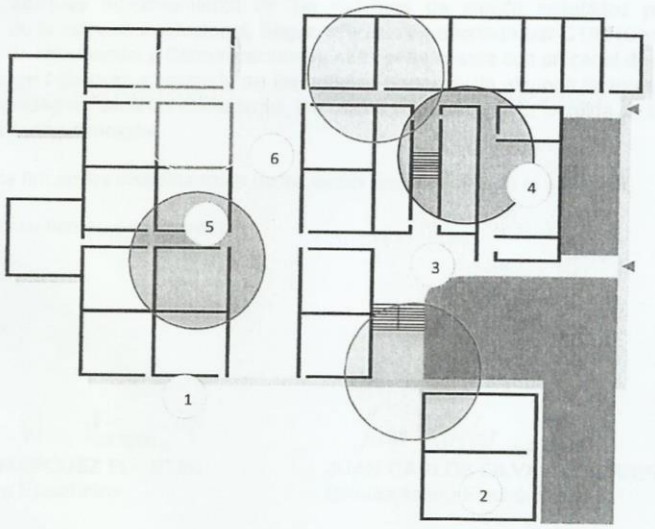
1. Campo Deportivo
2. Cafetería
3. Parqueadero
4. Ágora
5. Auditorio
6. Pasillo Secretaría

Una vez instalados los módulos, se realizaron pruebas de desempeño; obteniendo mejores resultados cuando cada una de las redes wifi se encontraban descongestionadas. Ya que al momento de contar con estudiantes que hacían uso del servicio, la señal se perdía y no se podía lograr retransmitir la emisora institucional. Además, la cobertura que tienen las redes wifi dentro de la sede es reducida, por ende tuvimos problemas para poder conectar

los módulos a la red, sobre todo en los sectores del parqueadero y la cancha. De modo que el sistema de sonido para la retransmisión de la emisora se ve impedido en su funcionamiento si hay demanda por parte de los estudiantes y demás usuarios en la sede.

La entrega de los módulos instalados se hizo al Sr *Felipe Manjarrez* y se dejó el respectivo software de monitoreo para el sistema de sonido en equipos de secretaría y coordinación.

Terminando la semana, parte del día 03 y todo el 04 de agosto se realizó la última instalación en la sede de La Plata, dejando 6 módulos ubicados de la siguiente manera:



1. Parqueadero
2. Cafetería
3. Hall del Auditorio
4. Auditorio
5. Oficina de Coordinación
6. Corredor Principal (Pasillo Principal)

Durante la prueba de los módulos presenciamos inconvenientes relacionados con la señal de internet y se puede asegurar que, según lo experimentado, ésta sede cuenta con el servicio de internet wifi más inestable de todas. Al igual que la sede de Garzón, la sede de La Plata tiene coberturas reducidas de sus redes y cuenta con bastante intermitencia. Por ende, todo el sistema de retrasmisión funcionó de manera regular.

Finalmente, la entrega de los módulos instalados se hizo a la Sra *Angélica Cruz Rojas* y se dejó el respectivo software de monitoreo para el sistema de sonido. Concluyendo la implementación de todo el proyecto.

Cabe resaltar que los planos aquí mostrados son resultado de la experimentación que se tuvo al momento de instalar los sistemas de sonido, por lo tanto, la cobertura de cada red inalámbrica es considerada una aproximación.

De antemano Sr Rector, agradecemos totalmente el apoyo brindado a este proyecto y ponemos en su conocimiento la principal preocupación que nos concierne con respecto a la calidad de la señal de red wifi ofrecida en cada una de las sedes, ya que de ella depende el correcto funcionamiento de los sistemas de sonido instalados para la retrasmisión de la emisora institucional. Según información aportada por CTIC (Centro de Tecnologías de Información y Comunicaciones), cada sede cuenta con un canal de 50 MB de internet cuya cobertura y potencia de las antenas depende de algunos factores como paredes y cantidad de usuarios conectados, los cuales pueden generar perdida de señal y no tener una buena conexión.

En constancia firman los coordinadores de las sedes conocedores de la situación.

Agradecemos su tiempo y colaboración.

Atentamente,

Luisa Fda. Márquez P.
LUISA FDA MÁRQUEZ PUENTES
Estudiante Ing Electrónica

Juan Carlos Silva
JUAN CARLOS SILVA GUTIERREZ
Estudiante Ing Electrónica

Coordinadores:

ANGÉLICA CRUZ
Sede La Plata

FELIPE MANJARREZ
Sede Garzón

Oscar Andrés Torres
ÓSCAR A. TORRES
Sede Pitalito