Automatización de mandos ir

Farney Camilo Serrato Villamizar

Universitaria Agustiniana
Facultad de Ingenierías
Programa de Ingeniería en Telecomunicaciones
Bogotá, D.C.
2020

Automatización de mandos ir

Farney Camilo Serrato Villamizar

Director
Guillermo Fernando Valencia Plata

Trabajo de grado para optar el título de Ingeniero en Telecomunicaciones

Universitaria Agustiniana
Facultad de Ingenierías
Programa de Ingeniería en Telecomunicaciones
Bogotá, D.C.
2020

Dedicatoria

A mi madre que ha sido mi mayor ejemplo de
superación y perseverancia
para culminar con éxito este pregrado.
A mis hermanos para que tomen como ejemplo el
esfuerzo y dedicación con la que lleve este proceso.
A mis familiares que siempre estuvieron pendiente de
mi formación y fueron de gran apoyo personal.

Resumen

El presente documento plasma el desarrollo de un sistema de mandos IR para controlar electrodomésticos remotamente desde una APP para sistemas operativos Android, recopila la información sobre los protocolos de mandos IR más conocidos entre fabricantes de electrodomésticos, se usó la plataforma Firebase y su función de base de datos en vivo para guardar los datos que se envían desde una aplicación móvil la cual fue desarrollada en el entorno APP Inventor 2, para la ejecución o envió de los pulsos IR se usó un dispositivo WeMos D1 R32 encargándose de leer los datos que están almacenados en la base datos de Firebase y sensores de recepción y transmisión infrarrojos, como referencia para el desarrollo del prototipo de mandos IR se tomó de la información recopilada el protocolo NEC el cual es usado por la marca SAMSUNG para el control de sus electrodomésticos.

Agradecimientos

Al Ingeniero Guillermo Fernando Valencia Plata por todo el apoyo brindado durante el desarrollo de este proyecto de grado, por brindarme todos sus conocimientos y guiarme para que todo lo planteado desde el inicio haya funcionado de manera correcta.

Tabla de contenidos

Idea de proyecto 10
Planteamiento del problema 11
Objetivos 12
Objetivo General 12
Objetivos específicos 12
Justificación 13
Marco Referencial 14
Estado del Arte 14
Marco Teórico 15
MIT App Inventor: 15
Firebase 16
Funciones de Firebase 17
Desarrollo 17
Realtime database. 17
Autenticación de usuarios. 17
Almacenamiento en la nube 18
Crash Reporting 18
Test Lab 18
Remote Config. 18
Modulo receptor IR 18
Modulo Emisor IR 20
Marco Legal 20
Normas Nacionales. 20
Metodología 23
Presupuesto 24
Cronograma 25
Protocolos de IR, configuración del servidor y plataforma de desarrollo de aplicaciones móviles Firebase 26
Protocolo RC5 27
Protocolo RC6 28

características del protocolo RC6: 28
Protocolo IR de transmisión Sony SIRC 29
Características protocolo SIRC 29
Servidor Firebase 30
Inicio Firebase 30
Desarrollo del sistema para enviar comandos IR 34
Codificación en Arduino 42
Librerías del código 42
Conexiones de los dispositivos 45
Pruebas de funcionamiento del sistema 46
Conclusiones 58
Referencias 59

Lista de figuras

Figura 1. Características de Frecuencia y sensibilidad espectral. 19
Figura 2. Ejemplo de marco de mensaje que utiliza el protocolo de transmisión NEC IR.. 27
Figura 3. Ejemplo de marco de mensaje con el protocolo de transmisión de infrarrojos Philips RC5…….....28
Figura 4. Descripción grafica protocolo SIRC.. 29
Figura 5. Consola Firebase 30
Figura 6. Espacio de trabajo proyecto Firebase.....31
Figura 7. URL de la Base de Datos 31
Figura 8. Clave Secreta Firebase 32
Figura 9. Menú cuenta de servicios 32
Figura 10. Creación del Proyecto 34
Figura 11. Screen de Inicio.....34
Figura 12. Screen Principal 35
Figura 13. Screen Lectura IR.....35
Figura 14. Selector del campo IR 36
Figura 15. Screen carga de lectura IR 36
Figura 16. Screen instrucciones.....37
Figura 17. Screen configuraciones 37
Figura 18. Screen Acerca de.....38
Figura 19. Ingreso clave secreta y URL Firebase.....38
Figura 20. Desarrollo de programación en bloques Menú Instrucciones 39
Figura 21. Desarrollo de programación en bloque Menú Instrucciones 2 39
Figura 22. Lógica Botón Lectura IR.....40
Figura 23. Lógica Botón Instrucción IoT 41
Figura 24. Lógica Botón Configuraciones del Sistema 41
Figura 25. Lógica Botón Acerca de.....42
Figura 26. Sintaxis librería RTClib.h. 43
Figura 27. Sintaxis librería EEPROM.h. 43
Figura 28. Sintaxis librería Wifi.h.. 43
Figura 29. Sintaxis librería Firebase ESP32 43

Figura 30. Función readCode conteo tiempo de los pulsos bajos 44
Figura 31. Función readCode conteo tiempo de los pulsos altos 45
Figura 32. Conexiones hardware. 46
Figura 33. Topología de la red y flujo de datos 46
Figura 34. Logo Auto_IR 47
Figura 35. Indicación para ingresar al menú configuraciones del sistema. 48
Figura 36. Menú configuraciones del sistema 48
Figura 37. Configuración hora del sistema 49
Figura 38. Configuracion Fecha del sistema 49
Figura 39. Prueba configuraciones del sistema monitor serial Ardunio. 50
Figura 40. Indicación para ingresar al Menú Lectura IR 50
Figura 41. Menú lectura IR 51
Figura 42. Selector de Campo IR 51
Figura 43. Confirmación selección campo IR 52
Figura 44. Menú carga de lectura IR 52
Figura 45. Pruebas campo seleccionado monitor serial Arduino. 53
Figura 46. Indicación para ingresar al menú IoT 54
Figura 47. Menú instrucciones o IoT 54
Figura 48. Selector para configuración de fecha IoT 55
Figura 49. Selector para configuración de hora IoT 55
Figura 50. Selector de campo IR IoT. 56
Figura 51. Configuraciones realizadas IoT 56
Figura 52. Prueba IoT monitor serial Arduino. 57

Idea de proyecto

Realizar un sistema de automatización mediante mando IR donde se pueda programar el encendido, apagado de cualquier equipo que cuente con esa tecnología, se implementara una aplicación móvil para sistemas operativos Android donde el usuario podrá ingresar la configuración de cualquier control IR y donde pueda programar la hora en la que desee encender y apagar los dispositivos entre otras funciones.

Planteamiento del problema

Las tecnologías están evolucionando cada vez más y la sistematización de diferentes dispositivos como televisores, DVD, aires acondicionados, control de cortinas, chimeneas entre otros más se está convirtiendo en la idealización del hogar inteligente de todos, en la actualidad hay variedad de dispositivos que permiten integrar todos estos e incluso se encuentran empresas que desarrollan la solución de acuerdo al requerimiento del cliente, estos sistemas tienden a ser complejos tanto para configurar como para instalar y también son costosos, la gran mayoría de estos equipos tienen sistemas de control por IR, según (GARCIA, 2014)“ El funcionamiento del control remoto, se basa en la emisión de secuencias de pulsos de luz IR, con un código que identifica la tecla pulsada”. Esta tecnología permite controlar las funciones que tienen los dispositivos y diseñar un sistema de mando IR que permita programar el funcionamiento de cualquiera de estos electrodomésticos mediante una APP móvil que sea fácil e intuitiva es una solución económica según estudio de presupuestos realizado.

Objetivos

Objetivo general

Implementar un sistema de automatización de electrodomésticos, usando como hardware un emisor de infrarrojo y WeMos D1 R32, enviando instrucciones a este por medio de una App móvil.

Objetivos específicos

Recopilar información sobre los diferentes protocolos de IR que manejan los electrodomésticos, instalación y configuración de servidores web y plataformas de desarrollo para aplicaciones móviles.

Desarrollar un sistema que permita enviar comandos a través de IR para controlar electrodomésticos usando un D1 R32 y una App móvil donde se puedan configurar las instrucciones.

Realizar las pruebas para verificar el correcto funcionamiento del sistema enviando configuraciones de manera local y externa en términos de red.

Justificación

El desarrollo de este proyecto permite dar una solución para automatizar diferentes electrodomésticos mediante mandos IR donde el usuario pueda decidir qué equipo desea configurar, él canal que desea ver en su tv cuando lleguen a su hogar, o la temperatura de la chimenea o calefacción para que esté cálido, así como la fecha y hora. La implementación de este sistema requiere de componentes los cuales son económicos en el mercado y al desarrollar la App móvil con una interfaz intuitiva para la configuración llamara la atención de los usuarios, lo que hace del proyecto una idea de negocio viable, se obtendrán aprendizajes sobre la realización de servidores WEB, desarrollo de aplicaciones móviles, análisis de señales, programación de Arduino y configuraciones para este, el cual un ingeniero en telecomunicaciones podría sacar provecho económico y resaltar en la hoja de vida al desarrollo de este proyecto.

Marco referencial

Estado del arte

Control a distancia mediante mando infrarrojo (IR) es un tutorial en Arduino realizado por (Diosdado, s.f.) Dónde: “montan un circuito sencillo, que nos permita capturar los datos que emite un mando a distancia y con estos datos vamos a encender o apagar diferentes Leds, el montaje que presento aquí se puede complicar todo lo que queráis añadiendo tantos Leds como necesitéis y recordar que al final un LED se puede sustituir por un relé y activar cargas de más potencia, por lo que podéis hacer pequeñas automatizaciones en casa utilizando este sistema”.

Este tutorial usa los elementos mínimos para realizar un sistema de mando por IR como lo es el transmisor y el receptor, el código de programación permite recepcionar la información que transmite un mando para así guardarlo y poderlo transmitir, se puede tomar como referencia ya que me permite tener una visión de cómo llegar a implementar mi idea y darle solución ya que es el principio de mi trabajo de grado.

“Un típico receptor de infrarrojos, es el AX-1838HS, que se consigue por poco más de unos euros. En su hoja de normas encontráis este diagrama, y son relativamente fáciles de encontrar, en dos formas. Independientes y montados en un soporte para utilizar sin complicaciones con nuestro Arduino” (Prometec, s.f.), este tutorial muestra la implementación de un mando de ir usando Arduino y referencia receptores que operan de manera correcta para este tipo de soluciones lo que es de ayuda al momento de elegir los componentes que se usaran en este proyecto.

Hoy en día la tecnología crece rápidamente a tal punto que los dispositivos que se permiten manejar por medio mandos IR y que conocemos desde hace tiempo (control remoto) ahora lo podemos hacer desde aplicativos móviles el tutorial controlar Arduino con un mando a distancia infrarrojo nos dice:

También es posible emplear Arduino para clonar un mando a distancia y, por ejemplo, controlar el encendido de la televisión a través de un smartphone, temporizar el encendido de un equipo de música, o encender el aire acondicionado a través de internet. (Llamas, 2016)

El principal reto de este proyecto de grado es lograr la conexión Dispositivo Android -Arduino

Para que el usuario envié las ordenes o configuraciones que desea automatizar, esto se logra implementando un servidor web y conectando el Arduino a internet, según investigaciones realizadas existen varias maneras de realizarlo el trabajo de grado Desarrollo de un Servidor Domótico nos muestra el “desarrollo e instalación de un servidor domótico de bajo coste en una

red doméstica. Centrado en la creación de un sistema comunicación entre las plataformas Arduino y Raspberry a través de red Ethernet/wifi." (Saiz, 2015), este trabajo de grado a través de una aplicación WEB y la comunicación entre el servidor y el Arduino usa el protocolo basado en UDP, los procesos de lectura que realizan son continuos y son almacenados BBDD, esta implementación es un claro ejemplo de la forma como se puede dar solución al reto de comunicación que se presenta en el proyecto.

Marco teórico

Mit app inventor.

Es un entorno de programación visual e intuitivo que permite a todos, incluso a los niños, crear aplicaciones completamente funcionales para teléfonos inteligentes y tabletas. Aquellos que son nuevos en MIT App Inventor pueden tener una primera aplicación simple en funcionamiento en menos de 30 minutos. Y, lo que, es más, nuestra herramienta basada en bloques facilita la creación de aplicaciones complejas y de alto impacto en mucho menos tiempo que los entornos de programación tradicionales. El proyecto MIT App Inventor busca democratizar el desarrollo de software empoderando a todas las personas, especialmente a los jóvenes, a pasar del consumo de tecnología a la creación de tecnología.

Un pequeño equipo de personal y estudiantes de CSAIL, dirigido por el profesor Hal Abelson, forma el núcleo de un movimiento internacional de inventores. Además de liderar el alcance educativo en torno a MIT App Inventor y realizar investigaciones sobre sus impactos, este equipo central mantiene el entorno de desarrollo de aplicaciones en línea gratuito que sirve a más de 6 millones de usuarios registrados.

Wemos d1 r32.

Es una tarjeta de desarrollo con forma de un Arduino Uno, está basada en el microcontrolador ESP32 WROOM 32 SMD el cual es un potente módulo genérico de Wi-Fi+BT+BLE MCU que se dirige a una amplia variedad de aplicaciones que van desde las redes de sensores de baja potencia hasta las tareas más exigentes, como la codificación de voz, la transmisión de música y la decodificación de MP3.

En el núcleo de este módulo está el chip ESP32-D0WDQ6*. El chip incorporado está diseñado para ser escalable y adaptable. Hay dos núcleos de la CPU que pueden ser controlados individualmente, y la frecuencia del reloj de la CPU es ajustable de 80 a 240 MHz. El usuario también puede apagar la CPU y hacer uso del co-procesador de baja potencia para vigilan

constantemente los periféricos para detectar cambios o cruce de umbrales. El ESP32 integra un rico conjunto de periféricos que van desde sensores táctiles capacitivos, sensores Hall, interfaz de tarjeta SD, Ethernet, SPI de alta velocidad, UART, I²S y I²C.

La integración de Bluetooth, Bluetooth LE y Wi-Fi asegura que una amplia gama de aplicaciones pueda ser dirigidas, y que el módulo es completo: el uso de Wi-Fi permite un gran alcance físico y una conexión directa a Internet a través de un router Wi-Fi, mientras que el uso de Bluetooth permite al usuario conectarse cómodamente al teléfono o a la transmisión balizas de baja energía para su detección. La corriente de reposo del chip ESP32 es inferior a 5 μA, lo que lo hace adecuado para aplicaciones electrónicas de batería y de uso. El módulo soporta una velocidad de datos de hasta 150 Mbps y 20 dBm de potencia de salida en la antena para asegurar el más amplio rango físico. Como tal, el módulo ofrece especificaciones líderes en la industria y el mejor rendimiento para la integración electrónica, rango, consumo de energía y la conectividad.

El sistema operativo elegido para el ESP32 es el freeRTOS con LwIP; el TLS 1.2 con aceleración por hardware está incorporado como bueno.

También se admite la actualización segura (cifrada) por aire (OTA), para que los usuarios puedan actualizar sus productos incluso después de su liberación, con el mínimo costo y esfuerzo.

Firebase.

Firebase de Google es una plataforma en la nube para el desarrollo de aplicaciones web y móvil. Está disponible para distintas plataformas (iOS, Android y web), con lo que es más rápido trabajar en el desarrollo.

Aunque fue creada en 2011 pasó a ser parte de Google en 2014, comenzando como una base de datos en tiempo real. Sin embargo, se añadieron más y más funciones que, en parte, permitieron agrupar los SDK de productos de Google con distintos fines, facilitando su uso.

Su función esencial es hacer más sencilla la creación de tanto aplicaciones webs como móviles y su desarrollo, procurando que el trabajo sea más rápido, pero sin renunciar a la calidad requerida.

Sus herramientas son variadas y de fácil uso, considerando que su agrupación simplifica las tareas de gestión a una misma plataforma. Las finalidades de estas se pueden dividir en cuatro grupos: desarrollo, crecimiento, monetización y análisis. Es especialmente interesante para que los desarrolladores no necesiten dedicarle tanto tiempo al backend, tanto en cuestiones de desarrollo como de mantenimiento.

Funciones de firebase.

Firebase dispone de diferentes funcionalidades, que se pueden dividir básicamente en 3 grupos: Desarrollo (Develop), Crecimiento (Grow) y Monetización (Earn), a los que hay que sumar la Analítica (Analytics).

Desarrollo.

El primer grupo de funciones es conocido como Desarrollo o Develop en Firebase. Como su nombre indica, incluye los servicios necesarios para el desarrollo de un proyecto de aplicación móvil o web. Estos contribuyen a que el proceso sea más rápido, puesto que se dejan determinadas actividades a mano de Firebase, mientras que otras permiten optimizar diversos aspectos para conseguir la calidad deseada**.**

Realtime database.

Una de las herramientas más destacadas y esenciales de Firebase son las bases de datos en tiempo real. Estas se alojan en la nube, son No SQL y almacenan los datos como JSON. Permiten alojar y disponer de los datos e información de la aplicación en tiempo real, manteniéndolos actualizados, aunque el usuario no realice ninguna acción.

Firebase envía automáticamente eventos a las aplicaciones cuando los datos cambian, almacenando los datos nuevos en el disco. Aunque no hubiera conexión por parte de un usuario, sus datos estarían disponibles para el resto y los cambios realizados se sincronizarían una vez restablecida la conexión.

Autenticación de usuarios.

La identificación de los usuarios de una app es necesaria en la mayoría de los casos si estos quieren acceder a todas sus características.

Firebase ofrece un sistema de autenticación que permite tanto el **registro** propiamente dicho (mediante email y contraseña) como el acceso utilizando perfiles de otras plataformas externas (por ejemplo, de Facebook, Google o Twitter), una alternativa muy cómoda para usuarios reacios a completar el proceso.

Así, este tipo de tareas se ven simplificadas, considerando también que desde aquí se gestionan los accesos y se consigue una mayor seguridad y protección de los datos. Se debe mencionar que Firebase puede guardar en la nube los datos de inicio de sesión con total seguridad, evitando que una persona tenga que identificarse cada vez que abra la aplicación.

Almacenamiento en la nube.

Firebase cuenta con un sistema de almacenamiento, donde los desarrolladores pueden guardar los ficheros de sus aplicaciones (y vinculándolos con referencias a un árbol de ficheros para mejorar el rendimiento de la app) y sincronizarlos. Al igual que la mayoría de las herramientas de Firebase, es personalizable mediante determinadas reglas.

Este almacenamiento es de gran ayuda para **tratar archivos de los usuarios** (por ejemplo, fotografías que hayan subido), que se pueden servir de forma más rápida y fácil. También hace la descarga de referencias a ficheros más segura.

Crash Reporting.

Para mantener y mejorar la calidad de la app, hay que prestar especial atención a los fallos, por lo que los seguimientos de errores (y también del rendimiento general de la app) son clave para poder actuar y solucionarlos.

Por ello, Firebase ofrece Crash Reporting, que detecta y ayuda a solucionar los problemas de la app, consiguiendo un informe de errores muy detallado (con datos como el dispositivo o la situación en la que se da la excepción) y organizado, puesto que los agrupa por similitud y los clasifica por gravedad.

Test lab.

El Laboratorio de pruebas permite **testear la app** en dispositivos Android virtuales basados en los parámetros que configuremos. De esta forma, es mucho más sencillo detectar posibles errores antes de lanzar la aplicación.

Remote config.

La configuración remota sirve para modificar ciertas funciones, aspectos o incluso la apariencia de la aplicación sin que sea necesario publicar una actualización de la misma. De esta forma, no se requiere ningún tipo de acción por parte del usuario y se trata de cambios mucho más dinámicos.

Existen diversos parámetros que permiten personalizar al detalle estos cambios, considerando factores como la ubicación o idioma del usuario, su dispositivo de acceso, etc.

Modulo receptor ir.

Para la recepción de los mandos IR en el proyecto se va a requerir de un módulo y según (DE SENSORES, 2020): Los detectores IR son pequeños microchips con una fotocélula que están sintonizados para detectar la luz infrarroja. Casi siempre se usan para la detección de control remoto: cada televisor y reproductor de DVD tiene uno de estos en el frente para detectar la señal

IR. Dentro del control remoto hay un LED IR, que emite pulsos IR para indicarle al televisor que encienda, apague o cambie los canales. La luz IR no es visible para el ojo humano, lo que significa que se necesita un poco más de trabajo para probar un experimento. Hay algunas diferencias entre estos y las fotoceldas de CdS:

Los detectores IR están especialmente filtrados para luz infrarroja, no son buenos para detectar luz visible. Por otro lado, las fotocélulas son buenas para detectar luz visible amarilla / verde, y no son buenas para luz IR

Los detectores de IR tienen un demodulador en su interior que busca una señal IR modulada a 38 KHz. Una luz que simplemente parpadea no será detectada por un LED IR, debe parpadear en PWM a 38KHz para la detección. En cambio, las fotocélulas no tienen ningún tipo de demodulador y pueden detectar cualquier frecuencia (incluida CC) dentro de la velocidad de respuesta de la fotocélula (que es de aproximadamente 1 kHz).

Los detectores IR son de salida digital: detectan una señal IR de 38KHz y una salida baja (0V) o no detectan ninguna señal y con salida alta (5V). Las fotocélulas actúan como resistencias, la resistencia cambia según la cantidad de luz a la que están expuestas.

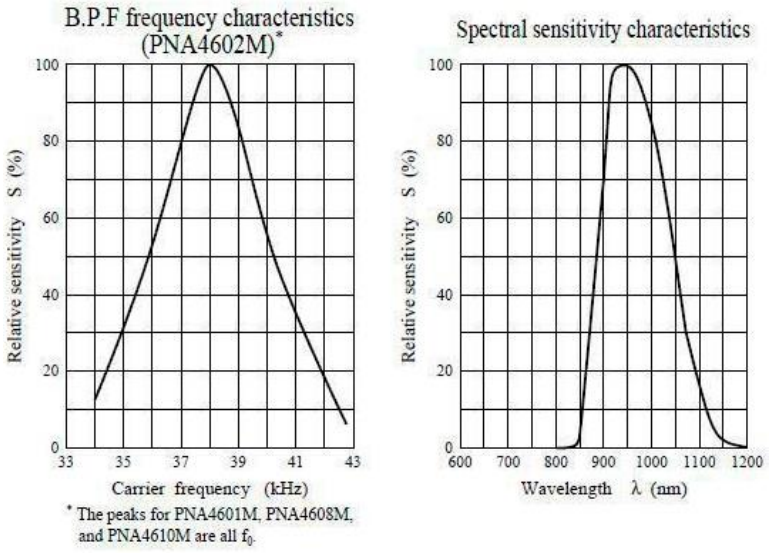

Figura 1. Características de Frecuencia y sensibilidad espectral. De sensores (2020)

Como se puede ver en estos gráficos de la hoja de datos, la frecuencia máxima detectada es de 38 KHz y el valor máximo del color del LED es de 940 nm. Se puede usar desde aproximadamente 35 KHz a 41 KHz, pero la sensibilidad disminuirá de tal forma que no se detectará muy bien desde lejos. Del mismo modo, se puede usar un LED de 850 a 1100 nm, pero no funcionarán tan bien como 900 a 1000 nm, ¡así que asegúrese de obtener los LED correspondientes! Consulta la hoja de

datos de su LED IR para verificar la longitud de onda. Trate de obtener uno de 940 nm: recuerda que 940 nm no es luz visible.

Modulo emisor ir.

La emisión de los mandos IR será por medio de un módulo y según (Electronica CIDEA , 2020):Los diodos infrarrojos (IR) funcionan convirtiendo la corriente eléctrica en luz infrarroja; mientras que los detectores infrarrojos hacen lo opuesto al detectar luz infrarroja y convertirla en una corriente eléctrica. La corriente generada por un detector infrarrojo es una señal que indica que existe ese tipo de luz. El infrarrojo es una longitud de onda de luz que está más allá del rango de la visión humana. Esto hace al infrarrojo una herramienta excelente para aplicaciones donde se requiere la luz, pero donde la luz visible podría ser una distracción o de otra forma no deseada. El uso de diodos infrarrojos emisores de luz, o LEDs, hace posibles a los sistemas de control remoto en varios proyectos.

El IR383 está espectral mente emparejado con foto transistor, foto diodo y el módulo receptor de infrarrojos, moldeado en un paquete de plástico azul transparente.

- Alta fiabilidad
- Alta intensidad radiante
- Longitud de onda máxima $\lambda p = 940nm$
- Espaciamiento de plomo de 2.54 mm
- Sin plomo
- Baja tensión directa

Aplicaciones: Sistema de transmisión de aire libre, unidades de control remoto infrarrojo con alto requerimiento de potencia, detector de humo y sistema aplicado infrarrojo

Marco legal

Normas nacionales.

- Artículo 3 de la ley 1480 del 2011, (Congreso de la Republica, 2019)

  "Derechos:

  - Derecho a recibir productos de calidad: Recibir el producto de conformidad con las condiciones que establece la garantía legal, las que se ofrezcan y las habituales del mercado.

- Derecho a la seguridad e indemnidad: Derecho a que los productos no causen daño en condiciones normales de uso y a la protección contra las consecuencias nocivas para la salud, la vida o la integridad de los consumidores.
- Derecho a recibir información: Obtener información completa, veraz, transparente, oportuna, verificable, comprensible, precisa e idónea respecto de los productos que se ofrezcan o se pongan en circulación, así como sobre los riesgos que puedan derivarse de su consumo o utilización, los mecanismos de protección de sus derechos y las formas de ejercerlos.
- Derecho a recibir protección contra la publicidad engañosa.
- Derecho a la reclamación: Reclamar directamente ante el productor, proveedor o prestador y obtener reparación integral, oportuna y adecuada de todos los daños sufridos, así como tener acceso a las autoridades judiciales o administrativas para el mismo propósito, en los términos de la presente ley. Las reclamaciones podrán efectuarse personalmente o mediante representante o apoderado.
- Protección contractual: Ser protegido de las cláusulas abusivas en los contratos de adhesión, en los términos de la presente ley.
- Derecho de elección: Elegir libremente los bienes y servicios que requieran los consumidores.
- Derecho a la participación: Organizarse y asociarse para proteger sus derechos e intereses, elegir a sus representantes, participar y ser oídos por quienes cumplan funciones públicas en el estudio de las decisiones legales y administrativas que les conciernen, así como a obtener respuesta a sus peticiones.
- Derecho de representación: Los consumidores tienen derecho a hacerse representar, para la solución de las reclamaciones sobre consumo de bienes y servicios, y las contravenciones a la presente ley, por sus organizaciones, o los voceros autorizados por ellas.
- Derecho a informar: Los consumidores, sus organizaciones y las autoridades tendrán acceso a los medios masivos de comunicación, para informar, divulgar y educar sobre el ejercicio de los derechos de los consumidores.

- Derecho a la educación: Los ciudadanos tienen derecho a recibir educación sobre los derechos de los consumidores, formas de hacer efectivos sus derechos y demás materias relacionadas.
- Derecho a la igualdad: Ser tratados equitativamente y de manera no discriminatoria. (pág. 1).”

Metodología

El desarrollo de este trabajo de grado estaría enfocado al método de investigación cuantitativo ya que se pretende exponer hipótesis sobre la ejecución del sistema de mandos IR para controlar los electrodomésticos mediante una aplicación móvil.

Sobre la investigación cuantitativa "Utiliza la recolección de datos para probar hipótesis con base en la medición numérica y el análisis estadístico, con el fin establecer pautas de comportamiento y probar teorías." (Sampieri, 2014) Se empleara el análisis cuantitativo cuando se implemente el sistema y se tomen mediciones de las señales que transmite el sistema de automatización de mandos IR, así como las señales que recibe el dispositivo y así saber el comportamiento técnico como las frecuencias, distancias, de diferentes dispositivos que son operados por mando IR y poder realizar comparativos de estas señales (TX,RX), así como la recopilación de información sobre las aplicaciones que permiten desarrollar software para Android el cual será la interfaz con el usuario para programar y configurar el dispositivo de control IR, lo que llevara a la construcción de una solución que refute la hipo tesis planteada

Presupuesto

Tabla 1.

Presupuesto necesario para el proyecto

Tipo	Referencia	Cantidad	Precio Unitario	Total
Recursos Necesarios	WeMos D1 R32	1	$ 30.500	$ 30.500
	Ds3231	1	$ 10.000	$ 10.000
	Receptor Ir	1	$ 1.000	$ 1.000
	Transmisor Ir	1	$ 2.000	$ 2.000
	Protoboard	1	$ 13.000	$ 13.000
	Jumper De Conexión	20	$ 500	$ 10.000
	Fuente de Alimentación	1	$5.000	$5.000
	Laptop	1	$0.000	$0.000
Recursos Disponibles	Servicios De Ingeniería	1	$300.000	$300.000
	Uso De Equipos (Tv, Aires Acondicionados, Etc.)	1	$0.000	$0.000
Total, Recurso Necesarios			$0.000	$ 371.500

Nota. Autoría propia

Cronograma

Tabla 2.

Cronograma del Anteproyecto

Fecha	Actividades	1	2	3	4	5	6	7
marzo 16 - marzo 26	Realizar Investigación acerca de los formatos sobre tecnología IR							
marzo 27 - abril 6	Realizar Investigación sobre servidores WEB							
abril 7 - abril 17	Realizar investigación sobre las aplicaciones que permiten desarrollar APP móviles en sistemas operativos Android							
abril 18 - abril 28	recopilación de datos de investigaciones y conclusiones							
abril 29 - mayo 9	Desarrollo del código para recepción de instrucciones en Arduino							
Fecha	Actividades	8	9	10	11	12	13	14
Mayo 10 – mayo 16	Integración del módulo wifi ESP8266 al hardware y desarrollar el código para tener el Arduino conectado a internet							
Mayo17 - junio 7	Desarrollo de la app móvil y pruebas de funcionamiento							
Junio 8 – junio 14	Configuración de servidor WEB							
Junio 14-22 junio	Pruebas de conexión APP móvil con el servidor WEB							

Nota. Autoría propia

Protocolos de ir, configuración del servidor y plataforma de desarrollo de aplicaciones móviles firebase

Los mandos remotos desde su inicio tuvieron varios avances tecnológicos desde mandos ultrasonidos hasta los que hoy en día se conocen en el mercado con tecnología infrarroja, en la cual se basa este proyecto de grado, los diferentes fabricantes de electrodomésticos han creado sus propios protocolos IR, los más conocidos son NEC, Philips RC5, Philips RC6 y SIRC.

Según (ALTIUM , 2017) " El protocolo NEC utiliza codificación de distancia de pulso de los bits del mensaje. Cada ráfaga de pulsos tiene una longitud de 562,5 µs, a una frecuencia portadora de 38 kHz (26,3 µs). Los bits lógicos se transmiten de la siguiente manera:

- '0' lógico: una ráfaga de pulsos de 562,5 µs seguida de un espacio de 562,5 µs, con un tiempo de transmisión total de 1,125 ms
- '1' lógico: una ráfaga de pulsos de 562,5 µs seguida de un espacio de 1,6875 ms, con un tiempo de transmisión total de 2,25 ms

Cuando se presiona una tecla en el control remoto, el mensaje transmitido consiste en lo siguiente, en orden:

- Una ráfaga de pulso inicial de 9 ms (16 veces la longitud de ráfaga de pulso utilizada para un bit de datos lógicos)
- un espacio de 4.5ms
- la dirección de 8 bits para el dispositivo receptor
- el inverso lógico de 8 bits de la dirección
- el comando de 8 bits
- el inverso lógico de 8 bits del comando
- una ráfaga de pulsos final de 562,5 µs para indicar el final de la transmisión del mensaje.

Los cuatro bytes de bits de datos se envían primero el bit menos significativo. La Figura 2 ilustra el formato de una trama de transmisión NEC IR, para una dirección de 00h(00000000b) y un comando de ADh (10110101b)".

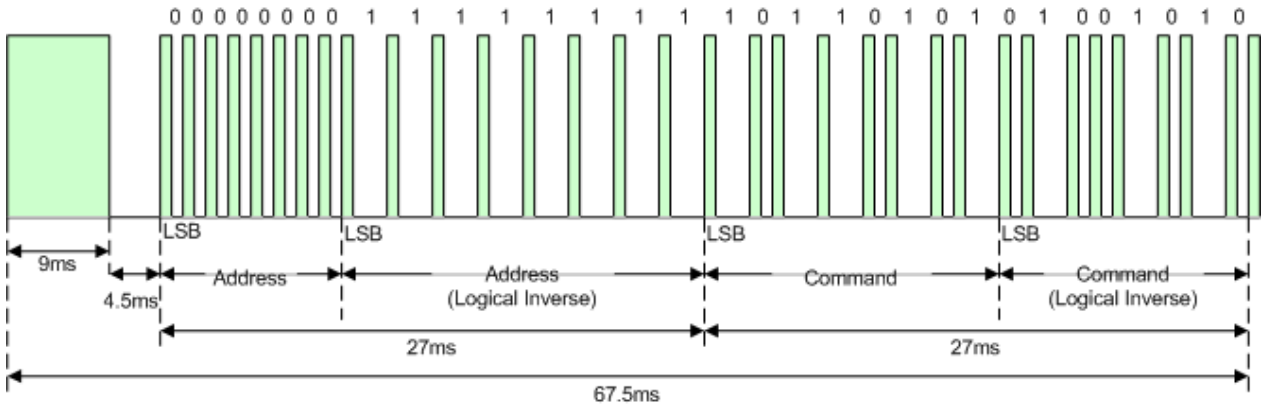

Figura 2. Ejemplo de marco de mensaje que utiliza el protocolo de transmisión NEC IR. Altium (2017).

Protocolo rc5

El protocolo de transmisión RC5 fue desarrollado por la fábrica Philips, sobre este protocolo (ALTIUM , 2017) nos dice que:

"RC5 IR utiliza la codificación Manchester de los bits del mensaje. Cada ráfaga de pulso (marca - transmisor RC ENCENDIDO) tiene una longitud de 889us, a una frecuencia portadora de 36kHz (27.7us). Los bits lógicos se transmiten de la siguiente manera:

- '0' lógico: una ráfaga de pulsos 889us seguida de un espacio 889us, con un tiempo de transmisión total de 1.778ms
- '1' lógico: un espacio 889us seguido de una ráfaga de pulsos 889us, con un tiempo de transmisión total de 1.778ms

Cuando se presiona una tecla en el control remoto, la trama de mensaje transmitida consta de los siguientes 14 bits, en orden:

- dos bits de inicio (S1 y S2), ambos lógicos '1'.
- a Bit de alternancia (T). Este bit se invierte cada vez que se suelta una tecla y se vuelve a presionar.
- la dirección de 5 bits para el dispositivo receptor
- el comando de 6 bits.

Los bits de dirección y comando se envían primero el bit más significativo. La Figura 3 ilustra el formato de una trama de transmisión IR de Philips RC5, para una dirección de `05h`(`00101b`) y un comando de `35h`(`110101b`)".

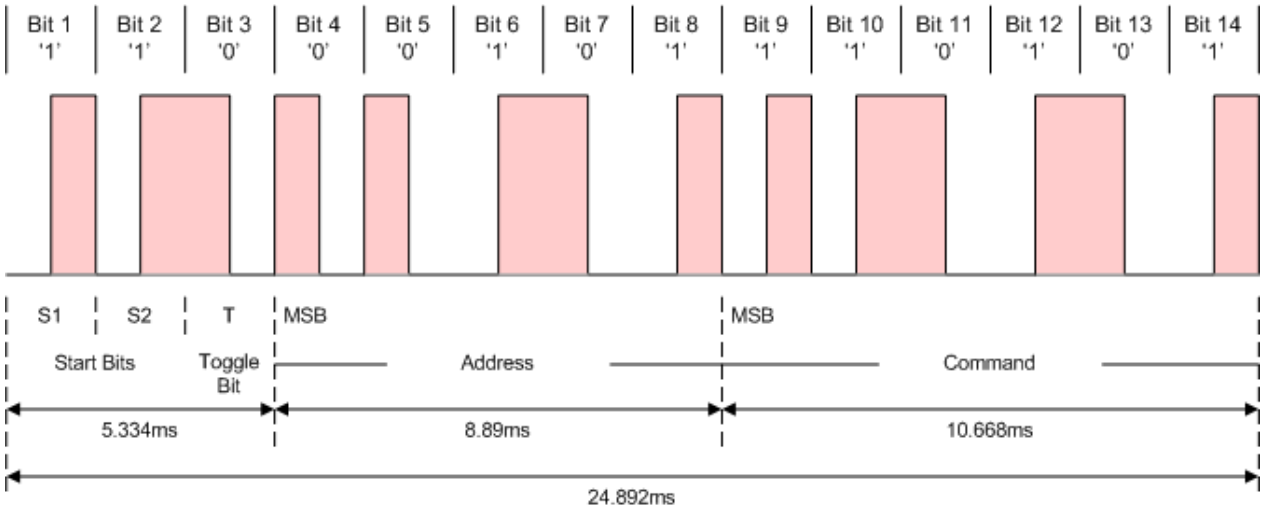

Figura 3. Ejemplo de marco de mensaje con el protocolo de transmisión de infrarrojos Philips RC5. Altium (2017).

Protocolo rc6

Este protocolo reemplaza al protocolo RC-5. Al igual que RC-5, Philips también definió el nuevo protocolo RC-6. Es un protocolo muy versátil y bien definido

características del protocolo rc6.

- Diferentes modos de funcionamiento, según el uso previsto.
- Modos dedicados de Philips y modos OEM.
- Longitud de comando variable, dependiendo del modo de operación.
- Codificación bifásica (también conocida como codificación Manchester).
- Frecuencia portadora de 36 kHz.
- Fabricante Philip

Las señales RC-6 se modulan en una portadora de infrarrojos de 36 kHz. El ciclo de trabajo de este portador debe estar entre el 25% y el 50%.

Los datos se modulan mediante la codificación Manchester. Esto significa que cada bit (o símbolo) tendrá una marca y un espacio en la señal de salida. Si el símbolo es un "1", la primera mitad del tiempo de bit es una marca y la segunda mitad es un espacio. Si el símbolo es un "0", la primera mitad del tiempo de bit es un espacio y la segunda mitad es una marca.

La unidad de tiempo principal es 1t, que es 16 veces el período de la portadora como se muestra en la ecuación 1

$$\frac{1}{36k \times 16} = 444\mu s \qquad (1)$$

Protocolo ir de transmisión Sony sirc

Existen tres versiones para el protocolo: 12-bits, 15-bits y 20-bits. Se puede asumir que las versiones de 15-bits y 20-bits difieren en el número de bits transmitidos en la secuencia de comando.

Características protocolo sirc.

• Para el Protocolo, existen versiones de: 12-bits, 15-bits y 20-bits (12-
bits será descrita en elpresente documento)

• Longitud de: 5-bits de dirección y 7-bits de comando (Protocolo de 12-bits)

• Modulación por Ancho de Pulsos (PWM)

• Frecuencia de portadora de 40 Khz.

• Tiempo de bit de 1.2ms ó 0.6ms

El Protocolo SIRC usa una codificación por ancho de pulsos para los bits. El pulso que representa el estado lógico "1" es 1.2ms de presencia de la portadora de 40kHz, mientras que el ancho para el estado lógico "0" es 0.6ms de largo. Todas las presencias de portadora son separadas por un intervalo sin presencia de portadora de duración de 0.6ms. El ciclo de trabajo recomendado, de la portadora es 1/4 ó1/3

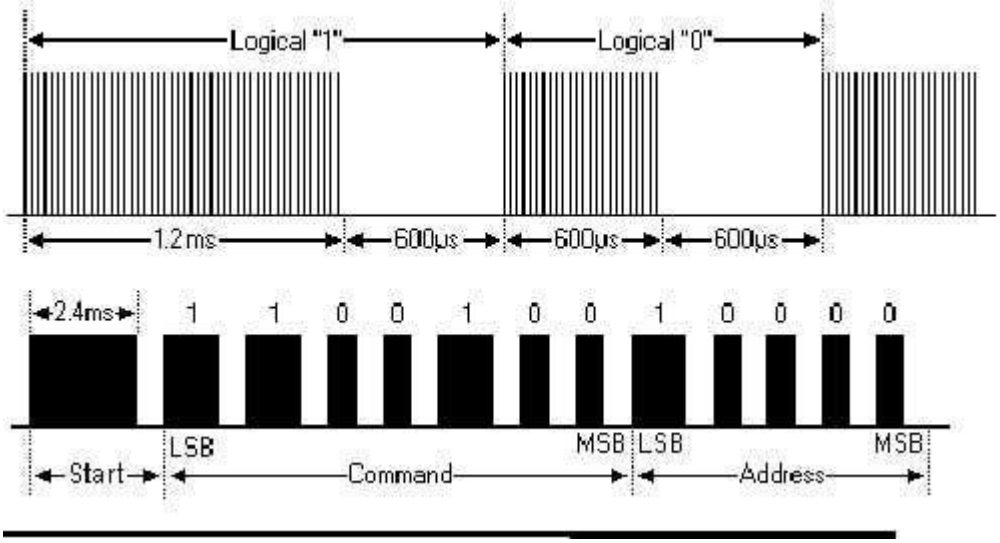

Figura 4. Descripción grafica protocolo SIRC. Scribd (2012).

Servidor firebase

En el desarrollo del estado del arte se encontraron proyectos que realizaban la instalación y configuración de sus propios servidor WEB lo que incurre en tener una maquina física, un domino para este, lo que eleva el presupuesto del proyecto, existen otros servidores web que se ofrecen en el mercado que cobran una cuota para permitir el uso de este, por esta razón se utilizó la plataforma Firebase de Google que es totalmente gratis, cuenta con grandes características y lo más importante es totalmente compatible con el entorno de desarrollo para la creación de aplicaciones Android (APP INVENTOR) lo que permitió la comunicación y que el usuario pueda enviar las instrucciones, los pulsos del control y toda las opciones que se disponen en la APP de manera más ágil cumpliendo con lo que se necesita para la realización del proyecto.

Inicio firebase

Para iniciar la creación de este sistema tuve que crear una base de datos donde se enviarán la información en tiempo real que se registren en la aplicación móvil y a su vez el módulo D1 R32 recibirá estos datos para ejecutar la instrucción.

Para ingresar al servidor de Firebase se debe tener una cuenta de correo de Google, después de realizar el ingreso al servidor de Firebase se añadió un proyecto que tiene como nombre AUTOMATIZACION IR para empezar a trabajar la base de datos

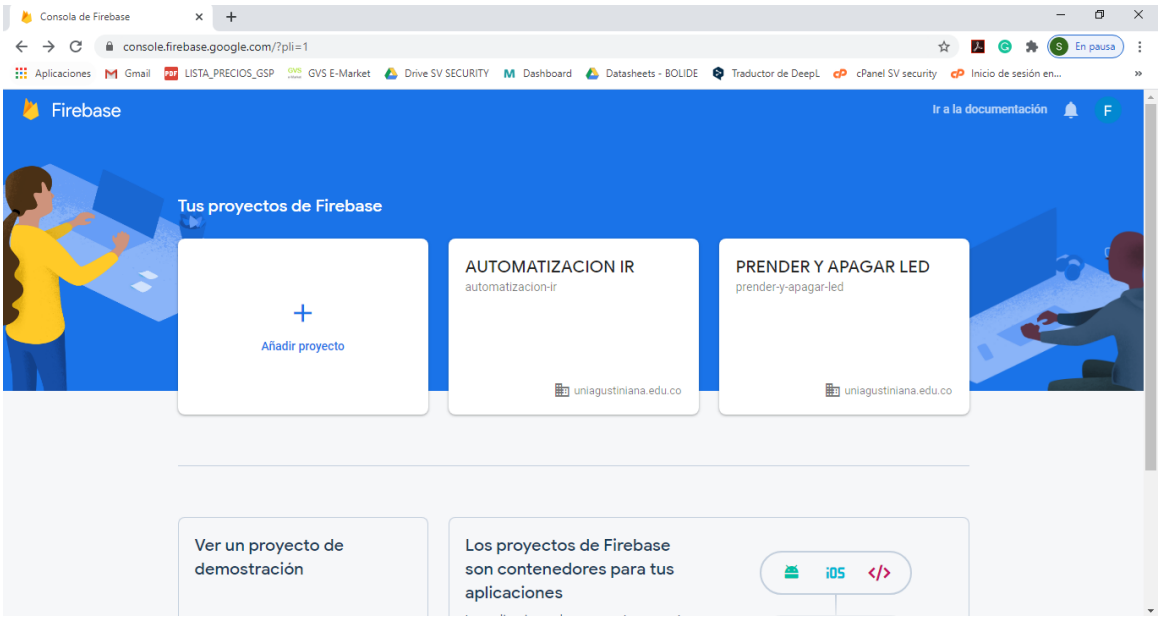

Figura 5. Consola Firebase. Autoría propia

Cuando se ingresa al proyecto vamos a encontrar las características que nos ofrece Firebase, me enfocare en la característica de base de datos en tiempo real, la cual nos va a permitir comunicarnos con la aplicación que realizaremos en APP Inventor y a su vez con el D1 R32

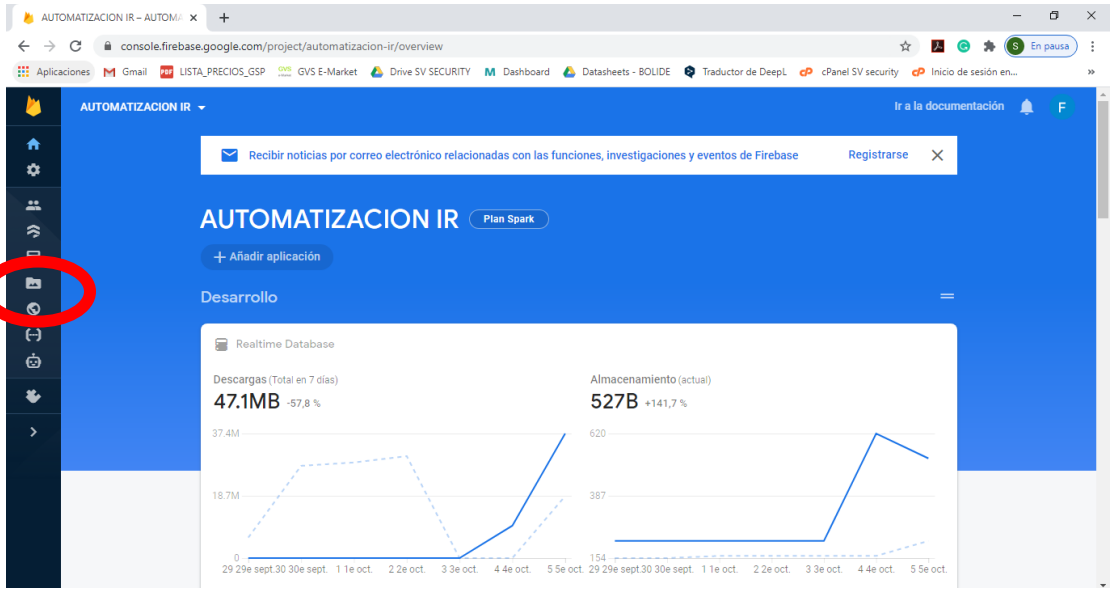

Figura 6. Espacio de trabajo proyecto Firebase. Autoría propia

En la base de datos en tiempo real vamos a visualizar cualquier cambio que se envié desde la APP móvil, esto es posible debido a que en el D1 R32 y en el APP Inventor se debe ingresar la URL del proyecto y la clave secreta de la base de datos, con estos dos datos vamos a poder conectarnos sin necesidad de tener una IP privada o realizar alguna apertura de puertos en la red de los equipos.

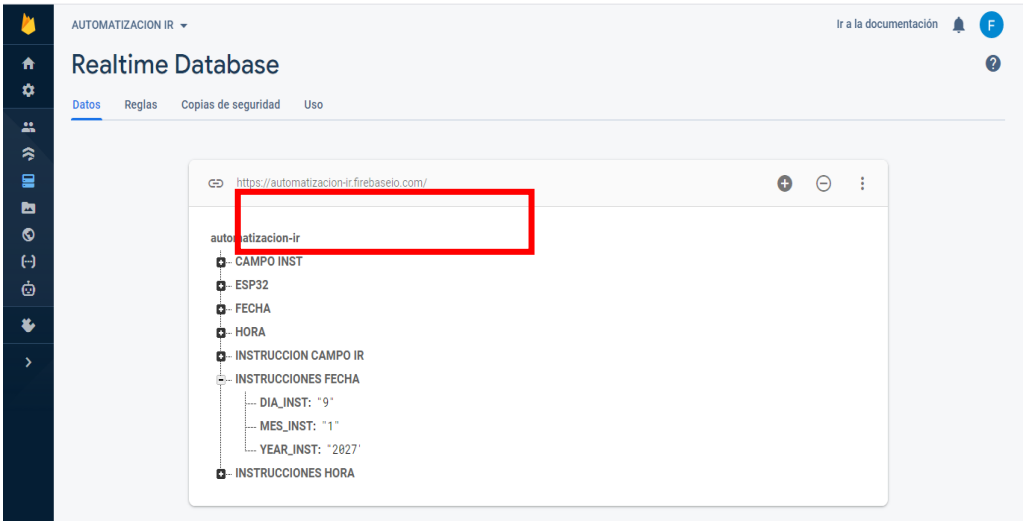

Figura 7. URL de la Base de Datos. Autoría propia

La URL la vamos a visualizar cuando ingresamos a la característica de Realtime Database (figura 7) se debe guardar esta porque será el primer requisito para poder tener comunicación entre los dispositivos.

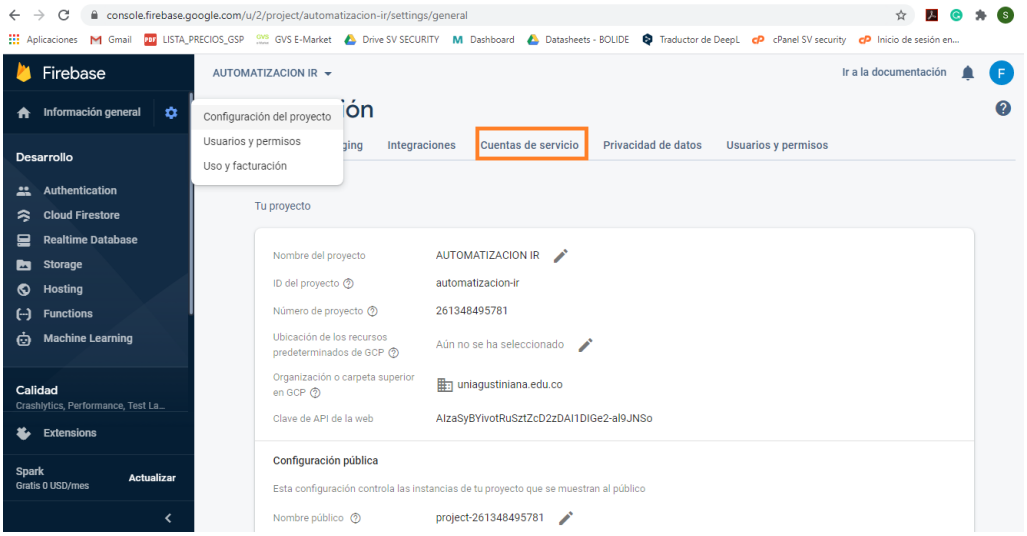

Figura 8. Clave Secreta Firebase. Autoría propia

La clave secreta de Firebase la vamos a buscar siguiendo la siguiente ruta:

Configuraciones del proyecto→ cuentas de servicio

Encontraremos el menú que se visualiza en la figura 10

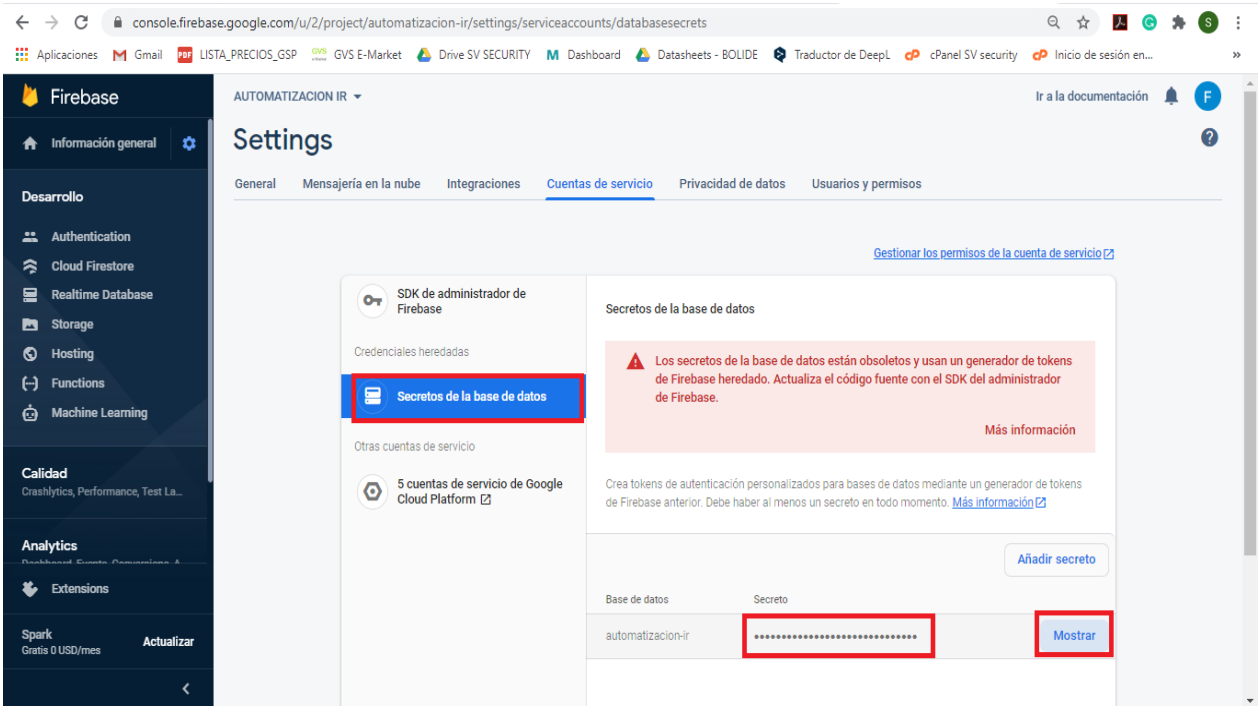

Figura 9. Menú cuenta de servicios. Autoría propia

Cuando ya se ingresa al menú de cuentas de servicios vamos a secretos de la base de datos allí vamos a encontrar la contraseña secreta dando clic en la opción **mostrar,** guardamos esta ya que es el segundo requisito para la comunicación de la app móvil y el D1 R32.

Desarrollo del sistema para enviar comandos ir

Para el desarrollo de la aplicación móvil en la plataforma Android se usó el entorno de programación APP Inventor, para ingresar se debe tener una cuenta de Google.

Se creo un proyecto el cual se nombró AutoIR (figura 10) donde se desarrolla la interfaz gráfica de la aplicación y los menús del sistema se configurarán en el modo desarrollador

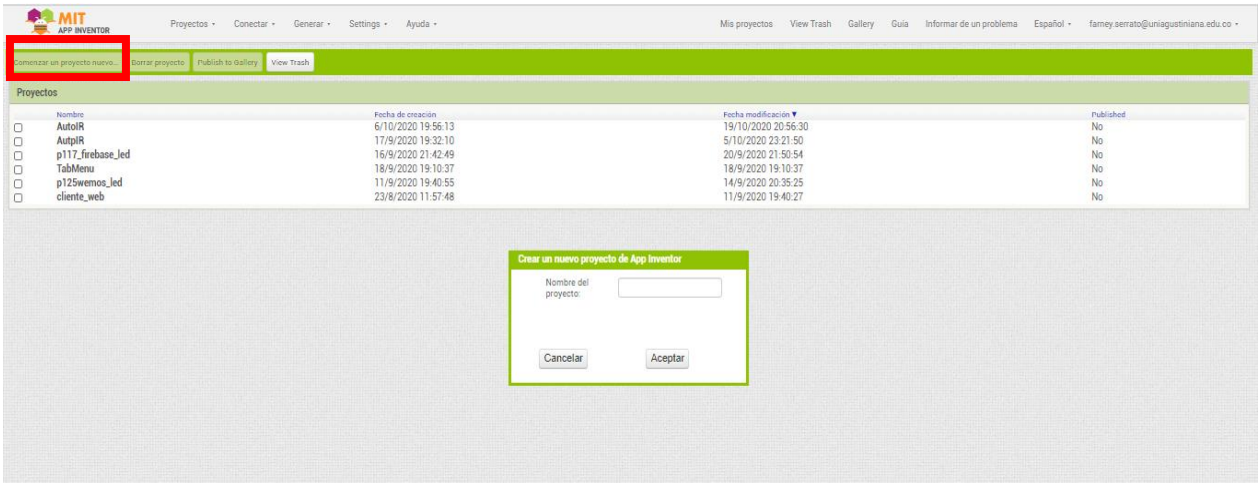

Figura 10. Creación del Proyecto. Autoría propia

En el costado izquierdo del entorno se encuentran las diferentes opciones (imagen 11) que se usaron para empezar a crear los seis screen que componen la APP

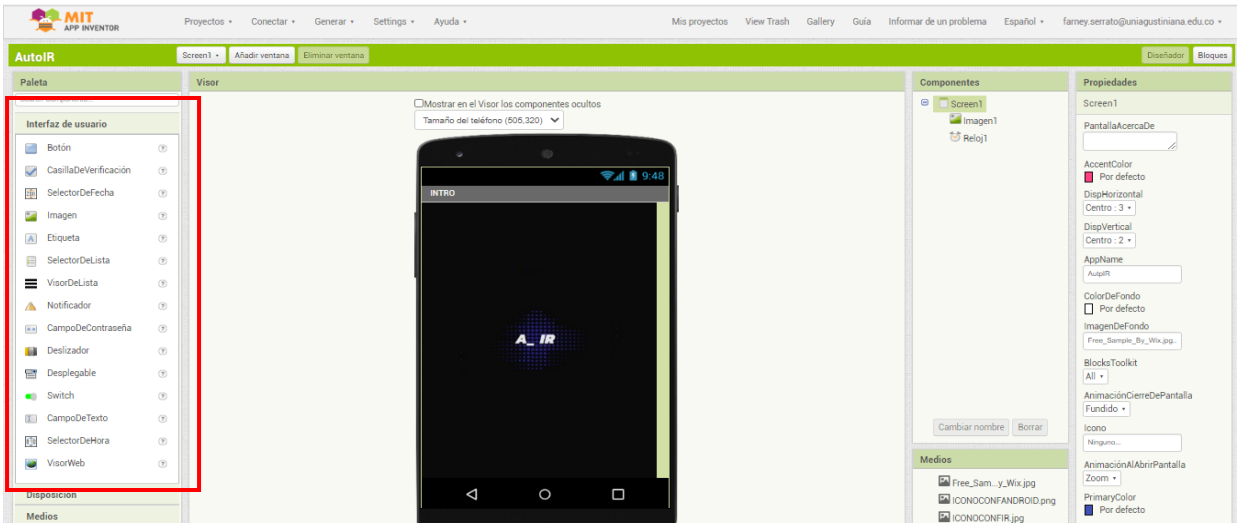

Figura 11. Screen de Inicio. Autoría propia

El screen de inicio está compuesto por el logo de la aplicación y tiene una duración de tres segundos, pasado ese tiempo pasara el screen de principal o donde se encuentran las opciones para iniciar configurar lo que desee el usuario.

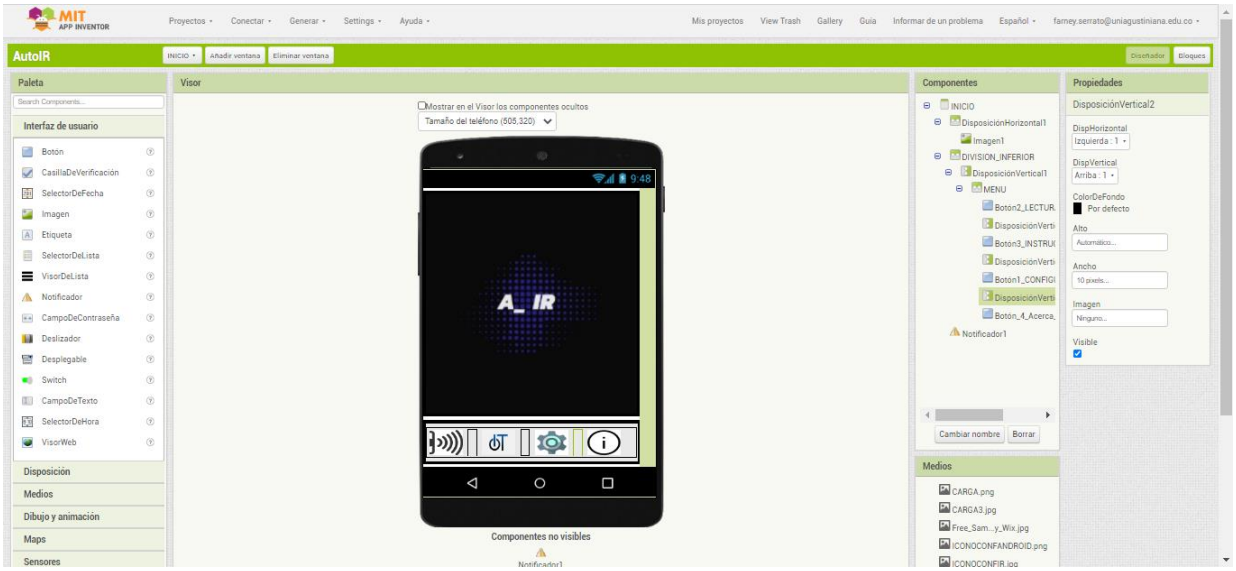

Figura 12. Screen Principal. Autoría propia

El screen principal cuenta con cuatro botones, lectura IR, IOT, Configuraciones del sistema, Acerca de, cuando el usuario obture alguno de estos botones lo dirigirá para que proceda a realizar la acción.

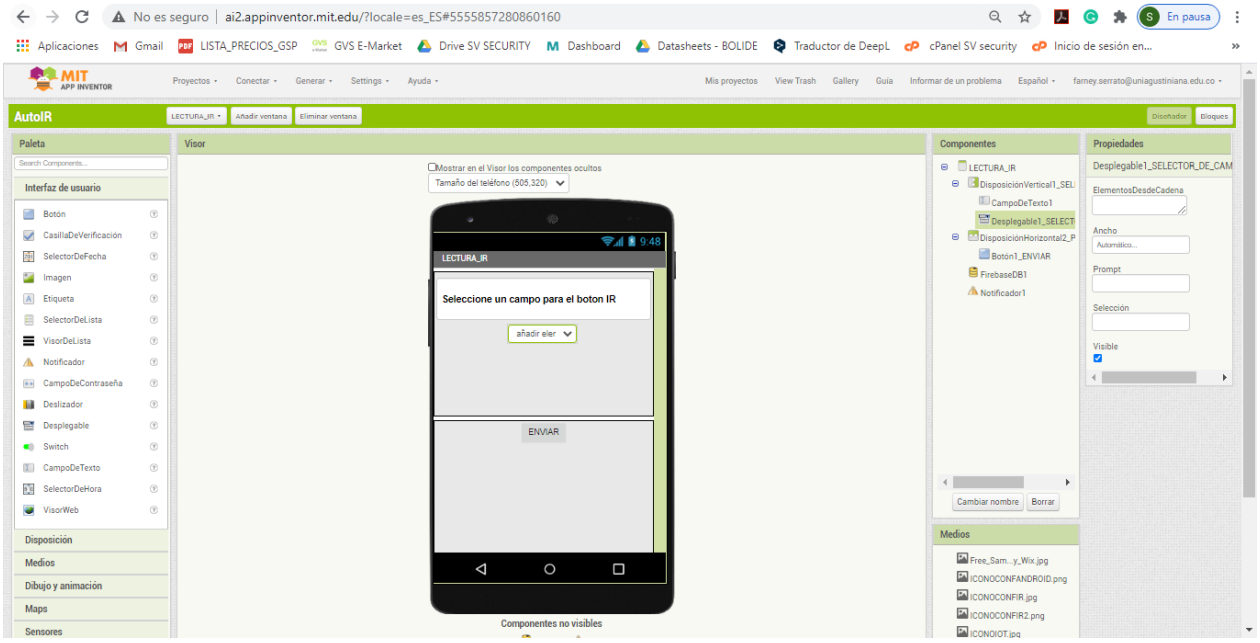

Figura 13. Screen Lectura IR. Autoría propia

En el screen de lectura será el campo en la memoria del D1 R32 donde se almacenará la recepción de pulsos que se envían desde el control remoto de los dispositivos a controlar, el usuario tendrá nueve campos para elegir

Figura 14. Selector del campo IR. Autoría propia

Una vez el usuario seleccione el campo deseado y presione el botón enviar automáticamente se abrirá el Screen de carga para la lectura de los pulsos IR (imagen 15) se debe presionar el botón deseado en el control remoto y apuntar al sensor receptor IR una vez se realice esta acción para confirmar se debe obturar el botón enviar si la lectura del control es correcta se visualizará un mensaje informando de esta acción y de no ser correcta la lectura se mostrara un mensaje de alerta donde el usuario deberá volver a realizar los pasos anteriores de nuevo.

Figura 15. Screen carga de lectura IR. Autoría propia

Figura 16. Screen instrucciones. Autoría propia

En el screen Instrucciones el usuario configurara la fecha, hora y el campo IR que desee que se accione para controlar un dispositivo (figura 16).

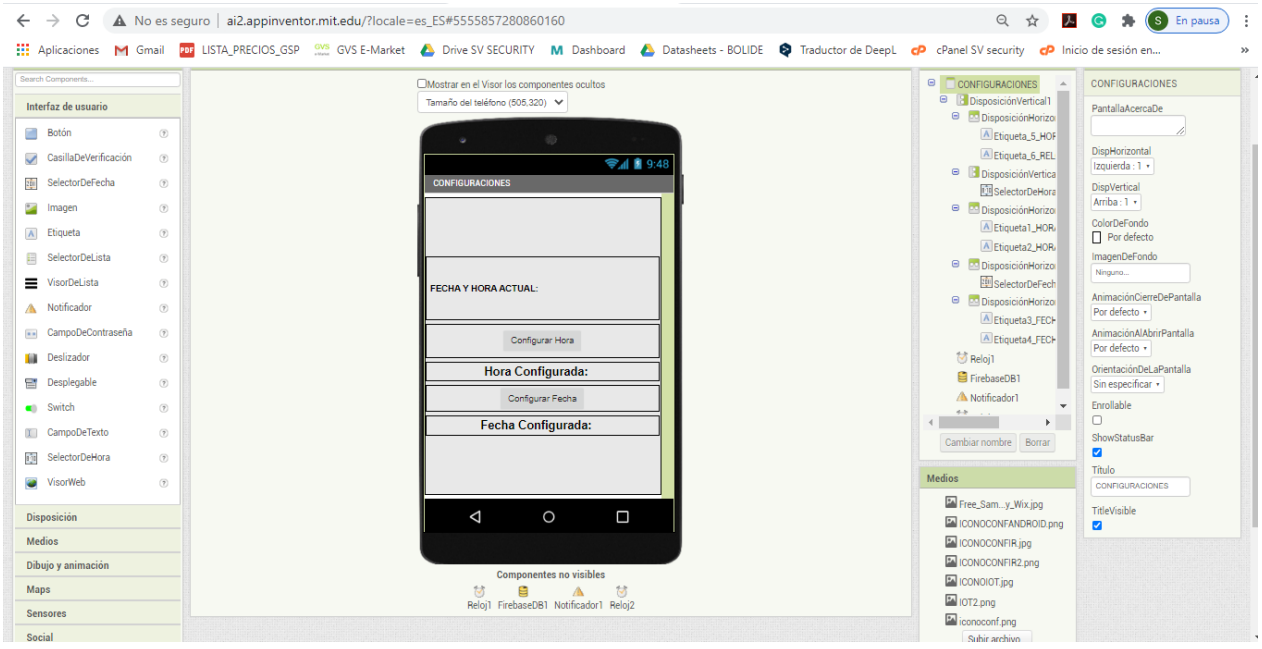

Figura 17. Screen configuraciones. Autoría propia

El screen de configuraciones permite realizar configuraciones de la hora y fecha del sistema el cual es controlado por el módulo DS3231.

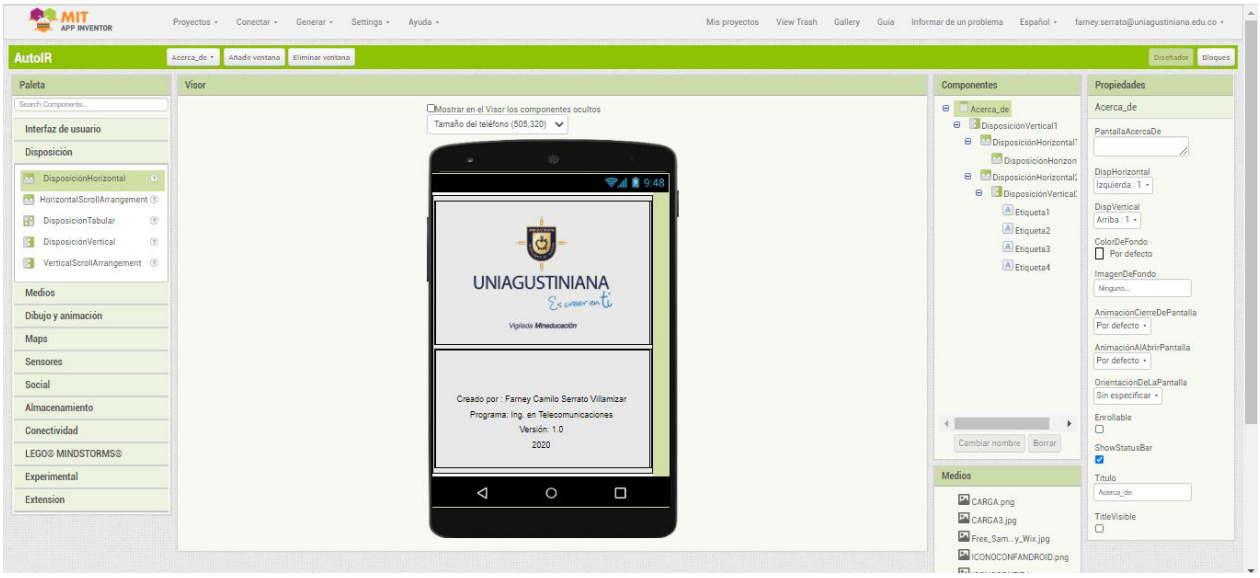

Figura 18. Screen Acerca de. Autoría propia

En el screen acerca de se visualiza la información de la versión de la aplicación y el nombre del desarrollador de la aplicación (figura 18).

En cada screen de los cuatro botones de configuraciones que se desarrolló se tuvo que colocar la clave secreta y la URL que se mencionó en la figura 8 y figura 10 para que tenga conexión con el servidor de Firebase

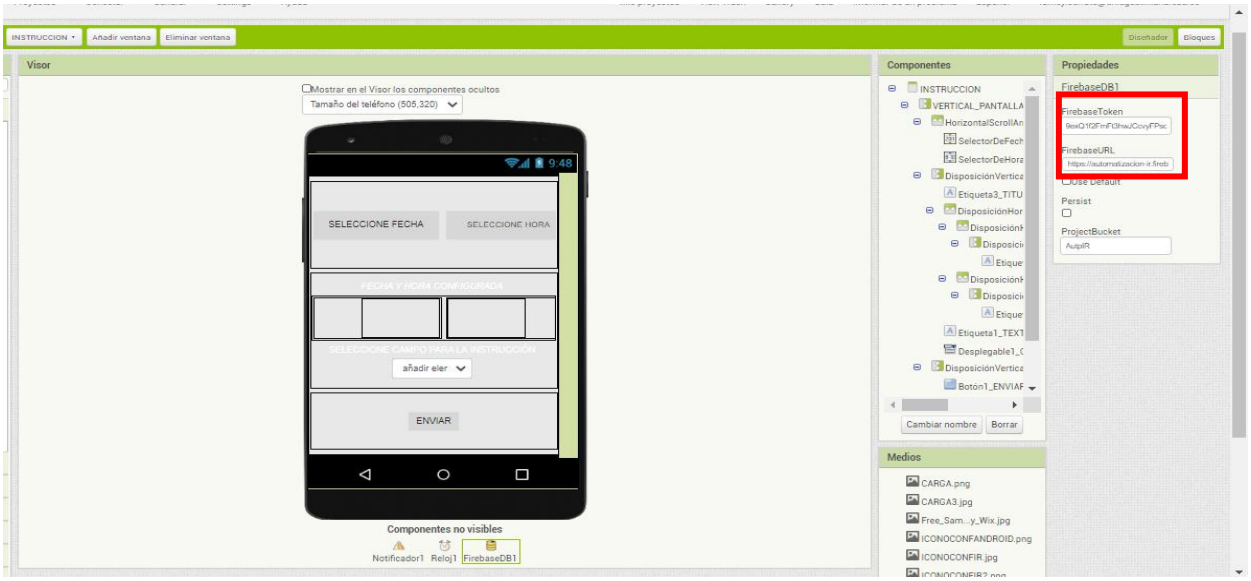

Figura 19. Ingreso clave secreta y URL Firebase. Autoría propia

Después de crear los menús en el modo “Diseñador” se debe programar cada una de las funciones de estas en el modo “Bloques” estas configuraciones tienen una lógica muy intuitiva y fácil de realizar

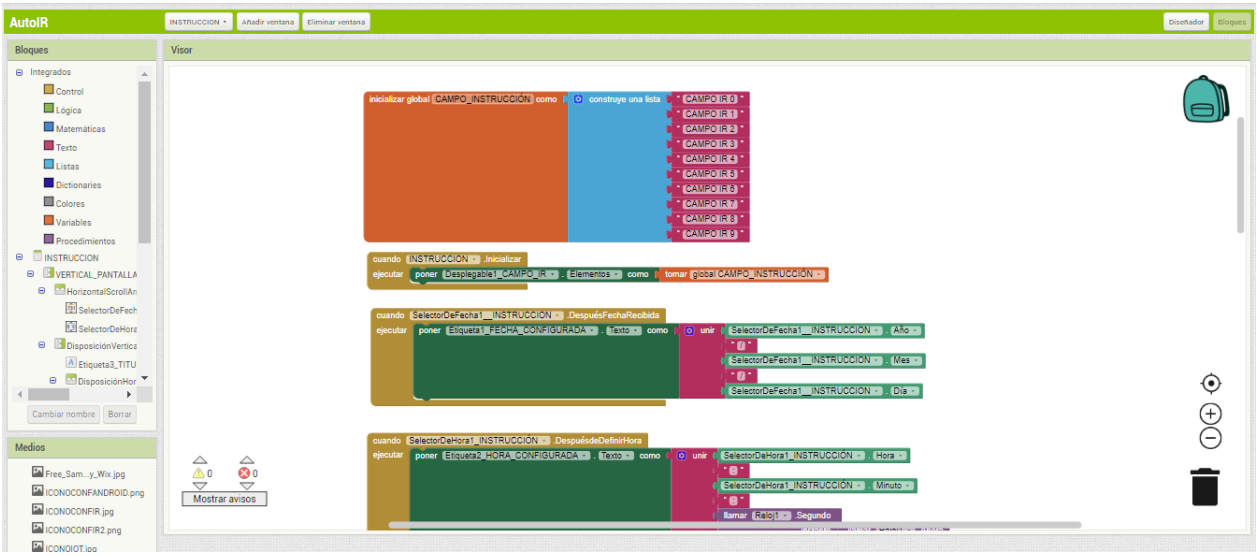

Figura 20. Desarrollo de programación en bloques Menú Instrucciones. Autoría propia

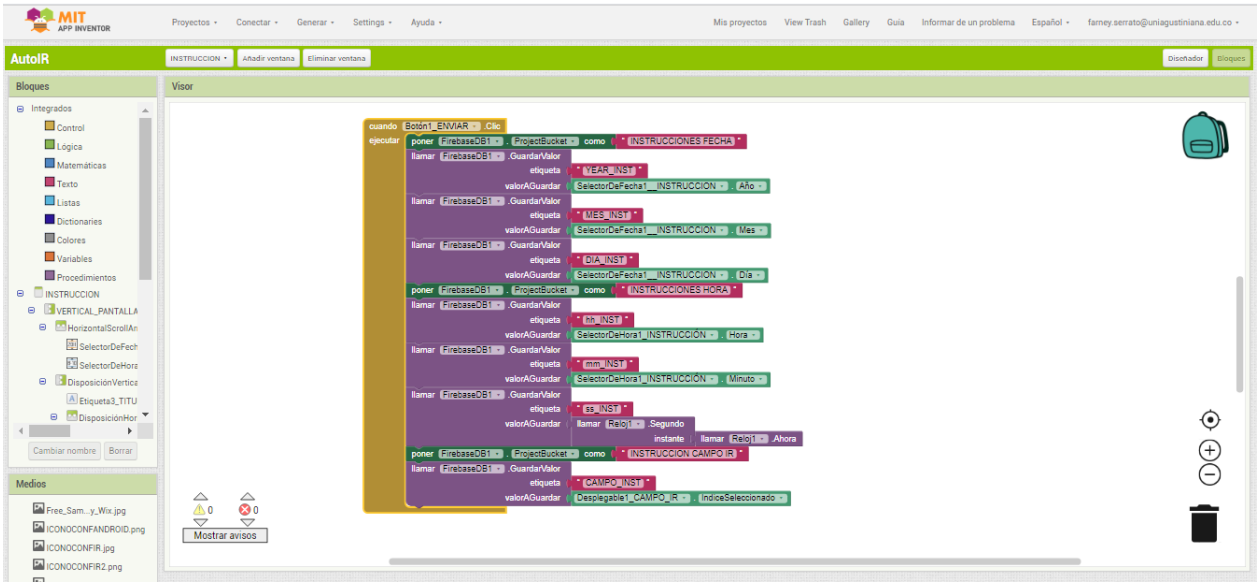

Figura 21. Desarrollo de programación en bloque Menú Instrucciones 2. Autoría propia

Lógica del software

La logica de la aplicación movil estara explicada en cuatro diagramas de flujo haicendo referencia a los cuatros botones de la APP.

El primer diagrama se visualizara en la figura 22 y muestra el proceso realizado cuando el usuario presione el boton de lectura IR, seleccione el campo en memoria de la EEPROM del D1R32 y guarde los pulsos del protocolo NEC en el campo que desea el usuario

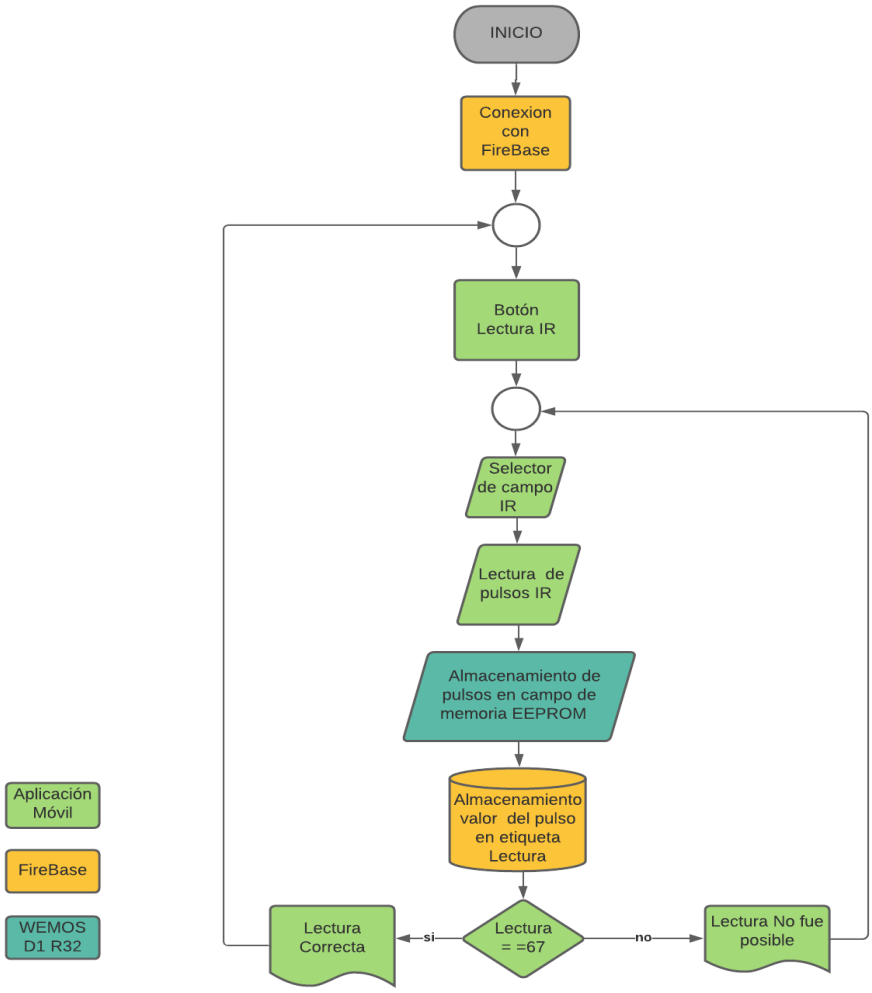

Figura 22. Lógica Botón Lectura IR. Autoría propia

El botón a continuación es el de Instrucción IoT figura 23 en este se muestra el proceso cuando el usuario obture este botón y envía la información a Firebase para que el D1 R32 la ejecute según lo configurado

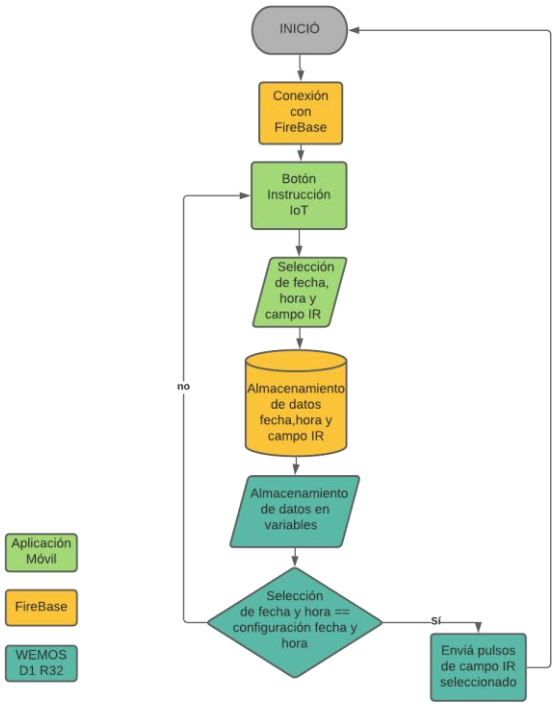

Figura 23. Lógica Botón Instrucción IoT. Autoría propia

El tercer diagrama que se visualizara es el del botón de configuraciones del sistema figura 24, este explicara la lógica cuando el usuario realiza cualquier cambio en la configuración de la fecha y hora del sistema

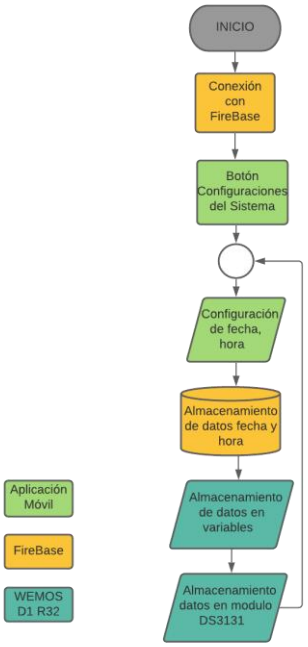

Figura 24. Lógica Botón Configuraciones del Sistema. Autoría propia

El cuarto y último botón es él Acerca de, figura 25, en este se muestra el proceso cuando el usuario ingrese a esta opción y revise los datos del desarrollador del sistema y la versión.

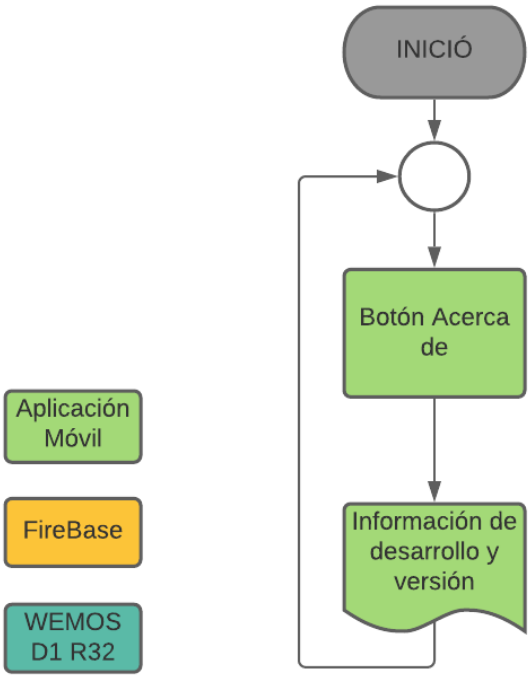

Figura 25. Lógica Botón Acerca de. Autoría propia

Codificación en Arduino.

El desarrollo de la programación está constituido por un código principal el cual toma los campos enviados por el servidor de Firebase según el dato recibido el D1 R32 realiza una función como configurar la hora del módulo RTC3231, configurar el campo de memoria de la EPROOM donde se guarda la señal obtenida por el receptor IR, configurar una hora y fecha para realizar una instrucción (por ejemplo, prender o apagar un dispositivo).

Librerías del código.

El código está constituido por las siguientes librerías:

- RTClib.h: fue desarrollada por Adafruit, permite trabajar con el reloj RTC_DS3231 y así poder sincronizar la hora, fecha, año y así el Arduino pueda enviar las instrucciones según lo configure el usuario.
  La sintaxis la encontramos en la figura 26

```
//#include <Wire.h>  //Solo si usas reloj fisico
#include <RTClib.h>
RTC_Millis rtc;     //Reloj por software que usa millis()
RTC_Micros rtc;     //Reloj por software que usa micros()
//RTC_DS1307 rtc;   //Reloj con DS1307
//RTC_DS3231 rtc;   //Reloj con DS3231
//RTC_PCF8523 rtc;   //Reloj con PCF8523
```

Figura 26. Sintaxis librería RTClib.h. Arduwiki (2020)

- EEPROM.h: esta librería permite configurar la memoria EEPROM que está integrada en el Arduino y así poder guardar información, los datos almacenados no se van a borrar si apaga. La sintaxis la encontramos en la figura 23-

```
#include <EEPROM.h>
EEPROM.write(dirección, dato);
EEPROM.read(dirección);
```

Figura 27. Sintaxis librería EEPROM.h. Arduwiki (2020)

- WiFi.h: esta biblioteca permite que una placa Arduino se conecte a Internet. Puede servir como servidor que acepta conexiones entrantes o como cliente que realiza conexiones salientes. La biblioteca admite el cifrado WEP y WPA2 Personal, pero no WPA2 Enterprise. La sintaxis la encontramos en la figura 24.

```
#include <WiFi.h>
```

Figura 28. Sintaxis librería Wifi.h. Arduino.cc (2020).

- Firebase ESP32.h: Biblioteca cliente Arduino de Google Firebase Realtime Database para Espressif ESP32, Esta biblioteca cliente proporciona las operaciones más confiables para leer, almacenar, actualizar, eliminar, respaldar y restaurar los datos de la base de datos Firebase Realtime. La sintaxis la encontramos en la figura 25.

```
#include <FirebaseESP32.h>
```

Figura 29. Sintaxis librería Firebase ESP32. Autoría propia

Lectura de los pulsos ir

Para la lectura de los pulsos que envía el mando de los electrodomésticos se creó una función llamada readCode, esta recibe el pulso primer pulso y compara si es un pulso bajo (0) de ser así inicia a contar cuánto dura este pulso hasta que cambie a un pulso alto (1) y guarda este pulso bajo en la variable lowDuration si esta valor guardado es menor a 50001 lo guarda en un vector buffer[i] como se muestra en la figura 30.

```
int readCode()
{
  int i = 0;
  unsigned long startTime;
  unsigned long endTime;
  unsigned long lowDuration = 0;
  unsigned long highDuration = 0;
  while(digitalRead(irRxPin) == HIGH) {}; // espera el primer pulso
  while(highDuration < 50001)
  {

     startTime = micros();
     while(digitalRead(irRxPin) == LOW) {};
     endTime = micros();
     lowDuration = endTime - startTime;
     if (lowDuration < 50001)
     {
       buffer[i] = (byte)(lowDuration >> 4);
       i ++;
     }
```

Figura 30. Función readCode conteo tiempo de los pulsos bajos. Autoría propia

Una vez realice el cambio de estado a alto empezara a contar la duración de este pulso hasta que de nuevo vuelva cambiar al estado bajo, y lo guarda en una variable que se llama highDuration si el valor guardado es menor que 50001 lo guarda en un vector buffer [i] como se visualiza en la figura 31.

```
      startTime = micros();
      while(digitalRead(irRxPin) == HIGH) {};
      endTime = micros();
      highDuration = endTime - startTime;
      if (highDuration < 50001)
      {
        buffer[i] = (byte)(highDuration >> 4);
        i ++;
      }
    }
     buffer[0] = 242;
    buffer[1] = 242;
    return i;
}
```

Figura 31. Función readCode conteo tiempo de los pulsos altos. Autoría propia

Conexiones de los dispositivos

Para el desarrollo del hardware del proyecto se usó un dispositivo WeMos D1 R32 el cual tiene un procesador ESP32 y es el encargado de leer la información almacenada en el servidor de Firebase para ejecutar las instrucciones correspondientes según el usuario lo desee, para controlar la fecha y hora del sistema se utilizó el módulo DS3231 que es el encargado de esta función, para la recepción de los pulsos IR se empleó un diodo led receptor IR el cual capta las señales enviadas desde los mandos del dispositivo que se quiere controlar como se muestra en la figura 33 (RX IR), el transmisor IR es el encargado de enviar los pulsos que se encuentran en la memoria EEPROM el flujo de la transmisión la encontramos en la figura 33 (TX IR) según la configuración que realice el usuario, todos estos dispositivos esta interconectados, figura 32.

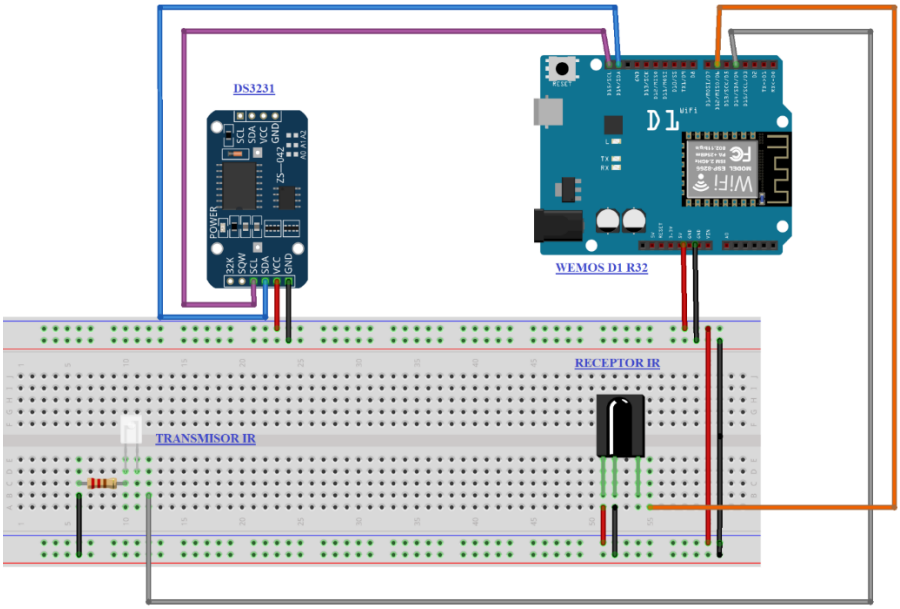

Figura 32. Conexiones hardware. Autoría propia

Topología de la red y flujo de datos

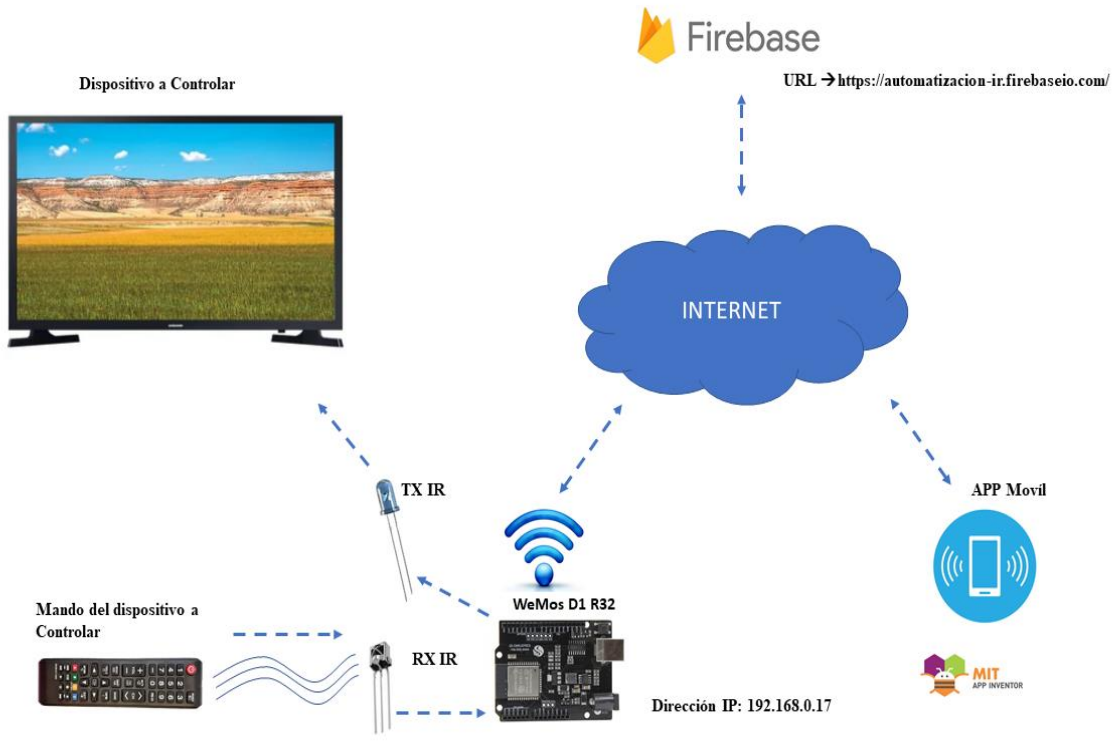

Figura 33. Topología de la red y flujo de datos. Autoría propia

Pruebas de funcionamiento del sistema

Para iniciar las pruebas del funcionamiento del sistema se instaló la aplicación para sistemas en un smartphone con sistema operativo Android ejecutando el APK que se obtuvo del entorno APP inventor, una vez instalado en la pantalla del celular se va a visualizar el icono de esta app como se muestra en la imagen 34

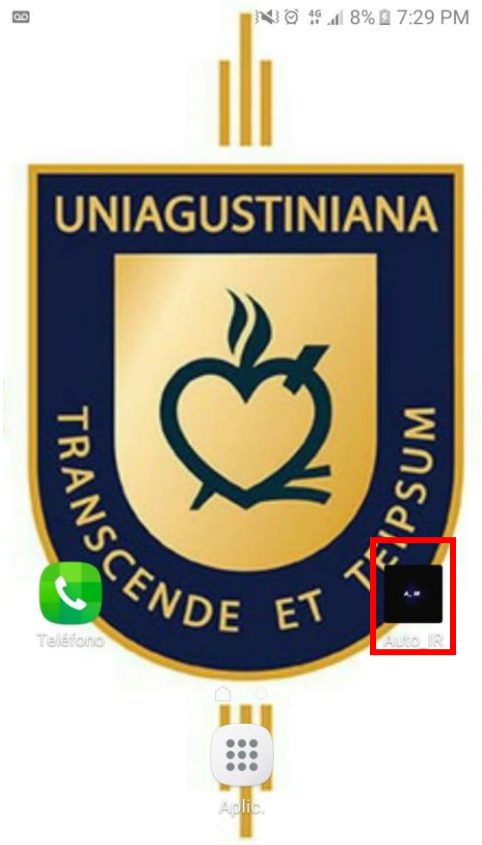

Figura 34. Logo Auto_IR. Autoría propia

Al oprimir el logo se va a iniciar la aplicación e ingresara al screen principal de esta, se presiona el botón de configuraciones del sistema, figura 35.

Figura 35. Indicación para ingresar al menú configuraciones del sistema. Autoría propia

Y encontraremos en la parte superior la fecha y hora actual que servirá como guía para configurar esta sin tener que mirar un reloj u otro dispositivo (figura 36)

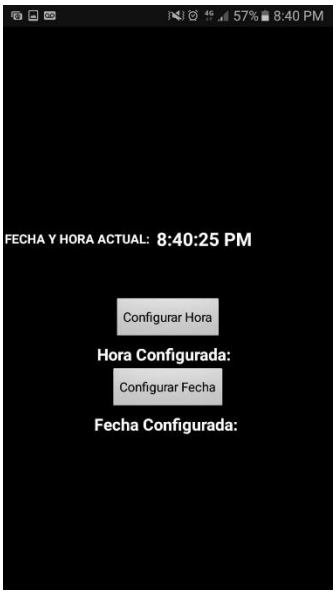

Figura 36. Menú configuraciones del sistema. Autoría propia

Se procedió a configurar estos datos presionando el menú configurar hora y fecha respectivamente como lo evidencia la figura 37 y 38.

Figura 37. Configuración hora del sistema. Autoría propia

Figura 38. Configuracion Fecha del sistema. Autoría propia

Para confirmar que estas configuraciones fueron enviadas al D1 R32 se realizó una configuración en el código de Arduino para que se muestre está en el monitor serial

Figura 39. Prueba configuraciones del sistema monitor serial Ardunio. Autoría propia

Como se visualiza en la figura 39 las configuraciones fueron enviadas correctamente al D1 R32 lo que nos indica que el modulo DS3231 empezara a operar con los datos anteriormente mencionados.

La siguiente función a realizar prueba es la lectura IR, para este vamos a ingresar al botón que nos dirigirá a este menú como nos muestra la figura 35

Figura 40. Indicación para ingresar al Menú Lectura IR. Autoría propia

Cuando el usuario ingresa va a encontrar un menú muy sencillo donde seleccionara el campo donde quiere que quede guardado el botón o acción a realizar

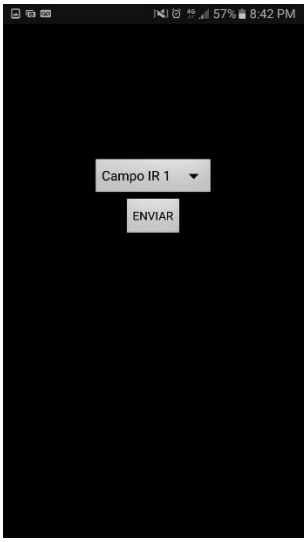

Figura 41. Menú lectura IR. Autoría propia

Al presionar el selector se va a desplegar las opciones de almacenamiento como se muestra en la figura 42

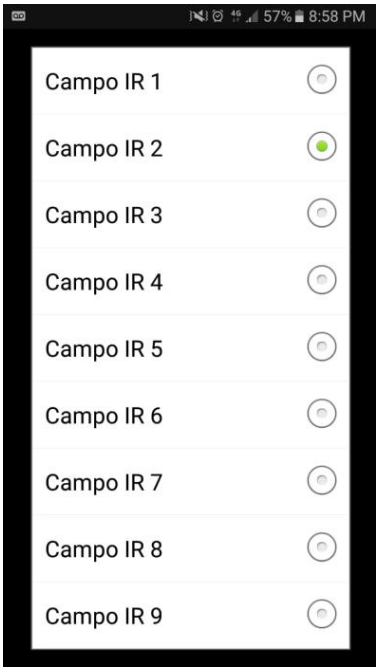

Figura 42. Selector de Campo IR. Autoría propia

Una vez seleccionado el campo en la aplicación se mostrará un mensaje de confirmación con la selección realizada para que el usuario tenga certeza de esta.

Figura 43. Confirmación selección campo IR. Autoría propia

La aplicación volverá al menú de lectura IR para que se confirme presionando en el botón "ENVIAR" seguido de esta acción se abrirá el menú de carga de lectura IR, donde se debe oprimir el botón y apuntar el control remoto del electrodoméstico al sensor IR

Figura 44. Menú carga de lectura IR. Autoría propia

Una vez se oprima el botón apuntando el control remoto al sensor se procede a presionar el botón enviar si la lectura de los pulsos IR son correctas se mostrará un mensaje confirmando esta acción, sino es correcta de igual manera se mostrará un mensaje de alerta para que el usuario repita los pasos anteriores, para confirmar que los campos fueron enviados se revisó el monitor serial

Figura 45. Pruebas campo seleccionado monitor serial Arduino. Autoría propia

Como evidenciamos en la figura 45 fue enviado correctamente los datos al D1 R32 y fue almacenado en el campo dos de la memoria EPROOM de este dispositivo.

Por último, se realizó pruebas del menú IoT o menú de instrucciones donde el usuario escogerá la fecha, hora y el campo que desee ejecutar, para ingresar a este se oprime el botón IOT como se muestra en la imagen 41

Figura 46. Indicación para ingresar al menú IoT. Autoría propia

Después de obturar el botón la aplicación abrirá el menú donde ese encuentran las instrucciones que el usuario desee configurar

Figura 47. Menú instrucciones o IoT. Autoría propia

Cuando el usuario seleccione la configuración de fecha se abrirá un selector para que ingrese la fecha deseada y confirme está presionando la opción Aceptar.

Figura 48. Selector para configuración de fecha IoT. Autoría propia

Igual que la instrucción anterior se realiza para la selección de la hora abriendo el selector para la hora

Figura 49. Selector para configuración de hora IoT. Autoría propia

Por último, es seleccionar el campo IR que desea activar siguiendo las instrucciones que se configuraron anteriormente

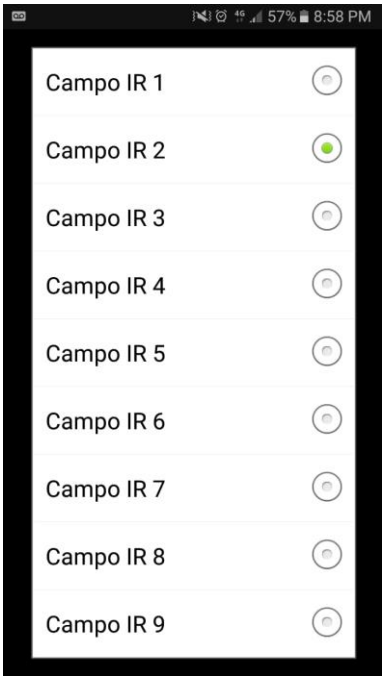

Figura 50. Selector de campo IR IoT. Autoría propia

Una vez realizada las configuraciones anteriores se confirma estas oprimiendo el botón ENVIAR

Figura 51. Configuraciones realizadas IoT. Autoría propia

Confirmamos que las instrucciones fueron enviadas al D1 R32 en el monitor serial y que se ejecute a la fecha, hora y campo

Figura 52. Prueba IoT monitor serial Arduino. Autoría propia

Como se muestra en la imagen 52 las instrucciones fueron enviadas correctamente y se activó a la hora que se selecciono

Video de pruebas del sistema

Para visualizar el correcto funcionamiento del sistema se creó un video donde explico paso a paso el proceso que debe realizar un usuario para controlar un dispositivo con mandos IR, este esta anexa a la entrega del presente documento.

Conclusiones

Después de recopilar la información sobre los protocolos IR NEC, RC5, RC6 Y SRC, se encontró que tienen una estructura muy similar, cuentan con un inicio, una dirección y el código del botón o acción, lo que permite tener un mejor análisis del protocolo NEC y su comportamiento en la lectura IR del proyecto.

Al desarrollar la aplicación para Android en el entorno APP INVENTOR se logró crear y configurar los menús de manera muy sencilla sin tener que usar algún lenguaje de programación complejo ya que este usa diagramas de bloques lo que hace intuitivo su desarrollo.

La plataforma Firebase de Google almacena la información enviada desde la aplicación móvil en tiempos de aproximadamente dos segundos y sin presentar perdidas, alrededor de cuatro segundos el WeMos D1 R32 recibe la información almacenada que tiene Firebase para ser ejecutada.

La memoria EEPROM del dispositivo WeMos D1 R32 soporto los bits de los comandos IR almacenados durante las pruebas sin presentar volcamiento o errores que impidiera la recepción o transmisión de los pulsos IR.

Referencias

AI Thinker. (2020). Obtenido de Anxinke Technology CO;LTD: https://www.mikrocontroller.net/attachment/338570/Ai-thinker_ESP-07_WIFI_Module-EN.pdf

ALTIUM. (2017). *ALTIUM* . Obtenido de https://techdocs.altium.com/display/FPGA/NEC+Infrared+Transmission+Protocol

ALTIUM. (2017). *ALTIUM.* Obtenido de https://techdocs.altium.com/display/FPGA/Philips+RC5+Infrared+Transmission+Protocol

Arduino C. (2020). Obtenido de https://store.arduino.cc/usa/arduino-uno-rev3

Congreso de la Republica. (2019). Obtenido de http://www.secretariasenado.gov.co/senado/basedoc/ley_1480_2011.html

DE SENSORES. (2020). Obtenido de https://desensores.com/sensores-arduino/proyectos-basicos-con-arduino-para-principiantes/modulo-transmisor-receptor-ir-arduino/

DEVELOPERS. (2014). Obtenido de https://developer.android.com/studio/intro?hl=es-419

Diosdado, R. (s.f.). *Zona Maker*. Obtenido de https://www.zonamaker.com/arduino/modulos-sensores-y-shields/control-a-distancia-mediante-mando-infrarrojo-ir

Electronica CIDEA . (2020). *Electronica CIDEA*. Obtenido de http://electronicacidea.com.mx/store/index.php?route=product/product&product_id=617

Garcia, V. (2014). *HISPAVILA*. Obtenido de https://www.hispavila.com/control-remoto-ir/

https://www.prometec.net/infrarrojos/. (s.f.). *Prometec*.

Llamas, L. (2016). Obtenido de https://www.luisllamas.es/arduino-mando-a-distancia-infrarrojo/

O.S GROUP . (2020). *O.S. GROUP* . Obtenido de https://www.osgroup.co/que-es-un-servidor-web/#:~:text=El%20proceso%20es%20un%20ejemplo,pronunciado%20motor%20X%20)%20de%20NGNIX.

Saiz, S. (2015). *Internet de las cosas: Desarrollo de un servidor.* Obtenido de https://riunet.upv.es/bitstream/handle/10251/56081/SORIANO%20-%20Internet%20de%20las%20cosas%3A%20Desarrollo%20de%20un%20servidor%20Dom%C3%B3tico.pdf?sequence=2

Sampieri, R. (2014). Metodologia de la Investigacion Sexta edicion. En R. H. Sampieri. Mexico.

Garcia. V. (2015). Control remoto. Obtenido de https://www.hispavila.com/control-remoto-ir/