

UNIVERSIDAD MAYOR DE SAN ANDRES

FACULTAD DE TECNOLOGIA

CARRERA DE ELECTRONICA Y TELECOMUNICACIONES



**SISTEMA DE CONTROL PARA UN EQUIPO DE TELEVISION
BASADO EN EL RECONOCIMIENTO DE PATRONES DE VOZ**

Proyecto de Grado presentado para obtener el Grado de Licenciatura

POR: JAZMINA TERESA DEL ROCIO GAMBARTE VARGAS

TUTOR: Ing. LUIS RAMIRO VELARDE CHAVEZ

LA PAZ – BOLIVIA

2023

DEDICATORIA:

Ofrezco esta dedicatoria a mis padres Peter Mario Gambarte Mejia y Maria Susana Vargas Velasquez por su apoyo, paciencia y cariño que me dieron en todo momento, a mis hermanas Karla y Cindy por su compañía, ayuda y motivación para así poder culminar mis estudios en esta casa superior.

AGRADECIMIENTOS

A mi carrera por permitirme realizarme como profesional y brindarme todas las oportunidades para el ejercicio de mi profesión.

A todos los catedráticos de mi querida facultad quienes son gestores de aprendizaje, en especial al Ing. Ramiro Velarde por su inestimable asesoramiento en los aspectos técnicos de este Proyecto, por su apoyo y por la confianza depositada en mí.

INDICE

CAPITULO 1	1
1. Antecedentes del Proyecto.	1
1.1. Planteamiento del problema	2
1.2. Objetivos	2
1.2.1. General	2
1.2.2. Específicos	2
1.3. Justificación	3
1.3.1. Justificación técnica	3
1.3.2. Justificación económica	3
1.3.3. Justificación social	3
1.4. Delimitación Espacial.....	4
1.5. Metodología	4
a) Enfoque Metodológico	4
b) Tipo de la investigación	4
CAPITULO 2	5
2.1. Marco Teórico.	5
2.1.1. Personas de la tercera edad.	5
2.1.2. Personas con discapacidad.	7
2.1.3. Sistemas de control de voz.	10
2.1.4. Codificador.	15
2.1.5. Decodificador.	16
2.1.5.1. Aplicaciones del Decodificador.	17
2.1.6. Filtros.....	17
2.2. Fundamento Teórico Aplicativo.....	23
2.2.1. Arduino.....	23
2.2.2. Lenguaje de programación C++	30
2.2.3. Módulo Bluetooth.	32
2.2.4. Aplicación App Inventor.	37
2.2.5. Led infrarrojo:.....	42
2.2.6. Transistor:.....	44
2.2.7. Televisor	46
2.2.7.1. Televisor LG.	46

2.2.7.2.	Televisor Samsung	47
2.2.7.2.1.	Características.	47
CAPITULO 3		49
3.1.	Desarrollo Practico Experimental.	49
3.1.1.	Arduino.....	51
3.1.2.	Circuito simulado en ISIS.	52
3.1.3.	Programa del Arduino Uno.	54
3.1.4.	Sistema de detección de fallas.....	58
3.1.5.	App Inventor.....	63
3.1.5.1.	Código de App Inventor.....	64
CAPITULO 4		68
4.1.	Análisis de Costos.	68
4.1.1.	Costos Directos.....	68
4.1.2.	Costos Indirectos.....	69
4.1.3.	Costo total del proyecto.....	69
CAPITULO 5		70
5.1.	Conclusiones.	70
5.2.	Recomendaciones.	70

RESUMEN

El reconocimiento de voz es el proceso de reconocer automáticamente a una persona en base a la información emitida por su voz. Esta técnica permite hacer uso de la voz como herramienta en la verificación de la identidad de una persona con la finalidad de controlar diversas aplicaciones mediante comandos de voz. Esta tecnología es aprovechada para diversas aplicaciones, en el presente proyecto dicha tecnología se aplica para ayudar a un sector vulnerable de la población que son las personas mayores de edad, personas discapacitadas o personas con características especiales.

A lo largo del presente proyecto, se observará el análisis, diseño y la implementación del software de reconocimiento de voz utilizando el lenguaje de programación Arduino, que a su vez interactúa con hardware externo a un dispositivo móvil mediante una interfaz Bluetooth.

La interfaz Bluetooth que se implementa en el presente proyecto es el módulo Bluetooth HC-05 viene configurado como esclavo, es decir, listo para escuchar solicitudes de conexión, también se puede configurar como maestro por medio de comandos AT.

Para el control del televisor se realizó el registro de comandos del control remoto del mismo, cargando estos en el Arduino Uno logrando operar el televisor por comandos de voz.

Con todo lo expuesto anteriormente, el presente proyecto pretende mejorar la calidad de vida de las personas mayores de edad, personas discapacitadas o personas con características especiales, desarrollando para tal fin la implementación de un sistema de reconocimiento de voz para el control de un equipo de televisión utilizando lenguaje de programación Arduino.

INTRODUCCION

A través del tiempo, el hombre en su afán por simplificar sus tareas cotidianas ha desarrollado métodos y herramientas que le han permitido, mejorar su calidad de vida. Como resultado de esa búsqueda constante e incansable, se ha realizado grandes avances científicos y tecnológicos en diversos campos del saber, dentro de los cuales podemos destacar el procesamiento digital de señales específicamente el reconocimiento de patrones de voz.

Estas tecnologías, de manera general, se pueden englobar en un solo objetivo, crear canales de comunicación adecuados a las capacidades de las personas. Algunas de estas personas se ven con la problemática de no contar con ningún tipo de movimiento físico para poder comunicarse, como es el caso de las personas cuadripléjicas que están aisladas de su entorno a pesar de que su habla goza de total capacidad, y es torno a estas características que se ha buscado dentro de un área muy específica, encontrar una solución a esta problemática, lo cual nos conduce al reconocimiento de voz.

El reconocimiento de voz es el proceso de reconocer automáticamente a una persona en base a la información emitida por su voz. Esta técnica permite hacer uso de la voz como herramienta en la verificación de la identidad de una persona con la finalidad de controlar diversas aplicaciones mediante comandos de voz.

Sistema de control para un equipo de televisión basado en el reconocimiento de patrones de voz

CAPITULO 1

1. Antecedentes del Proyecto.

La tecnología a lo largo de los años ha evolucionado de forma significativa en lograr que el esquema de comunicación de los seres humanos haya conseguido una representación efectiva a través de la conjunción de una aplicación informática y su desarrollo en una interfaz sencilla de manejar, para que el usuario final pueda llevar a cabo las tareas cotidianas desde su casa y simplemente utilizando la voz como herramienta para la conclusión de su objetivo final.

Por ejemplo, se puede encontrar en casi cualquier edificio una puerta que se abre automáticamente ante la presencia de un caminante, también podemos ver baños cuyos grifos se cierran o abren cuando algún objeto se les acerca y las luces en algunos corredores que solo se encienden si algo se mueve, todos los adelantos en tecnología pueden parecer comunes y ordinarios, sin embargo, se ha necesitado de algún tiempo y esfuerzo por parte de los creadores para que sea así.

La diferencia del presente proyecto de grado con los anteriores mencionados es que se diseñara e implementar un sistema de control para un equipo de televisión, mediante el reconocimiento de patrones de voz utilizando lenguaje de programación Arduino, para personas mayores de edad, personas discapacitadas o personas con características especiales.

Estas personas al tener limitaciones físicas, siendo estas principalmente de tipo motriz, requieran de mucha ayuda a la hora de realizar las actividades que para una persona normal serían muy fáciles de hacer.

1.1. Planteamiento del problema

Las personas mayores de edad, cuadripléjicas y con características especiales tienen limitaciones de movimientos lo que hace que las mismas no sean autónomas ni independientes en su desenvolvimiento diario, por ejemplo, no les es fácil hacer uso de los equipos electrónicos que poseen en su hogar, tal es el caso del televisor que es uno de los equipos de mayor distracción dentro del hogar.

A raíz de estos percances estas personas precisan de las nuevas tecnologías para poder desenvolverse en la vida cotidiana, sobre todo, de la manera más cómoda posible, ya que, si bien en el mercado existen televisores con control de voz, estos tienen costos elevados lo que limita el poder adquirir para un centro de personas mayores de edad o discapacitadas.

1.2. Objetivos

1.2.1. General

Implementar un sistema de control de reconocimiento de patrones de voz basado en el lenguaje de programación con Arduino para un equipo de televisión destinado a personas de la tercera edad, cuadripléjicas y con discapacidad física

1.2.2. Específicos

- Realizar el algoritmo de reconocimiento de voz por software.
- Desarrollar una aplicación para sistema operativo Android con compatibilidad de reconocimiento de voz para personas de la tercera edad.
- Implementar el sistema electrónico con integración, para la aplicación móvil en el envío y recepción de información con el equipo de televisión.

1.3. Justificación

1.3.1. Justificación técnica

Se justifica técnicamente debido a que en el presente proyecto se innovará la forma de control de un equipo de televisión ya que el control se realizará mediante comandos de voz, utilizando el lenguaje de programación Arduino y desarrollando una aplicación móvil para el sistema operativo Android, y así contribuir con el avance de la tecnología.

1.3.2. Justificación económica

La justificación económica la analizaremos desde dos puntos de vista: la inversión que se realizará para llevar a cabo el proyecto es mínima porque se utilizan dispositivos Arduino y los beneficios que se obtendrán para personas de edad avanzada serán mayores en comodidad.

1.3.3. Justificación social

Se justifica socialmente debido a que con el presente proyecto se pretende ayudar a mejorar la calidad de vida de un sector vulnerable de la población que son las personas de la tercera edad; mediante el presente proyecto se pretende que las personas de edad avanzada tengan mayores facilidades de acceso a la tecnología. Con todo lo expuesto anteriormente se pretende que la vida de estas personas mejore en varios aspectos, ya que hoy en día la sociedad en su mayoría cuenta con un dispositivo móvil.

1.4. Delimitación Espacial

El proyecto se enmarca en las personas de la tercera edad el cual se implementara en la Residencia de descanso para adultos mayores o Residencia Geriátrica La Estancia, que se encuentra ubicado en la zona de Irpavi II, calle 5 Las Retamas № 28, o centros para personas con capacidades especiales.

1.5. Metodología

a) Enfoque Metodológico

La perspectiva de esta investigación será cuantitativa utilizando el método deductivo, es decir va de lo más general a lo más específico o particular que en este caso son los proyectos aplicados a un microcontrolador específico.

b) Tipo de la investigación

La investigación será del tipo Aplicada, ya que tiene por objetivo resolver un determinado problema o planteamiento específico, enfocándose en la búsqueda y consolidación del conocimiento para su aplicación y, por ende, para el enriquecimiento del desarrollo científico de la carrera de Electrónica y Telecomunicaciones.

El trabajo se encuentra en el área de I+D (investigación y desarrollo), específicamente. Esta área se refiere a las actividades de investigación que lleva a cabo la carrera de Electrónica y Telecomunicaciones para mejorar las investigaciones existentes y conducir al desarrollo de nuevos productos y procedimientos.

CAPITULO 2

2.1. Marco Teórico.

2.1.1. Personas de la tercera edad.

Se denomina envejecimiento al proceso continuo, heterogéneo, universal e irreversible que determina una pérdida de la capacidad de adaptación de forma progresiva. Asimismo, es un fenómeno extremadamente variable, influido por múltiples factores arraigados en el contexto genético, social e histórico del desarrollo humano, cargado de afectos y sentimientos. En esta etapa del ciclo de vida, todas las estructuras desarrolladas en la etapa anterior decaen, con cambios a nivel físico, cognitivo, emocional y social que siguen. Si bien esta etapa tiene un punto final definido (la muerte), la edad de inicio no se ha determinado definitivamente porque no todos los individuos envejecen de la misma manera. Sin embargo, dado que la edad biológica es un indicador del estado real del cuerpo, se considera que tiene 65 años o más. (Alvarado & Salazar, 2014)



Figura 1: Adultos Mayores en una casa de reposo

Fuente: Aguilar R., Druet N. (2020) Aprendizaje vivencial en la tercera edad [Figura] Recuperado de <https://observatorio.tec.mx/edu-bits-blog/aprendizaje-vivencial-tercera-edad>

2.1.1.1. Clasificación de las personas mayores.

1. Persona mayor (sana o enferma) autónoma: incluye a las personas mayores de 65 años, sanas y afectas de enfermedades (agudas o crónicas) no potencialmente incapacitantes.
2. Persona mayor frágil o de alto riesgo de dependencia: este concepto indica situación de riesgo de dependencia (pero sin padecerla aún), siendo potencialmente reversible, o pudiendo mantener la autonomía mediante una intervención específica.
3. Paciente geriátrico: si cumple tres o más de las siguientes condiciones:
 1. Mayor de 75 años
 2. Pluripatología
 3. Incapacidad funcional (alto riesgo de dependencia)
 4. Deterioro mental
 5. Problemática social en relación con su estado de salud

2.1.1.2. Fragilidad.

La fragilidad es un estado asociado al envejecimiento, que se caracteriza por una disminución de la reserva fisiológica que se traduciría en el individuo en un aumento del riesgo de incapacidad, una pérdida de la resistencia y una mayor vulnerabilidad ante eventos adversos, y que se manifiesta en mayor morbilidad y mortalidad. (Jauregui & Rubin, 2012)

En ese sentido este estado se traduce en:

- Pérdida de resistencia (la sarcopenia o pérdida de masa muscular es un factor clave en la debilidad)
- Mayor vulnerabilidad
- Mayor riesgo de discapacidad

2.1.2. Personas con discapacidad.

Una discapacidad es un impedimento físico, intelectual o sensorial en algunas personas que afecta la forma en que interactúan y participan plenamente en la sociedad a largo plazo. El término handicap se considera despectivo fuera del ámbito legal.

La Convención Internacional sobre los Derechos de las Personas con Discapacidad, ratificada por las Naciones Unidas en 2006, generalmente define a las personas con una o más discapacidades como personas con discapacidad. En algunas áreas, términos como "discapacitado", "ciego", "sordo", incluso cuando se usan correctamente, pueden considerarse despectivos o despectivos porque para algunas personas estos términos "marcan" a quienes sufren de personas con discapacidad, que interpretan como una forma de discriminación. En estos casos, para evitar conflictos semánticos, a veces se utilizan las formas "funcionalmente diverso" o "discapacitado".



Figura 2: Personas discapacitadas en un maratón

Fuente: Mejía J. (2020). Celebremos el día de las personas con discapacidad. [Figura]. Recuperado de <https://www.habitantesiete.com/celebremos-el-dia-de-las-personas-con-discapacidad/>

2.1.2.1. Tipos de discapacidad.

El tipo de discapacidad puede ser:

- Impedimento físico o de movimiento: Una pérdida o limitación permanente del movimiento de una persona.

- Alteración sensorial

 - o Visión: pérdida de la visión o dificultad para ver con un ojo.

 - o Audición: Pérdida de audición o audición limitada.

 - o Idioma: Pérdida o limitación del idioma.

- Discapacidad mental: subdividida en:

 - o Discapacidad intelectual: Incluye limitaciones en el aprendizaje de nuevas habilidades.

 - o Trastornos psicosociales: Limitaciones para establecer relaciones sociales y/o afectivas. Un ejemplo de esto es el trastorno del espectro autista.



Figura 3: Personas con diferentes tipos de discapacidad

Fuente: Egorov V. (2017). Ayudando a una persona discapacitada. [Figura], Recuperado de <https://alamy.es/ilustracion-de-estilo-plano-conjunto-de-personas-con-discapacidad>

2.1.2.2. Personas con lesiones de la médula espinal

Una lesión de la médula espinal, lo que implica daño en cualquier parte de la médula espinal o en los nervios del extremo del canal espinal (cola de caballo), a menudo causa cambios permanentes en la fortaleza, la sensibilidad y otras funciones del cuerpo debajo del sitio de la lesión.

Además, la parálisis por una lesión de la médula espinal puede denominarse:

- **Tetraplejía.** También conocida como cuadriplejía, esto significa que los brazos, las manos, el tronco, las piernas y los órganos pélvicos están afectados por la lesión de la médula espinal.
- **Paraplejía.** Esta parálisis afecta la totalidad o parte del tronco, las piernas y los órganos pélvicos.

Las primeras siete vértebras del cuello se denominan vértebras cervicales. La primera vértebra de arriba es C1, la siguiente es C2, y así sucesivamente. Las lesiones de la médula espinal cervical a menudo resultan en la pérdida de la función del brazo y la pierna, lo que resulta en tetraplejía. (Mayo Foundation for Medical Education and Research, 2022)



Figura 4: Persona cuadripléjica

Fuente: Matzner P. (2014). Tetraplejía: causas, tratamientos y cuidados. [Figura]. Recuperado de <https://www.sunrisemedical.es/blog/tetraplejia>

2.1.3. Sistemas de control de voz.

El crecimiento de los sistemas de reconocimiento de voz ha ido creciendo a lo largo de los años. Originalmente, las voces pregrabadas con menús de opciones se usaban en aplicaciones telefónicas automatizadas. Estos representan opciones que el usuario puede seleccionar presionando un botón en el teléfono, que luego actúa como un reconocedor de comandos de voz donde el usuario puede indicar cualquier palabra en el menú de voz (De Pablos, Fernando;, 2008). Posteriormente, las aplicaciones de telefonía han crecido considerablemente, lo que permite una mayor flexibilidad en el reconocimiento y la síntesis de voz. En el mercado informático, a su vez, se han comercializado programas de reconocimiento y síntesis de voz que se ejecutan en ordenadores personales, incluyendo software comercial y freeware. Estos programas permiten el dictado automático de documentos, el control del sistema operativo y la navegación web a través de comandos de voz. Son independientes del hablante y permiten el entrenamiento previo del sistema, lo que mejora la precisión del reconocimiento, ya que los usuarios pueden corregir los errores a medida que ocurren. De esta forma, el sistema aprende dinámicamente y cuanto más se usa, mayor es la eficiencia. Las ventajas de estas tecnologías para el usuario son la velocidad de generación de documentos y la comodidad de operar el sistema operativo. Las desventajas que se encuentran comúnmente en muchas aplicaciones de control de voz son la capacitación previa, lo que significa una pérdida de tiempo inicial, y la falla de reconocimiento, lo que significa que el trabajo que se está realizando se demora en decirle al sistema qué es.

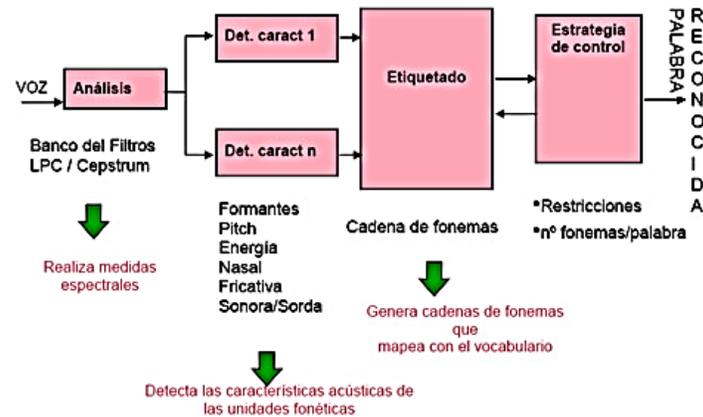


Figura 5: Decodificador acústico-fonético

Fuente: Martin F. (2008). Control por voz [Figura]. Recuperado de <https://www.casadomo.com/2008/10/20/control-por-voz>

Una técnica determinada para la parametrización reducida de la señal de voz es el reconocimiento de patrones (Carmona, 2009). Este último es muy complejo y no se utiliza en la práctica. Entre las técnicas de procesamiento basadas en patrones, cabe destacar aquellas que se apoyan en el uso de redes neuronales, por un lado, y las que realizan procesamiento según Modelos Ocultos de Markov por otro. El tercer tipo combina las dos técnicas. En cualquier caso, necesitan entrenar previamente el sistema para que el corpus de voz (base de datos) obtenga más información y mejore la tasa de éxito del reconocedor.

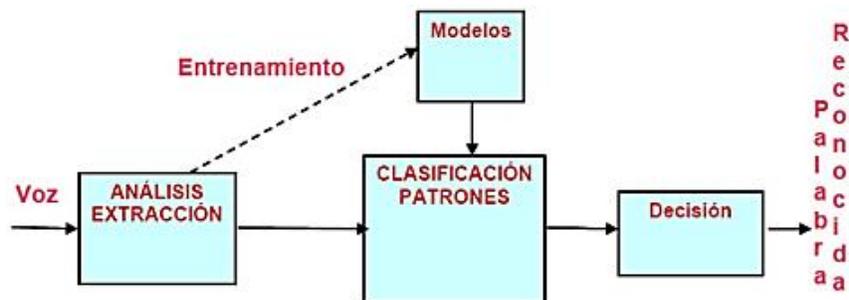


Figura 6: Esquema de entrenamiento en una fase y reconocimiento.

Fuente: Martin F. (2008). Control por voz [Figura]. Recuperado de <https://www.casadomo.com/2008/10/20/control-por-voz>

Una aplicación reciente en la tecnología del habla es el reconocimiento del hablante. Esta función detecta la huella biométrica de cada persona, el timbre, que se puede utilizar en los sistemas de autenticación o para diferenciar a los usuarios y aplicar diferentes perfiles de uso en el control del sistema.

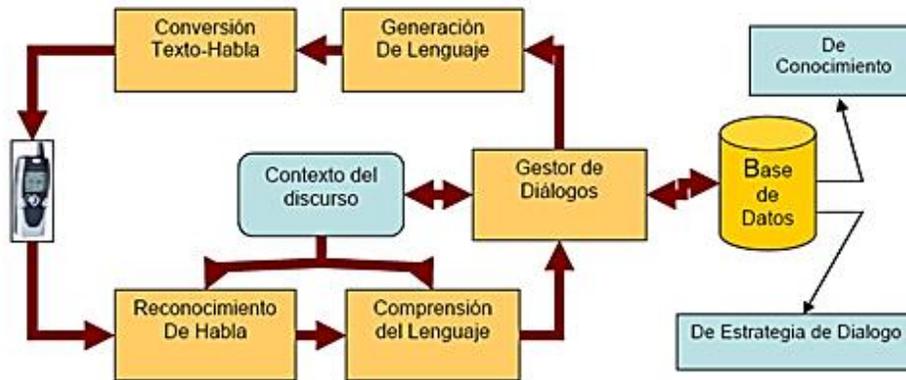


Figura 7: Aplicación de reconocimiento de voz con gestión de diálogo y síntesis de voz en respuesta a consultas telefónicas

Fuente: Martin F. (2008). Control por voz [Figura]. Recuperado de <https://www.casadomo.com/2008/10/20/control-por-voz>

Otra parte de hablar con una máquina, otra dirección de la conversación es la generación de habla artificial (text-to-speech) o síntesis de voz. Actualmente, los sintetizadores de voz logran sonidos “muy reales” que simulan emociones y cada vez se acercan más a la forma de hablar de los humanos. El sintetizador de voz permite la personalización, con la posibilidad de elegir el género, el acento y la velocidad del habla sintetizada.



Figura 8: Reconocimiento de voz que se divide en 3 partes: Pre-procesamiento, Reconocimiento y Comunicación.

Fuente: Martín F. (2008). Control por voz [Figura]. Recuperado de <https://www.casadomo.com/2008/10/20/control-por-voz>

Uno de los aspectos a considerar en la comunicación hombre-máquina es el diseño de la interfaz. El tipo más común de interfaz de usuario de voz en el servicio de atención al cliente es IVR (Respuesta de voz interactiva). Es un sistema telefónico automatizado que emplea tecnología de reconocimiento de voz para interactuar con las personas que llaman. (Quality Unit, LLC, 2022)

Además, idealmente, el sistema debería poder adaptarse a varios usuarios en función de su nivel de conocimiento sobre el sistema en sí, sus preferencias previas o el entorno en el que se encuentran. También se trata de imitar algunos fenómenos humanos típicos: confirmación durante las conversaciones, inicio de conversaciones menos fluidas, etc. Debe tener en cuenta que las conversaciones de persona a persona son fluidas, a menudo con oraciones rotas, superpuestas, incompletas o mal estructuradas. La interacción con la máquina debe estar estructurada para que los objetivos del administrador de diálogo se ejecuten correctamente.

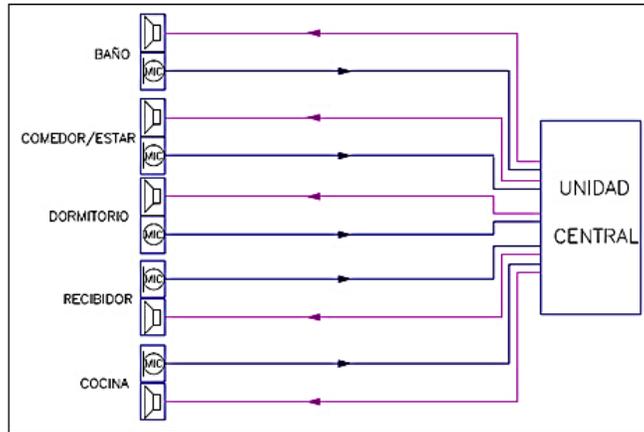


Figura 9: Topología de conexionado con cableado dedicado desde la unidad central hasta cada estancia en estrella.

Fuente: Martín F. (2008). Control por voz [Figura]. Recuperado de <https://www.casadomo.com/2008/10/20/control-por-voz>

El control por voz en el hogar digital se puede realizar de diferentes formas dependiendo del tipo de elementos de captura de voz que se utilicen y de la forma en que se transmiten las señales de audio entre los elementos que componen el sistema de control por voz. Desde un punto de vista técnico, la conexión más sencilla es una conexión en estrella desde el panel de control a cada elemento transmisor o receptor. Esta conexión se puede realizar mediante señales analógicas o digitales. En este caso de señales de transmisión digital, la longitud del cableado se puede aumentar sin pérdida de calidad en comparación con las señales analógicas. En el segundo caso, el costo económico es mayor porque tiene que usar convertidores A/D y D/A en el sistema de control y hardware de transmisión, respectivamente.

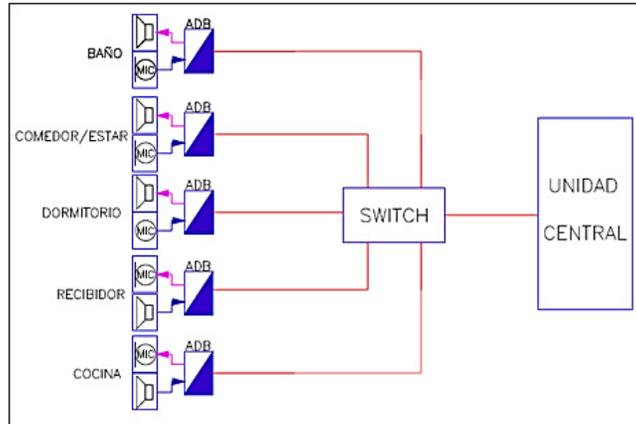


Figura 10: Topología de conexión mediante una red LAN cableada.

Fuente: Martín F. (2008). Control por voz [Figura]. Recuperado de <https://www.casadomo.com/2008/10/20/control-por-voz>

El tipo recomendado es utilizar una red LAN para transmitir señales entre los sistemas central y auxiliar de adquisición y lanzamiento. En todo sistema de captura y lanzamiento es necesario agregar un adaptador al BUS (tarjeta de red). En el mercado actual podemos encontrar una gran variedad de sistemas de reconocimiento de voz.

2.1.4. Codificador.

Los codificadores nos permiten “compactar” la información, generando un código de salida a partir de la información de entrada.

También entendemos como un codificador (códec) a un esquema que modula una serie de transformaciones de una señal o información. Estos pueden convertir la señal en un formato encriptado para transmisión o encriptación, u obtener una señal de un formato encriptado adecuado para ver o editar (no necesariamente en su forma original). (Gonzales, 2010)

En este caso, los codificadores se utilizan en archivos multimedia para comprimir audio, imágenes o video porque la forma original de dichos archivos es demasiado grande para ser procesada y transmitida por los sistemas de comunicación

actualmente disponibles. También se utilizan para la compresión de datos para archivos de menor tamaño.

Según esta nueva definición, podemos clasificar los codificadores en códecs sin pérdidas y códecs con pérdidas, dependiendo de si la información recuperada coincide exactamente o se aproxima a la original.

2.1.5. Decodificador.

Un decodificador es un circuito integrado por el que se introduce un número y se activa una y sólo una de las salidas, permaneciendo el resto desactivadas. Por lo general, estos circuitos se utilizan normalmente como decodificadores/desmultiplexores. Esto se debe a que un demultiplexor puede funcionar como un decodificador. (Gonzales, 2010)

Por ejemplo, si se tiene un decodificador de 2 entradas con $2^2=4$ salidas, su funcionamiento será como se muestra en la siguiente tabla, donde se considera la lógica "1" para activar las salidas.

Entradas		Salidas			
A1	A0	D3	D2	D1	D0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Tabla 1: Tabla de verdad de 2 a 4 líneas

Decodificador. (2021). [Tabla]. Recuperado de
Fuente: <https://areatecnologia.com/electronica/codificador-decodificador.html>

Un tipo de decodificador ampliamente utilizado es el decodificador de siete segmentos. Este circuito decodifica la información de entrada BCD en un código de siete segmentos adecuado para mostrarse en una pantalla de siete segmentos.

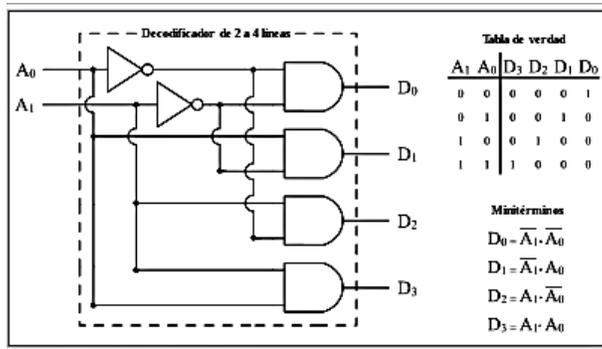


Figura 11: Decodificador de 2 a 4 líneas

Decodificador. (2021). [Tabla]. Recuperado de
 Fuente: <https://areatecnologia.com/electronica/codificador-decodificador.html>

2.1.5.1. Aplicaciones del Decodificador.

Su función principal es direccionar el espacio de memoria. El decodificador de entrada N puede direccionar 2^N espacios de memoria.

Para direccionar 1 Kib de memoria, se necesitan 10 bits porque el número de salidas es 2¹⁰, lo que equivale a 1024.

Por lo tanto:

- Para 20 bits, tienes 2²⁰ bits, lo que equivale a 1Mib.
- Para 30 bits, tienes 2³⁰ bits, lo que equivale a 1 Gib.

2.1.6. Filtros.

Un filtro electrónico es un sistema que tiene como función manipular y modificar el espectro de frecuencia de la señal de entrada para obtener en la salida la función que se requiera aplicar a los diferentes sistemas. (Torres, 2005)

Entre sus diversas aplicaciones podemos mencionar:

- Activos o pasivos.

- Analógicos o digitales.
- De paso alto (HPF), de paso bajo (LPF), de paso banda (BPF), elimina banda (filtros notch o filtros trampa) o "*pasa-todo*".
- De tiempos discretos o continuos.
- IIR (Respuesta infinita al impulso) o FIR (Respuesta finita al impulso).

Los filtros electrónicos más comunes son los filtros lineales.

2.1.6.1. Tipos de filtros.

2.1.6.1.1. Filtro Capacitivo.

Los filtros capacitivos son aquellos circuitos que utilizan capacitores, inductores y resistencias, se les conoce como filtros pasivos, la ventaja de estos filtros es su baja sensibilidad y su poca disipación de energía.

Utiliza circuitos electrónicos y microprocesadores para detectar perturbaciones dañinas de la red, como armónicos y picos de tensión, y una vez detectadas, son desviadas inmediatamente (en milisegundos) gracias al uso de semiconductores de última generación. (Torres, 2005)

La instalación es muy sencilla, en paralelo al aporte energético del concesionario, evitando que anomalías en la red afecten a los equipos eléctricos conectados a la red. Nuestro equipo de ingenieros e instaladores diseña e instala según sus necesidades y entrega filtros en funcionamiento. También puede realizar análisis y seguimiento del rendimiento.

El filtro capacitivo protege su instalación, evitando perturbaciones eléctricas, como variaciones de voltaje y picos de corriente por quemar el equipo instalado hasta 680V.

2.1.6.1.2. Filtros Activos.

Cumplen el mismo propósito que los filtros pasivos, pero en su señal de salida pueden presentar toda o parte de la señal de entrada. Para lograrlo se combinan componentes activos y pasivos. El uso de amplificadores operacionales es común para lograr resonancia y alto factor Q sin el uso de bobinas. Por lo tanto, al usar amplificadores, tienen más potencia y son relativamente fáciles de sintonizar.

2.1.6.1.3. Analógicos y Digitales.

2.1.6.1.3.1. Filtros Analógicos.

Filtro clásico. Diseñado para manejar señales analógicas. Diseñado con componentes analógicos como resistencias, condensadores y amplificadores operacionales.

2.1.6.1.3.2. Filtros Digitales.

Diseñado para manejar señales digitales. El procesamiento de señales digitales permite construir económicamente varios filtros. La señal analógica se captura y se convierte en una señal digital que consta de una serie de números a través de la conversión de analógico a digital. Un programa de computadora en una CPU o DSP dedicado calcula una serie de números de salida. Esta salida ahora se puede convertir en una señal a través de un proceso de conversión de digital a analógico. A veces, el "ruido" de la conversión puede ser problemático, pero este ruido se puede controlar y limitar con varios filtros. La frecuencia de la señal de entrada debe ser limitada, de lo contrario se producirán efectos de aliasing.

Hoy en día, la mayoría de los filtros son digitales porque los sistemas digitales tienen muchos beneficios sobre los sistemas analógicos: repetibilidad, estabilidad, capacidad del programador para redefinir, tamaño, etc.

2.1.6.1.4. Según la respuesta de Frecuencia.

Teóricamente, los filtros se caracterizan por su función de transferencia, por lo que cualquier configuración de componentes activos o pasivos que logren una determinada función de transferencia se consideraría un filtro de algún tipo.

Los inductores bloquean las señales de alta frecuencia y conducen las señales de baja frecuencia, mientras que los condensadores hacen lo contrario. Un filtro que transfiere corriente a tierra a través de un inductor o capacitor atenuará las señales de baja frecuencia menos que las señales de alta frecuencia, por lo que es un paso de baja frecuencia. Si la señal se conecta a tierra a través de un capacitor o un inductor, el filtro atenuará las señales de alta frecuencia menos que las señales de baja frecuencia, lo que lo convierte en un filtro de paso alto. Los resistores en sí mismos no tienen propiedades selectivas de frecuencia, pero están conectados a capacitores e inductores para determinar la constante de tiempo del circuito y, por lo tanto, la frecuencia a la que responde el circuito.

2.1.6.1.4.1. Filtro de paso bajo (LPF).

Se caracteriza por dejar pasar las frecuencias más bajas y atenuar las frecuencias más altas. Este tipo de filtro requiere dos terminales de entrada y dos de salida y una caja negra, también conocida como puerto dual o cuadropolo. De esta forma, la entrada recoge todas las frecuencias, y en la señal de salida sólo se encuentran las frecuencias permitidas por el filtro.

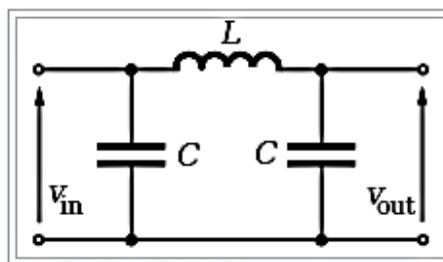


Figura 12: Filtro de paso bajo

Fuente: Peña J.C. (2013). Filtros Activos [Figura]. Recuperado de <https://www2.ulpgc.es/hege/almacen/download/29/29861/filtros.pdf>

2.1.6.1.4.2. Filtro de paso alto (HPF).

Consiste en una resistencia y un capacitor en serie, por lo que el capacitor solo permite que pasen frecuencias por encima de cierta frecuencia (llamada frecuencia de corte (F_c)) y atenúa las frecuencias por debajo de eso.

Su aplicación más común es eliminar o reducir información no deseada en el espectro de audio por debajo de 40-70 hercios (Hz). Es decir, está eliminando señales por debajo de las frecuencias audibles para proteger los altavoces y evitar la pérdida de potencia del amplificador.

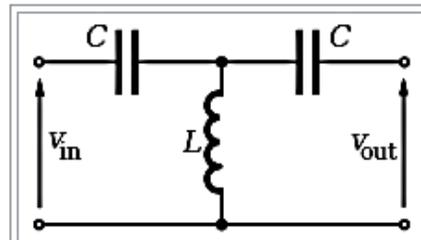


Figura 13: Filtro de paso alto

Fuente: Peña J.C. (2013). Filtros Activos [Figura]. Recuperado de <https://www2.ulpgc.es/hege/almacen/download/29/29861/filtros.pdf>

2.1.6.1.4.3. Filtro de paso banda (BPF).

Permite el paso de frecuencias contenidas en un determinado rango de frecuencias, es decir, entre la frecuencia de corte superior (FH) y la frecuencia de corte inferior (FL).

Se utiliza en ecualizadores de audio, lo que hace que algunas frecuencias se amplifiquen más que otras. También elimina los sonidos que parecen rodear las señales conocidas. Además de la electrónica y el procesamiento de señales, se utilizan en la ciencia atmosférica para procesar datos en el rango de 3 a 10 días.

2.1.6.1.4.4. Filtro de banda eliminada.

Es lo opuesto a un filtro de paso de banda, es decir, dificulta el paso de frecuencias en un rango de frecuencia determinado.

Son útiles para eliminar la interferencia de frecuencias conocidas, como las de redes electrónicas en equipos fijos en alrededor de 50 Hz.

2.1.6.1.4.5. Filtro multibanda.

Presenta diferentes rangos de frecuencia para los cuales se comporta de manera diferente.

2.1.6.1.4.6. Filtro variable.

Puede cambiar el margen de frecuencia según la función que desee realizar.

Para frecuencias altas (por encima de 100 MHz), los inductores a veces adoptan la forma de anillos o tiras de chapa, mientras que algunos condensadores consisten en tiras de metal adyacentes.

2.2. Fundamento Teórico Aplicativo.

2.2.1. Arduino.

Se trata de uno de los tipos de las placas más populares del mundo.

2.2.1.1. Qué es Arduino.



Figura 14: Placa Arduino

Fernández Y. (2022). Qué es Arduino, cómo funciona y qué puedes hacer con uno. [Figura]. Recuperado de Fuente: <https://www.xataka.com/basics/que-arduino-como funciona-que-puedes-hacer-uno>

Arduino es una plataforma de desarrollo basada en una placa electrónica de hardware libre que incorpora un microcontrolador reprogramable y una serie de pines hembra. Estos permiten establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores de una manera muy sencilla (principalmente con cables dupont). (MCI Electronics, 2022)

2.2.1.2. Funcionamiento de Arduino.

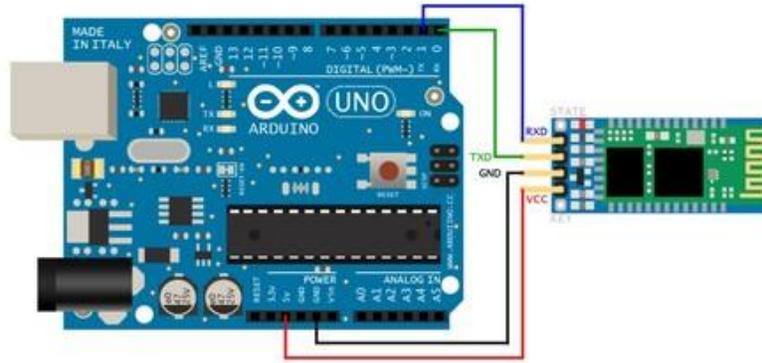


Figura 15: Placa Arduino conectando al Modulo Bluetooth

Fuente: Fernandez Y. (2022). Qué es Arduino, cómo funciona y qué puedes hacer con uno. [Figura]. Recuperado de <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>

Arduino es una placa de circuito basada en el microcontrolador ATMEL. Los microcontroladores son circuitos integrados que pueden grabar instrucciones, que puedes escribir en un lenguaje de programación que se puede usar en Arduino IDE (Entorno de desarrollo integrado). Estas instrucciones le permiten crear programas que interactúan con los circuitos de la placa.

Los microcontroladores Arduino tienen una interfaz de entrada, que es una conexión donde podemos conectar diferentes tipos de periféricos a la placa. La información de estos periféricos que conectes se transmitirá al microcontrolador, que será el encargado de procesar los datos que lleguen a través de ellos.

También cuenta con una interfaz de salida, que se encarga de transmitir la información ya procesada en el Arduino a otros periféricos. Estos periféricos pueden ser pantallas o altavoces en los que se reproducen los datos procesados, pero también pueden ser otras placas o controladores.

Partes que componen un Arduino (Autodesk, Inc., 2022)

- Potencia - USB (1) / Conector de Adaptador (2) ...
- Pines (5V, 3.3V, GND, Analog, Digital, PWM, AREF) ...
- Botón de reinicio (10) ...
- Indicador LED de alimentación (11) ...
- LEDs RX TX (12) ...
- Microcontrolador (13) ...
- Regulador de Voltaje (14)

2.2.1.3. Características Técnicas (Autodesk, Inc., 2022)

- Microcontrolador: Atmega328
- Voltaje de operación: 5V
- Voltaje de entrada (Recomendado): 7 – 12V
- Voltaje de entrada (Límite): 6 – 20V
- Pines para entrada- salida digital: 14 (6 pueden usarse como salida de PWM)
- Pines de entrada analógica: 6
- Corriente continua por pin IO: 40 mA
- Corriente continua en el pin 3.3V: 50 mA
- Memoria Flash: 32 KB (0,5 KB ocupados por el bootloader)
- SRAM: 2 KB
- EEPROM: 1 KB
- Frecuencia de reloj: 16 MHz

2.2.1.4. Tipos de Arduino. (Autodesk, Inc., 2022)

Arduino UNO:

Arduino de gama básica, todas las shields están diseñadas para usarse sobre esta placa. Cuenta 14 pines entrada/salidas digitales de las cuales 6 se pueden usar

como PWM, además cuenta con 6 entradas analógicas, además cuenta con I2C, SPI, además de un módulo UART.



Figura 16: Placa Arduino Uno

Fuente: Tipos de Arduino. [Figura]. Recuperado de <https://www.bolanosdj.com.ar/MOVIL/ARDUINO2/tiposarduino.pdf>

Arduino DUE:

El Arduino se basa en un microcontrolador de 32 bits, que tiene 54 entradas/salidas digitales y 12 entradas analógicas, 2 buses TWI, SPI y 4 UART. Todos los módulos basados en 3.3V funcionarán, no es compatible con 5V ya que dañará la placa. También tiene dos puertos USB internos para controlar periféricos.



Figura 17: Placa Arduino Due

Fuente: Tipos de Arduino. [Figura]. Recuperado de <https://www.bolanosdj.com.ar/MOVIL/ARDUINO2/tiposarduino.pdf>

Arduino Leonardo:

Arduino básico, con funcionalidad similar a Arduino, pero con 12 entradas analógicas y 20 entradas y salidas digitales. A diferencia de otros Arduinos con

microcontroladores ATmega32u4, no tiene controladores adicionales para controlar USB. Además, tiene más pines de interrupción externos. Dispone de comunicación TWI, SPI y dos UARTs.



Figura 18: Placa Arduino Leonardo

Fuente: Tipos de Arduino. [Figura]. Recuperado de <https://www.bolanosdj.com.ar/MOVIL/ARDUINO2/tiposarduino.pdf>

Arduino Mega 2560:

Arduino se basa en el microcontrolador ATmega2560. Tiene 54 entradas/salidas digitales, 16 de las cuales pueden usarse como PWM, 16 entradas analógicas y 4 UART, más dos modos PWI y un SPI. Tiene 6 interrupciones externas. Es compatible con todos los escudos de Arduino.



Figura 19: Placa Arduino Mega 2560

Fuente: Tipos de Arduino. [Figura]. Recuperado de <https://www.bolanosdj.com.ar/MOVIL/ARDUINO2/tiposarduino.pdf>

Arduino Mega ADK:

Exactamente igual que la Mega 2560, pero con la diferencia de que en este caso puede haber un host USB, que no es muy útil en este proyecto.

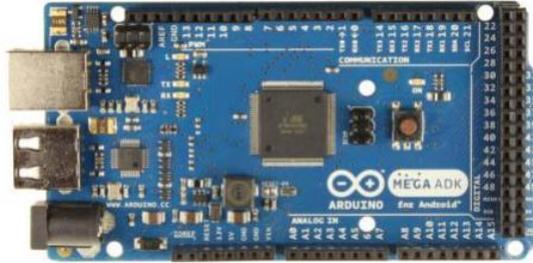


Figura 20: Placa Arduino Mega ADK

Fuente: Tipos de Arduino. [Figura]. Recuperado de <https://www.bolanosdj.com.ar/MOVIL/ARDUINO2/tiposarduino.pdf>

Arduino Micro:

Es exactamente similar al Leonardo, la única diferencia es su tamaño de construcción.

Es compatible con Arduino Shields, pero hay que montarlo de forma externa, es decir, mediante cableado, aunque si se construye nuestro propio shield no hay problema.



Figura 21: Placa Arduino Micro

Fuente: Tipos de Arduino. [Figura]. Recuperado de <https://www.bolanosdj.com.ar/MOVIL/ARDUINO2/tiposarduino.pdf>

Arduino Nano:

Arduino se basa en el microcontrolador Atmega328.

Es similar al Arduino Uno en términos de funcionalidad. La diferencia es el tamaño y la forma en que se conecta a la computadora para programarlo. Es compatible con la mayoría de los escudos, aunque de la misma manera que Arduino Micro.

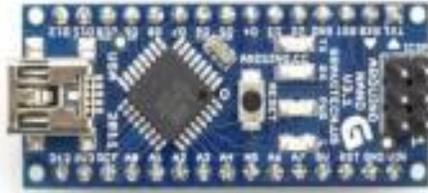


Figura 22: Placa Arduino Nano

Fuente: Tipos de Arduino. [Figura]. Recuperado de <https://www.bolanosdj.com.ar/MOVIL/ARDUINO2/tiposarduino.pdf>

Arduino YUN:

El Arduino YUN es un kit que funciona individualmente de forma complementaria, y, por un lado, tiene la versatilidad de un Arduino normal. En este caso, un ATmega 32u48 de 16 Mhz, por otro lado, un dispositivo con un microprocesador Atheros AR9331. Funciona en 400 Mhz Lilino (Linux basado en OperWrt (OperWrt-Yun)). Arduino funciona como una placa Leonardo. Contiene Ethernet, ranura SD y Wifi y está controlado por Lilino. Es compatible con todos los Shields y funciona de forma independiente.



Figura 23: Placa Arduino Yun

Fuente: Tipos de Arduino. [Figura]. Recuperado de <https://www.bolanosdj.com.ar/MOVIL/ARDUINO2/tiposarduino.pdf>

Arduino FIO:

Arduino se basa en el microcontrolador Atmega328p. Funciona a 8 Mhz y 3,3 V, dispone de 14 pines de entrada/salida digital (6 PWM), 8 pines de entrada analógica, e integra un conector de batería y módulo de carga correspondiente, así como un módulo de comunicación para montaje slot.xBee. Dispone de UART

TTL e interrupciones, lo que también nos permite ponerlo en modo reposo, también nos permite poner el XBEE en modo reposo, reduciendo así el consumo total. Además, posee tanto TWI (I2C) como SPI. Unas ventajas importantes de este dispositivo son el bajo consumo en Sleep y el poder programarlo mediante XBEE, sin necesidad conectarlo físicamente al ordenador.

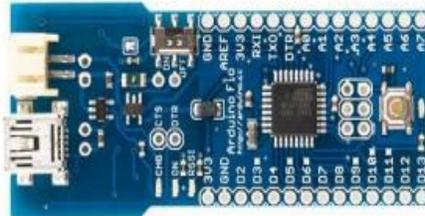


Figura 24: Placa Arduino Fio

Fuente: Tipos de Arduino. [Figura]. Recuperado de <https://www.bolanosdj.com.ar/MOVIL/ARDUINO2/tiposarduino.pdf>

2.2.2. Lenguaje de programación C++

C++ es un lenguaje de programación que proviene de la extensión del lenguaje C para que pudiese manipular objetos. A pesar de ser un lenguaje con muchos años, su gran potencia lo convierte en uno de los lenguajes de programación más demandados.

Fue diseñado a mediados de los años 80 por el danés Bjarne Stroustrup. Si intención fue la de extender el lenguaje de programación C (con mucho éxito en ese momento) para que tuviese los mecanismos necesarios para manipular objetos. Por lo tanto, C++ contiene los paradigmas de la programación estructurada y orientada a objetos, por lo que se le conoce como un lenguaje de programación multiparadigma.

A C++ primero se le conoció como “C con clases”. Luego se cambió a C++ que significa “incremento de C”, dando a entender que se trata de una extensión del lenguaje de programación C. (Robledano, 2019).

Las principales ventajas de programar en C++ son:

- **Alto rendimiento:** Es una de sus principales características, el alto rendimiento que ofrece. Esto es debido a que puede hacer llamadas directas al sistema operativo, es un lenguaje compilado para cada plataforma, posee gran variedad de parámetros de optimización y se integra de forma directa con el lenguaje ensamblador.
- **Lenguaje actualizado:** A pesar de que ya tiene muchos años, el lenguaje se ha ido actualizando, permitiendo crear, relacionar y operar con datos complejos y ha implementado múltiples patrones de diseño.
- **Multiplataforma**
- **Extendido:** C y C++ están muy extendidos. Casi cualquier programa o sistema están escritos o tienen alguna parte escrita en estos lenguajes (*desde un navegador web hasta el propio sistema operativo*).

2.2.2.1. Características del lenguaje C++

Algunas de las características más importantes que posee el lenguaje C++ son:

Compatibilidad con bibliotecas: A través de bibliotecas hay muchas funciones que están disponible y que ayudan a escribir código rápidamente.

Orientado a Objetos: El foco de la programación está en los objetos y la manipulación y configuración de sus distintos parámetros o propiedades.

Rapidez: La compilación y ejecución de un programa en C++ es mucho más rápida que en la mayoría de los lenguajes de programación.

Compilación: En C++ es necesario compilar el código de bajo nivel antes de ejecutarse, algo que no ocurre en otros lenguajes.

Punteros: Los punteros del lenguaje C, también están disponibles en C++.

Didáctico: Aprendiendo programación en C++ luego es mucho más fácil aprender lenguajes como Java, C#, PHP, Javascript, etc. (Robledano, 2019).

2.2.3. Módulo Bluetooth.

2.2.3.1. Protocolo Bluetooth.

Bluetooth es un protocolo de comunicaciones inalámbrico de corto alcance y bajo consumo de potencia en la banda ICM de 2,4 GHz que soporta tanto tráfico de datos como de audio. Su enlace es tan altamente confiable que hace de la tecnología una de las más aptas para cualquier tipo de aplicación en comunicaciones digitales, ya que habilita mecanismos de detección de error, ofrece una inmunidad natural a la interferencia empleando espectro disperso de salto de frecuencia FHSS a 1600 saltos por segundo y habilita procesos de encriptación para garantizar comunicaciones confiables y seguras. (Linares, Quijano, & Holguín , 2004)



Figura 25: Equipos con conexión Bluetooth

Fuente: Gonzales W. (2020). Qué es la tecnología Bluetooth y cómo funciona. [Figura]. Recuperado de <https://www.mundodeportivo.com/urbantecno/tecnologia/que-es-la-tecnologia-bluetooth-y-como-funciona>

Para emparejar o vincular un dispositivo con Bluetooth, nos referiremos a dos términos: maestro-esclavo, o maestro-esclavo en inglés.

El dispositivo que inicia el emparejamiento se denomina maestro y puede vincularse a varios dispositivos (hasta 7) que serán esclavos, pero estos últimos solo pueden conectarse a un maestro.



Figura 26: Equipo conectándose a otros dispositivos vía Bluetooth

Fuente: Gonzales W. (2020). Qué es la tecnología Bluetooth y cómo funciona. [Figura]. Recuperado de <https://www.mundodeportivo.com/urbantecno/tecnologia/que-es-la-tecnologia-bluetooth-y-como-funciona>

2.2.3.2. Modulo Bluetooth HC-05.

El módulo bluetooth HC-05 nos permite conectar de manera inalámbrica nuestros proyectos Arduino a un smartphone, celular o PC (Bluetooth) y fácil acceso al puerto serial. La transmisión es completamente transparente para el programador, por lo que se conecta directamente a los pines seriales de nuestro microcontrolador preferido (teniendo en cuenta los niveles de voltaje, ya que el módulo se alimenta con 3.3V). Todos los parámetros del módulo se pueden configurar mediante comandos AT. La placa también incluye un regulador de 3,3 V, que permite alimentar el módulo entre 3,6 V y 6 V. Este módulo es un complemento ideal para nuestros proyectos de robótica, domótica y control remoto, incluidos Arduino, PIC, Raspberry PI, ESP8266, ESP32, STM32. (Naylamp Mechatronics SAC, 2022)

La comunicación Bluetooth se produce entre dos tipos de dispositivos: maestro y esclavo. Si nuestro objetivo es conectar nuestro proyecto a un smartphone Android, podemos utilizar el módulo HC-06 o HC-05 configurado como esclavo. El módulo Bluetooth HC-05 viene configurado como esclavo, es decir, listo para escuchar solicitudes de conexión, pero podemos configurarlo para que funcione con un maestro mediante comandos AT. Por otro lado, si nuestro objetivo es

conectar dos proyectos, necesitaremos utilizar un módulo HC-05 configurado como maestro y un HC-06 (esclavo) o HC-05 configurado como esclavo.

Este módulo cumple con la especificación estándar de Bluetooth 2.0 y es perfectamente compatible con teléfonos o teléfonos inteligentes con Android, pero no con iPhone. Para usar con iPhone, recomendamos el módulo Bluetooth 4.0 BLE HM-10, que también es compatible con los teléfonos Android modernos.



Figura 27: Modulo Bluetooth HC-05

Fuente: Lozano R. (2018). modulo Bluetooth hc-05. [Figura]. Recuperado de <https://www.taloselectronics.com/blogs/tutoriales/configurar-modulo-bluetooth-hc-5>

2.2.3.3. Especificaciones Técnicas:

- Voltaje de operación: 3.6V - 6V DC
- Consumo corriente: 50mA
- Bluetooth: V2.0+EDR
- Frecuencia: Banda ISM 2.4GHz
- Modulación: GFSK(Gaussian Frequency Shift Keying)
- Potencia de transmisión: 4dBm, Class 2
- Sensibilidad: -84dBm a 0.1% BER
- Alcance 10 metros
- Interface comunicación: Serial TTL
- Velocidad de transmisión: 1200bps hasta 1.3Mbps

- Baudrate por defecto: 38400,8,1,n.
- Seguridad: Autenticación y encriptación
- Temperatura de trabajo: -20C a +75C
- Compatible con Android
- Dimensiones: 37*16 mm
- Peso: 3.6 gramos

2.2.3.4. Conexión.

Puede conectar el módulo directamente al Arduino, la secuencia de conexión es la siguiente.

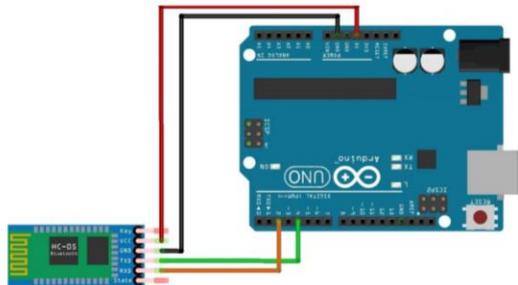


Figura 28: Modulo Bluetooth conectando a la Placa Arduino

Fuente: Lozano R. (2018). modulo Bluetooth hc-05. [Figura]. Recuperado de <https://www.taloselectronics.com/blogs/tutoriales/configurar-modulo-bluetooth-hc-5>

Como se puede ver, los pines Vcc y GND son los mismos de siempre. En cambio, los otros dos pines están cruzados. Conecte el pin RXD de HC-05 al pin 2 (TX) de Arduino y TXD al pin 4 (RX) de Arduino.

2.2.3.5. Vincular.

Vincular un módulo bluetooth a un teléfono no es diferente a lo que se está acostumbrado.

Se puede conectar los pines Vcc y GND del módulo HC y continuar.

Para conectar el módulo bluetooth al teléfono, primero se necesita saber el nombre del módulo y su contraseña o PIN. Si el módulo es nuevo, lo más probable es que el nombre sea HC-05 o Linvor, en caso de que este no sea el nombre, podemos usar comandos AT para cambiar el nombre y el PIN.

2.2.3.6. Comandos AT.

El comando AT (atención) se utiliza para configurar el módem. La tecnología ha evolucionado con el tiempo, pero estos comandos siguen siendo estándar, por ejemplo, en los puertos serie.

Con estos comandos podemos configurar el módulo, podemos ver y cambiar el nombre, la velocidad de transferencia, la contraseña y muchos otros aspectos. Pero por ahora nos centraremos en lo básico.

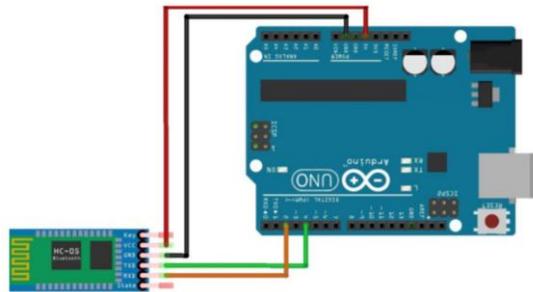


Figura 29: Modulo Bluetooth conectando a la Placa Arduino

Fuente: Lozano R. (2018). modulo Bluetooth hc-05. [Figura]. Recuperado de <https://www.taloselectronics.com/blogs/tutoriales/configurar-modulo-bluetooth-hc-5>

2.2.3.7. Conectando el dispositivo Bluetooth con Arduino.

Se debe ingresar en la ventana de configuración de Bluetooth del teléfono y buscar la ventana que muestra el nombre del modelo del sensor que se está utilizando ("linvor" en algunos casos). Se pedirá una contraseña, la cual es 1234.



Figura 30: muestra el Bluetooth activado y reconoce el módulo Bluetooth HC-05

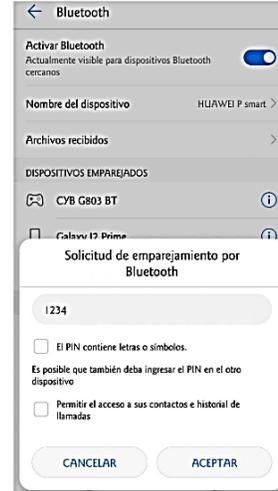


Figura 31: Introduciendo la contraseña

Fuente: Lozano R. (2018). modulo Bluetooth hc-05. [Figura]. Recuperado de <https://www.taloselectronics.com/blogs/tutoriales/configurar-modulo-bluetooth-hc-5>

2.2.4. Aplicación App Inventor.

Con la aplicación App Inventor, puede crear aplicaciones de Android sin código. Este servicio en línea utiliza un método de arrastrar y soltar para insertar y organizar los elementos que desea mostrar en su aplicación.

Lo mismo ocurre al programar, solo necesita seleccionar la parte del bloque donde se encuentra la instrucción. Se puede utilizar en varios idiomas incluido el español.



Figura 32: Logotipo de la aplicación Mit App Inventor

Fuente: Posada F. (2015). Creando aplicaciones para móviles Android con MIT App Inventor. [Figura]. Recuperado de https://intef.es/observatorio_tecno/creando-aplicaciones-para-moviles-android-con-mit-app-inventor-2/

2.2.4.1. Modo de uso App Inventor.

Debe iniciar sesión con su cuenta de Gmail. Después de eso, verá un panel donde se mostrarán todos sus proyectos.



Figura 33: Inicio a la aplicación App Inventor

Fuente: Posada F. (2015). Creando aplicaciones para móviles Android con MIT App Inventor. [Figura]. Recuperado de https://intef.es/observatorio_tecno/creando-aplicaciones-para-moviles-android-con-mit-app-inventor-2/

Los proyectos se pueden importar a App Inventor, por ejemplo, esta es la aplicación diseñada para esta ocasión.

En App Inventor, en el menú Proyecto, luego importe el proyecto (.aia) desde su computadora.

Luego busque el archivo ai descargado y selecciónelo.

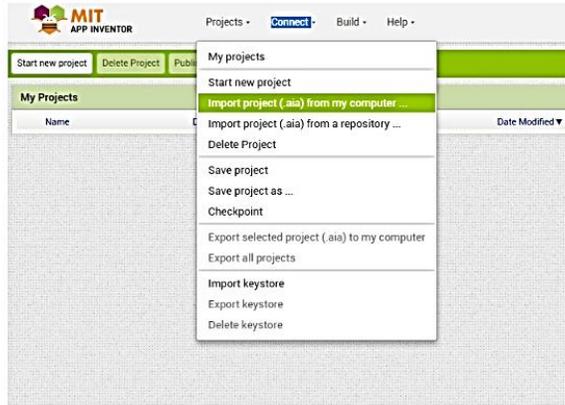


Figura 34: Buscando el archivo creado

Fuente: Posada F. (2015). Creando aplicaciones para móviles Android con MIT App Inventor. [Figura]. Recuperado de https://intef.es/observatorio_tecno/creando-aplicaciones-para-moviles-android-con-mit-app-inventor-2/

Se ha agregado a su panel de proyectos. Cuando lo abres, lo primero que notas es la pantalla principal de la aplicación.



Figura 35: Pantalla principal de la aplicación

Fuente: Posada F. (2015). Creando aplicaciones para móviles Android con MIT App Inventor. [Figura]. Recuperado de https://intef.es/observatorio_tecno/creando-aplicaciones-para-moviles-android-con-mit-app-inventor-2/

En el botón que muestra el bloque, se puede ver la ventana para programar la aplicación con el bloque.

La aplicación primero verifica si se ha presionado el selector de lista (conectar), si no, abrirá un menú donde debe elegir entre todos los dispositivos en la lista vinculada, donde debe estar el nombre del módulo.

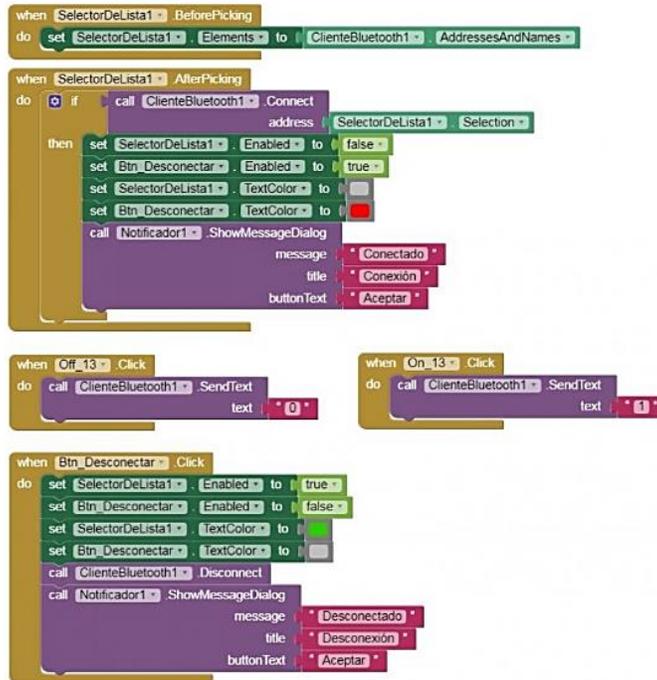


Figura 36: Código del proyecto creado

Fuente: Posada F. (2015). Creando aplicaciones para móviles Android con MIT App Inventor. [Figura]. Recuperado de https://intef.es/observatorio_tecno/creando-aplicaciones-para-moviles-android-con-mit-app-inventor-2/

Cuando abras la aplicación, verás un cuadro de texto donde tienes que escribir un código.



Figura 37: Cuadro de texto

Fuente: Posada F. (2015). Creando aplicaciones para móviles Android con MIT App Inventor. [Figura]. Recuperado de https://intef.es/observatorio_tecno/creando-aplicaciones-para-moviles-android-con-mit-app-inventor-2/

Para obtener este código, abra el menú Conectar en App Inventor, luego abra AI Companion.



Figura 38: Pantalla principal de la aplicación

Fuente: Posada F. (2015). Creando aplicaciones para móviles Android con MIT App Inventor. [Figura]. Recuperado de https://intef.es/observatorio_tecno/creando-aplicaciones-para-moviles-android-con-mit-app-inventor-2/

Después de eso, en la aplicación, presione el botón Conectar para seleccionar el HC-05 vinculado anteriormente y use los botones de encendido y apagado para controlar los LED.



Figura 39: Controlador del Led en la Aplicación

Fuente: Posada F. (2015). Creando aplicaciones para móviles Android con MIT App Inventor. [Figura]. Recuperado de https://intef.es/observatorio_tecno/creando-aplicaciones-para-moviles-android-con-mit-app-inventor-2/

2.2.5. Led infrarrojo:

Los Leds infrarrojos (IR) funcionan convirtiendo la corriente eléctrica en luz infrarroja, mientras que los detectores infrarrojos hacen lo contrario al detectar la luz infrarroja y convertirla en corriente eléctrica. La corriente eléctrica producida por el detector de infrarrojos es la señal de que esta luz está presente. El infrarrojo es una longitud de onda de luz que está más allá del alcance de la visión humana. Esto hace que el infrarrojo sea una excelente herramienta para aplicaciones que requieren luz, pero la luz visible puede distraer o ser innecesaria. El uso de diodos emisores de luz infrarroja o LED hace posibles los sistemas de control remoto en una variedad de proyectos. (Geekbot Electronics, 2021)

2.2.5.1. Led Emisor:

Los LED infrarrojos se utilizan en aplicaciones de control remoto, barreras fotoeléctricas, detección de objetos y más; los LED se empaquetan en 5 mm y funcionan a 940 nm.



Figura 40: Led Emisor

Fuente: Diodo infrarrojo. [Figura]. Recuperado de <https://avelectronics.cc/producto/diodo-infrarrojo-5mm/>

2.2.5.1.1. Características:

- Voltaje de operación: 1,7V
- Corriente máxima: 100mA
- Longitud de onda: 940nm

2.2.5.2. Led Receptor:

Un fotodiodo es un dispositivo que conduce una cantidad de corriente proporcional a la cantidad de luz que incide o incide sobre él, esto se suele utilizar junto con LED emisores de infrarrojos en aplicaciones como detectores de objetos, barreras fotoeléctricas, o controles infrarrojos.



Figura 41: Led Receptor

Fuente: Diodo infrarrojo. [Figura]. Recuperado de <https://avelectronics.cc/producto/diodo-infrarrojo-5mm/>

2.2.5.2.1. Características:

- Voltaje máximo inverso: 1,3V
- Angulo de visión: 20°
- Longitud de onda: 940nm

2.2.6. Transistor:

Un transistor es un dispositivo electrónico semiconductor que transmite una señal de salida en respuesta a una señal de entrada. Implementa las funciones de un amplificador, oscilador, conmutador o rectificador. El término “transistor” es un acrónimo de resistencia de transferencia (“resistencia de transferencia”). Actualmente está presente en casi todos los dispositivos electrónicos de uso cotidiano, como radios, televisores, reproductores de audio y video, relojes de cuarzo, computadoras, lámparas fluorescentes, tomógrafos, teléfonos celulares, aunque casi siempre en los llamados circuitos integrados.

2.2.6.1. Funcionamiento:

A diferencia de una válvula, un transistor es un dispositivo de control de corriente del que se extrae una corriente amplificada. En el diseño de circuitos, los transistores se consideran componentes activos, a diferencia de las resistencias, los condensadores y los inductores, que son componentes pasivos. 33

De forma simplificada, la corriente que circula por el colector es función amplificadora de la corriente inyectada en el emisor, pero si la base está alimentada por una fuente de corriente continua, el transistor solo regula la corriente que circula por sí mismo. La carga a través del colector depende del tipo de circuito utilizado. El factor de amplificación o ganancia logrado entre la corriente de colector y la corriente de base se denomina Beta del transistor. Otros parámetros para considerar y parámetros específicos para cada transistor incluyen: Emisor Colector, Emisor Base, Voltaje de Ruptura del Colector Base, Potencia Máxima, Disipación de Calor, Frecuencia de operación, y varias tablas donde se grafican diferentes parámetros como corriente base, voltaje colector emisor, voltaje base emisor, corriente emisor, etc. Los tres esquemas básicos (configuraciones) utilizados para la simulación de transistores son emisor común, colector y base comunes.

El modelo de transistor descrito más adelante, los transistores bipolares (transistores FET, MOSFET, JFET, CMOS, VMOS, etc.) no utilizan la corriente inyectada en el terminal base para modular la corriente del emisor o del colector, sino que utilizan el voltaje presente en el terminal de puerta y escale la conductancia del canal entre la fuente y los terminales de drenaje. Cuando la conductancia es cero y el canal está estrangulado, el campo eléctrico presente en el canal es responsable de conducir los electrones desde la fuente al drenaje debido al voltaje aplicado entre la puerta y la fuente. De esta manera, la corriente de salida en la carga se conecta al drenaje (D) será función amplificada de la tensión presente entre la compuerta y la fuente, de manera análoga al funcionamiento del triodo.

Los transistores de efecto de campo son la razón de la integración a gran escala en la actualidad. En términos generales, se pueden fabricar cientos de miles de transistores interconectados por centímetro cuadrado, en varias capas apiladas.

2.2.7. Televisor

2.2.7.1. Televisor LG.

El televisor LG 42LB6200 disfruta de acceso directo a tus películas 3D. La Serie LB6200 Cuenta con Tecnología 3D, Dual Play, 2 Entradas HDMI y Puerto USB.



Figura 42: Televisor LG modelo 42LB6200

Fuente: LG Cinema 3D. [Figura]. Recuperado de <https://www.lg.com/pe/tvs/lg-49LB6200>

2.2.7.1.1. Características.

Marca	LG
Tecnología	LED
Tamaño de pantalla	42"
Resolución	1920 x 1080 pixeles
Definición	Full HD
Dimensiones	570 mm x 959 mm x 56.2 mm
Peso	9.2 kg (con base 9.5 kg)
Curvo	No
3D	Si

HDMI	2
USB	USB 2.0 (1)

2.2.7.2. Televisor Samsung.

TV LED 32" UN32F4000AG HD. Con una resolución de 1366 x 768 y sintonizador ISDB-T DTV, tiene 2 entradas de HDMI y una de USB para que ver a tu contenido multimedia sea mucho más sencillo.



Figura 43: Televisor Samsung modelo UN32F4000AG

Fuente: TV Led 32". [Figura]. Recuperado de <https://www.falabella.com.ar/falabella-ar/product/1857410/TV-LED-32-UN32F4000AG-HD/1857410>

2.2.7.2.1. Características.

- Marca: Samsung
- Modelo: UN32F4000AG
- Resolución: 1366 x 768
- Salida de Sonido 10 W x 2 (RMS)
- Dolby Digital Plus / Dolby Pulse
- Apagado automático
- Sintonizador ISDB-T DTV

- Sintonizador Analógico
- Salida de Sonido (Mini Jack)
- Componente In (Y / Pb / Pr) :1
- Compuesto de entrada (AV) :1 (Uso común para Componente Y)
- Salida de Audio Digital (óptica): 1
- HDMI X 2
- USB X 1
- Peso sin soporte: 5.2 kg

CAPITULO 3

3.1. Desarrollo Practico Experimental.

Primeramente, se muestra el diagrama general de funcionamiento, en el cual se tiene como parte central un Arduino UNO el cual se conecta al módulo bluetooth para enviar y recibir información de una aplicación para sistema operativo Android.

Este también tiene un emisor infrarrojo para enviar información y para mandar distintos comandos al televisor.

Así mismo tiene un receptor infrarrojo para recibir la señal de controles remotos IR que usan muchos de los equipos domésticos como TVs, el cual a su vez se van mostrando vía bluetooth en la aplicación.

Este receptor tiene un filtro interno para detectar solo frecuencias infrarrojas cercanas a 32 o 38KHz, lo que lo hace compatible con la mayoría de los mandos infrarrojos, posee 3 pines de conexión GND, VCC y DATA, el cual nos permite conectar directamente a un pin digital de nuestro Arduino.

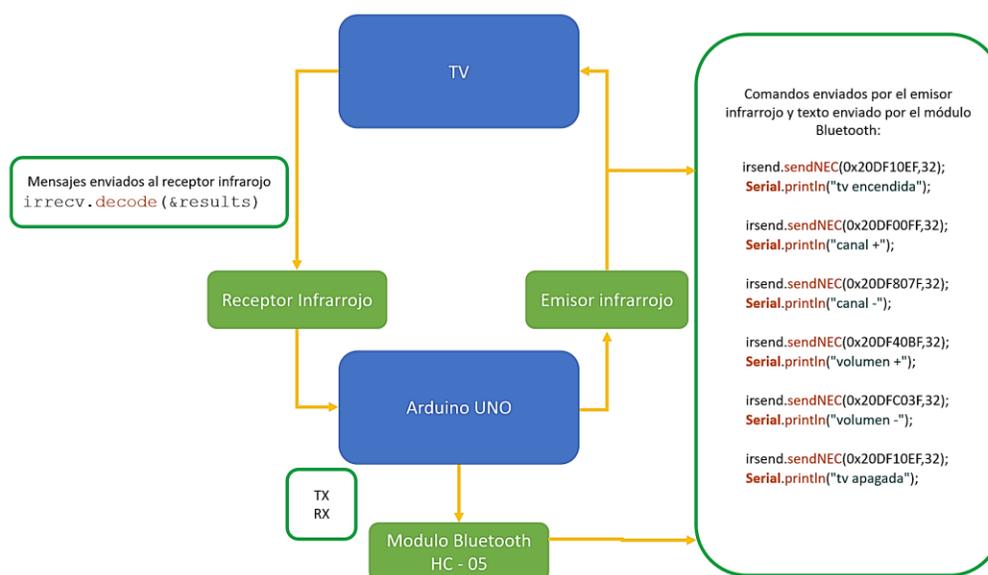


Figura 44: Diagrama general de funcionamiento

Fuente: Elaboración Propia

Para poder realizar el diseño de este proyecto presentaremos la lista de los componentes que se utilizaron en el mismo.

Lista de componentes:

- Placa Arduino Uno
- Módulo de Bluetooth
- Led Infrarrojo Emisor
- Led Infrarrojo Receptor VS1838B
- Transistor 2N 3904
- Dos resistencias de 1K Ω
- Dos resistencias de 220 Ω
- Placa virgen

Este proyecto se realizará con mandos de TV, en particular usaremos un control remoto de un TV LG, este mando usa el protocolo NEC que trabaja a 38KHz de frecuencia.



Figura 45: control de televisor LG

Fuente: LG Cinema 3D. [Figura]. Recuperado de <https://www.lg.com/pe/tvs/lg-49LB6200>

La dirección está asociada a un dispositivo TV, y el comando está asociado a la acción o función del botón. Por ejemplo, subir el volumen, apagar, subir canal, etc.

Para este tutorial vamos a trabajar como si se tratase de un solo bloque de datos de 32 bits.

A continuación, se muestra los valores de los datos correspondientes a los botones del control en mención:

BOTON	DATO 32 BITS (HEX)
Encender/Apagar	20DF10EF
Canal (+)	20DF00FF
Canal (-)	20DF807F
Volumen (+)	20DF40BF
Volumen (-)	20DFC03F

3.1.1. Arduino.

El Arduino elegido para este proyecto fue el “Arduino Uno” de gama básica, por ser la más comercial en nuestro país y a la vez por sus características con las cuales cuenta como ser:

- **Microcontrolador:** *ATmega328*
- **Voltaje Operativo:** *5v*
- **Voltaje de Entrada (Recomendado):** *7 – 12 v*
- **Pines de Entradas/Salidas Digital:** *14 (De las cuales 6 son salidas PWM)*
- **Pines de Entradas Análogas:** *6*
- **Memoria Flash:** *32 KB (ATmega328) de los cuales 0,5 KB es usado por Bootloader.*
- **SRAM:** *2 KB (ATmega328)*
- **EEPROM:** *1 KB (ATmega328)*
- **Velocidad del Reloj:** *16 MHZ.*

A su vez por su fácil manejo tiene todo lo necesario para manejar el controlador, simplemente conectamos al computador por medio del cable USB o una fuente de poder externa, que puede ser un adaptador AC-DC o una batería, cabe aclarar que si se alimenta a través del cable USB en el ordenador no es necesario una fuente externa.

3.1.2. Circuito simulado en ISIS.

En la siguiente figura podemos apreciar el circuito diseñado en la aplicación Proteus simulado en ISIS para poder ver e identificar como se realizó el diseño, conexión y los componentes utilizados.

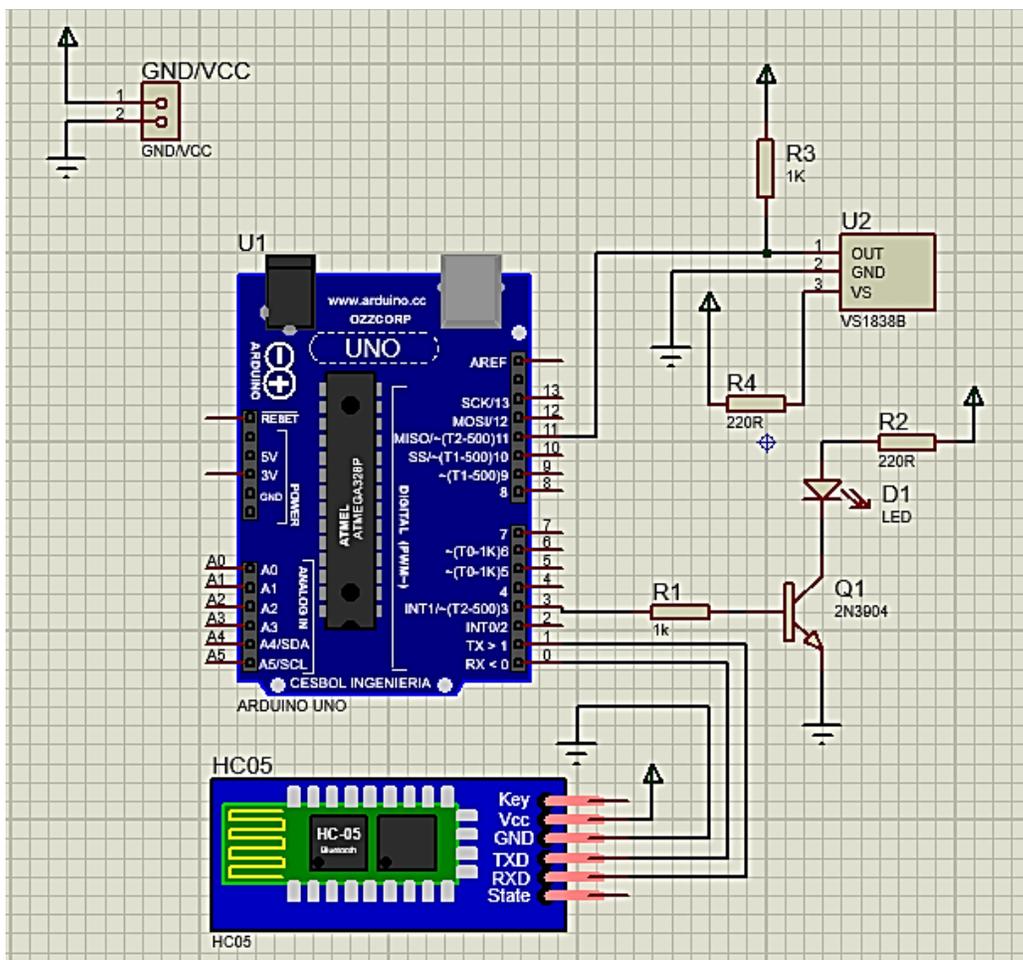


Figura 46: Circuito Diseñado y Simulado en ISIS

Fuente: Elaboración propia

- **Circuito en 3D.**

En la figura se puede observar como se ve el prototipo real del circuito diseñado.

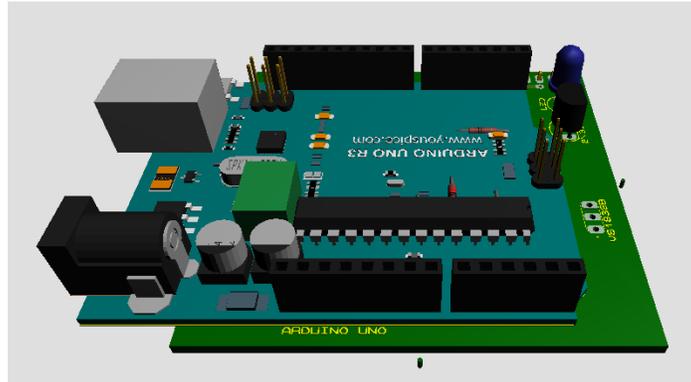


Figura 48: Circuito diseñado en 3D

Fuente: Elaboración propia

3.1.3. Programa del Arduino Uno.

Seguidamente se muestra el código del programa implementado en una placa basada en un microcontrolador como es el Arduino Uno, el cual fue elaborado en un lenguaje de programación que se puede utilizar en el entorno Arduino IDE, a su vez lo que permitió que el código funcionara correctamente fue la librería IRremote, es una de las librerías más usadas y completas para trabajar con protocolos de controles infrarrojos, tiene implementado varios protocolos de las marcas más conocidas como Sony, LG, Samsung, Sanyo, etc.

Código elaborado:

```
/*codigos tv lg
apagar:20DF10EF
canal +: 20DF00FF
canal -: 20DF807F
volumen +:20DF40BF
volumen -:20DFC03F
*/

#include "IRremote.h"
#include "IRremoteInt.h"
```

```

int RECV_PIN = 11;

IRrecv irrecv(RECV_PIN);

decode_results results;
IRsend irsend;
char x;
bool flag_error=false;
int sensor1 = 0; //conectado a la alimentacion de la placa
int sensor2 = 0; //conectado a la pata 2 del led emisor
int sensor3 = 0; //conectado a la pata 1 y el transistor 2n3904
int sensor4 = 0; //conectado a la señal del receptor infrarojo
void setup()
{
  Serial.begin(9600);
  delay(500);
  pinMode(13,1);
  irrecv.enableIRIn(); // Start the receiver
}
void loop()
{
  error_detection();

  digitalWrite(13,0);
  if (irrecv.decode(&results))
  {
    if((String)results.value != ""){
      Serial.println("Receptor error");
      delay(1000);
      Serial.println(" ");
    }

    irrecv.resume(); // Receive the next value
  }
  if(Serial.available())
  {
    x=Serial.read();
    switch(x)
    {
      case 'a':
        digitalWrite(13,1);
        irsend.sendNEC(0x20DF10EF,32);
        Serial.println("tv encendida");
        delay(1000);
        break;
      case 'b':
        digitalWrite(13,1);
        irsend.sendNEC(0x20DF00FF,32);
        Serial.println("canal +");
        delay(1000);
        break;
      case 'c':
        digitalWrite(13,1);
        irsend.sendNEC(0x20DF807F,32);
        Serial.println("canal -");
        delay(1000);
        break;
    }
  }
}

```

```

    case 'd':
        digitalWrite(13,1);
        irsend.sendNEC(0x20DF40BF,32);
        Serial.println("volumen +");
        delay(1000);
        break;
    case 'e':
        digitalWrite(13,1);
        irsend.sendNEC(0x20DFC03F,32);
        Serial.println("volumen -");
        delay(1000);
        break;
    case 'f':
        digitalWrite(13,1);
        irsend.sendNEC(0x20DF10EF,32);
        Serial.println("tv apagada");
        delay(1000);
        break;
}
}
}

void error_detection(){
    sensor1 = analogRead(A2); //conectado a la alimentacion de la placa
    sensor2 = analogRead(A3); // conectado a la pata 2 del led emisor
    sensor3 = analogRead(A4); // conectado a la pata 1 y el transistor
    sensor4 = analogRead(A5); // conectado a la señal del receptor
    infrarojo
    // Serial.print(sensor1);Serial.print(" ");
    // Serial.print(sensor2);Serial.print(" ");
    // Serial.print(sensor3);Serial.print(" ");
    // Serial.println (sensor4);

    if (sensor1<1010){
        Serial.println("bateria baja");
        delay(1000);
        Serial.println(" ");
    }
    else if (sensor3<400){
        Serial.println("led emisor desconectado");
        delay(1000);
        Serial.println(" ");
    }
    else if (sensor3>1020){
        Serial.println("receptor desconectado");
        delay(1000);
        Serial.println(" ");
    }
    else{
        flag_error=false;
    }
}
}

```

Finalmente, en la siguiente figura se muestra el prototipo del circuito finalizado en una placa diseñada, en la cual se puede observar el módulo bluetooth, los leds infrarrojos emisor y receptor, el transistor, las resistencias y el Arduino Uno que se encuentra conectado por debajo.



Figura 49: Prototipo del circuito finalizado

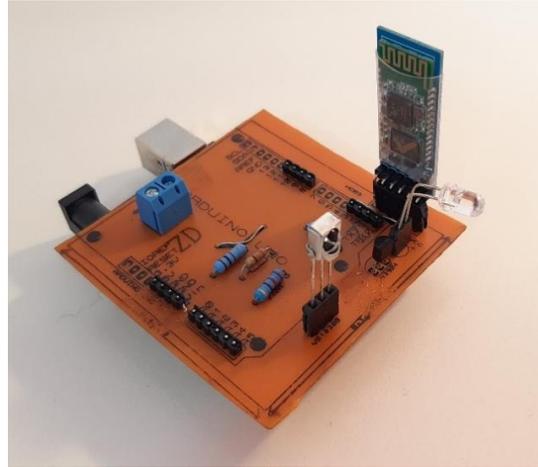


Figura 50: Prototipo del circuito finalizado

Fuente: Elaboración propia

Se observa en la siguiente figura las conexiones a los pines analógicos, que se van testeando cada 1 segundo en promedio aproximadamente, que es el tiempo de ejecución de cada ciclo de programa del código mostrado anteriormente.

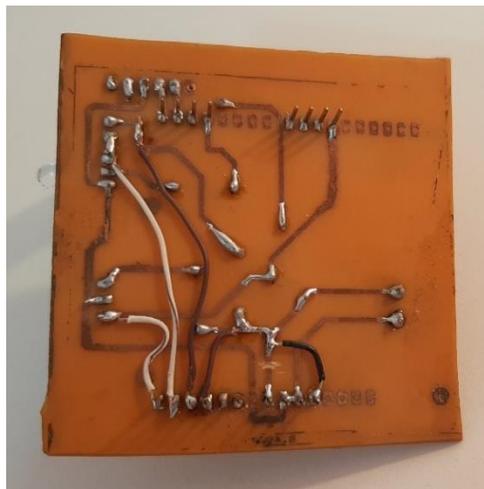


Figura 51: Placa quemada y finalizada (Parte de atrás)

Fuente: Elaboración propia

TP4 = Conectado a la señal del receptor infrarrojo.

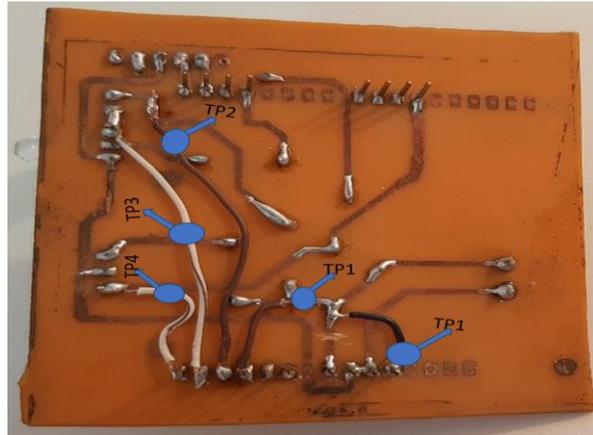


Figura 53: Placa diseñada y quemada (parte de atrás) mostrando los puntos TP (test point)

Fuente: Elaboración propia

Estos TP (test point) deben pasar los umbrales mostrados en el siguiente código que se ve a continuación en función al lenguaje C++

```
void error_detection(){
    sensor1 = analogRead(A2); //conectado a la alimentacion de la placa
    sensor2 = analogRead(A3); // concetado a la pata 2 del led emisor
    sensor3 = analogRead(A4); // concetado a la pata 1 y el transistor
    2n3904
    sensor4 = analogRead(A5); // conectado a la señal del receptor
    infrarojo
    // Serial.print(sensor1);Serial.print(" ");
    // Serial.print(sensor2);Serial.print(" ");
    // Serial.print(sensor3);Serial.print(" ");
    // Serial.println (sensor4);

    if (sensor1<1010){
        Serial.println("bateria baja");
        delay(1000);
        Serial.println(" ");
    }
    else if (sensor3<400){
        Serial.println("led emisor desconectado");
        delay(1000);
        Serial.println(" ");
    }
    else if (sensor3>1020){
        Serial.println("receptor desconectado");
        delay(1000);
        Serial.println(" ");
    }
    else{
```

```

    flag_error=false;
}
}

```

Los voltajes medidos a través de las entradas ADC se muestran en las siguientes figuras:

- a) **Señal normal:** Las medidas analógicas medidas cuando no existe ningún error se muestran en la siguiente figura.

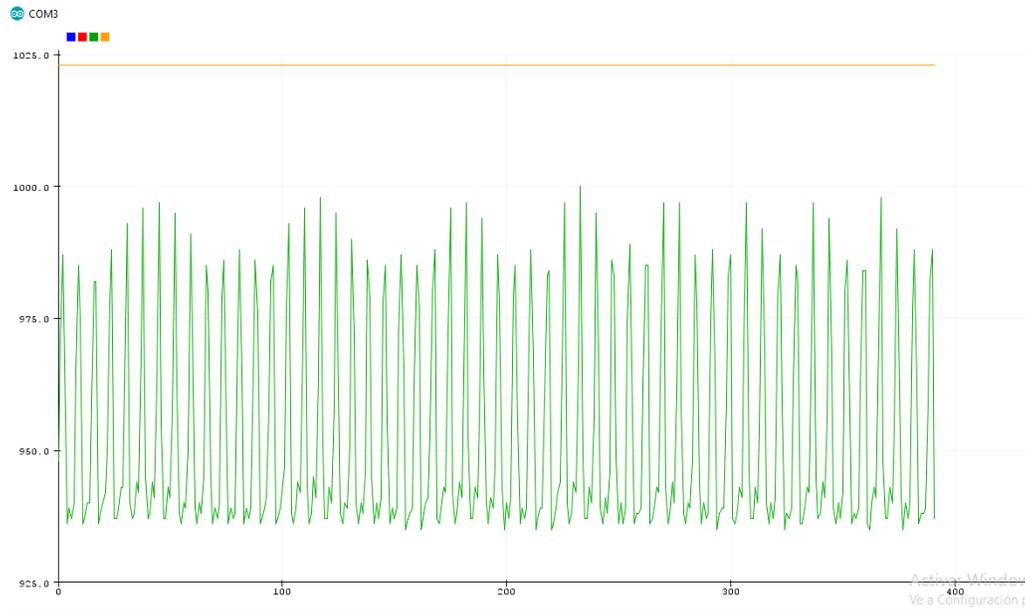


Figura 54: Medidas analógicas mostrando señal normal

Fuente: Elaboración propia

Donde:

- Azul sensor 1, conectado a la alimentación de la placa.
- Rojo Sensor 2, conectado a la pata 2 del led emisor.
- Verde Sensor 3, conectado a la pata 1 y el transistor 2n3904.
- Naranja Sensor 4, conectado a la señal del receptor infrarrojo.

Se observa que la mayoría de las señales tienden a estar en estado ALTO o en 5V que se traduce a una lectura de 1023 en un converso ADC de 10 bits. A excepción de la señal verde correspondiente al punto de conexión del transistor 2N3904 que tiene una oscilación mínima de entre 940 a 990 a través del conversor, esto debido a la librería utilizada.

Naturalmente el sensor 1, conectado a la alimentación bajará siempre y cuando se alimente con una fuente de tensión menor a 5V.

b) Señal falla en el receptor infrarrojo

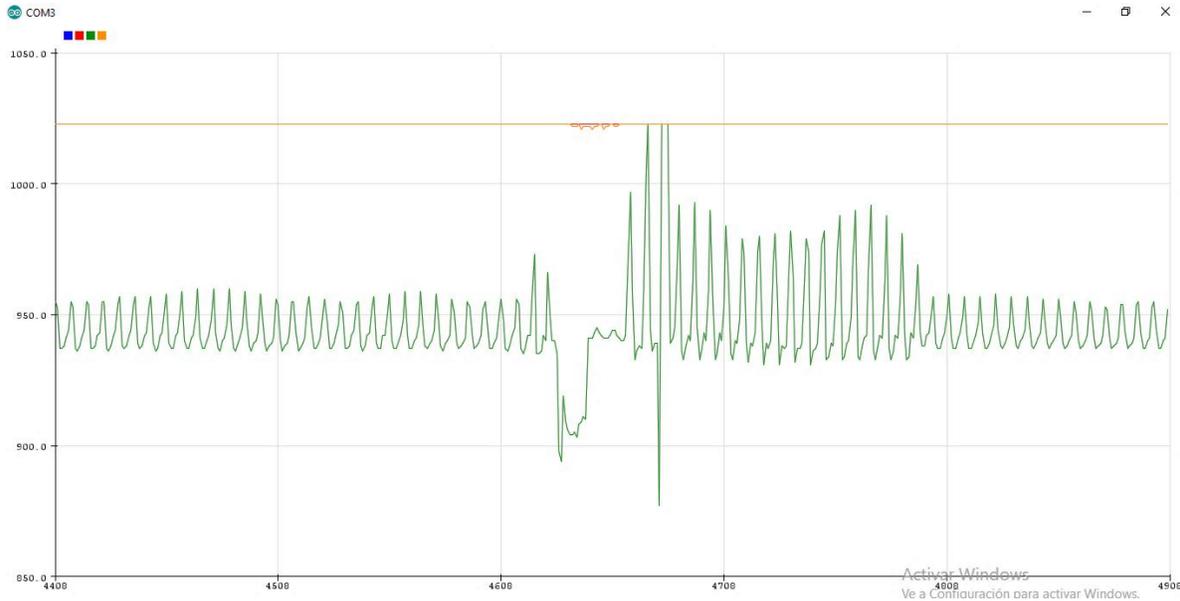


Figura 55: Medidas analógicas mostrando señal con falla en el receptor infrarrojo

Fuente: Elaboración propia

Donde:

- Azul sensor 1, conectado a la alimentación de la placa.
- Rojo Sensor 2, conectado a la pata 2 del led emisor.
- Verde Sensor 3, conectado a la pata 1 y el transistor 2n3904.
- Naranja Sensor 4, conectado a la señal del receptor infrarrojo.

Cuando existe una falla en el receptor infrarrojo como una desconexión se registran dos oscilaciones en el sensor 3 y 4, se observa una oscilación grande en el sensor 3 y una oscilación muy pequeña y característica en el sensor 4, conectado a la señal del receptor infrarrojo, es esta la señal que se utilizará para detectar el error en el receptor infrarrojo.

c) Error de led emisor infrarrojo

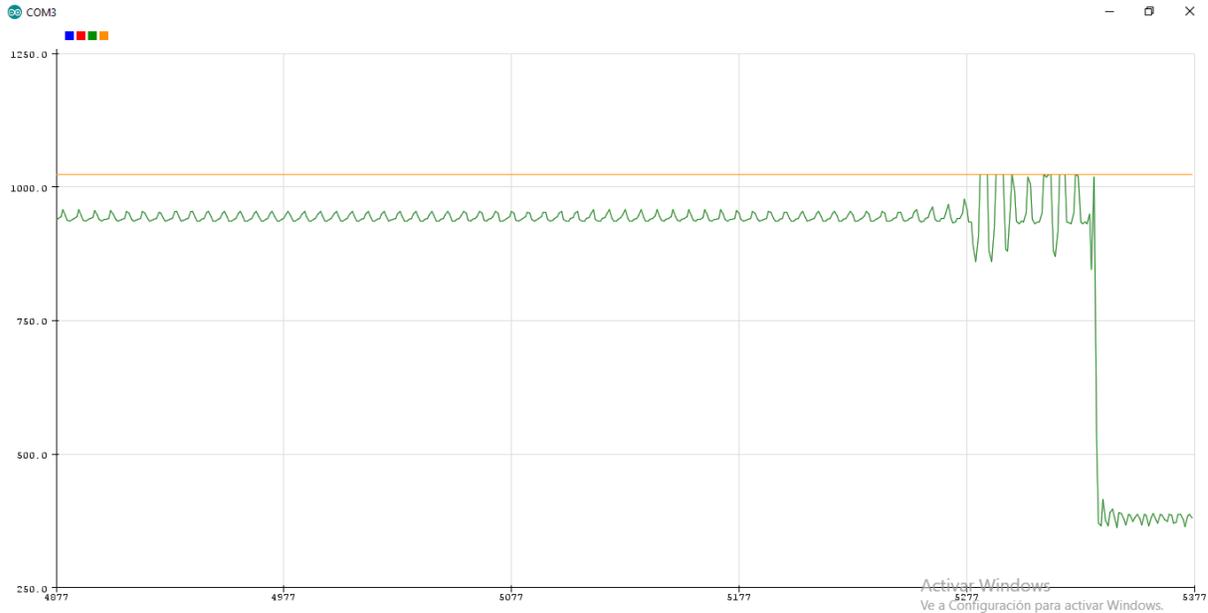


Figura 56: Medidas analógicas mostrando señal con error en el Led emisor infrarrojo

Fuente: Elaboración propia

Donde:

- Azul sensor 1, conectado a la alimentación de la placa.
- Rojo Sensor 2, conectado a la pata 2 del led emisor.
- Verde Sensor 3, conectado a la pata 1 y el transistor 2n3904.
- Naranja Sensor 4, conectado a la señal del receptor infrarrojo.

Cuando existe una falla en el led emisor infrarrojo como una desconexión se registra una fuerte caída de tensión en el sensor 3, es esta la señal que se utilizará para detectar el error en el receptor infrarrojo es esta bajada de tensión que se reconoce con un detector de umbral como se observó en el programa realizado.

3.1.5. App Inventor.

En este punto vamos a ver diseño de la aplicación para Android que creamos, de esta manera se podrá observar el diseño en el celular.

Una vez instalada la aplicación en dicho equipo activaremos el Bluetooth del equipo, luego entramos a la aplicación y seleccionamos “Dispositivos Bluetooth” ahí seleccionamos “00:19:10:11:09:87 HC-05” que es el módulo de Bluetooth que estamos utilizando en el circuito, una vez vinculados los dispositivos seleccionamos “Conectar Dispositivos” y en seguida se activara en verde el botón “Iniciar comandos de voz”, seleccionamos y con ayuda de micrófono de Google podremos decir el comando de voz deseado y automáticamente cada 20 seg. aparecerá el micrófono si deseamos cambiar de canal o subir y bajar el volumen de la televisión, una vez estemos satisfechos seleccionamos “detener comandos de voz” y así podremos ver tranquilamente el programa de televisión elegido.



Figura 57: Inicio de la aplicación creada con App Inventor

Fuente: Elaboración propia

Los reportes de errores que se presentan en el circuito de interfaz se muestran en la aplicación en el espacio reservado para ello, para mostrar inmediatamente al usuario el tipo de error.

3.1.5.1. Código de App Inventor.

A continuación, mostraremos el código diseñado para la aplicación de equipo Android.

El código muestra los seis comandos destinados al control del televisor, ponemos condicionales para inicializar el proceso, llamando un Activity que pide la activación del Bluetooth en caso de que no estuviera activado, e iniciamos los estados de los botones.

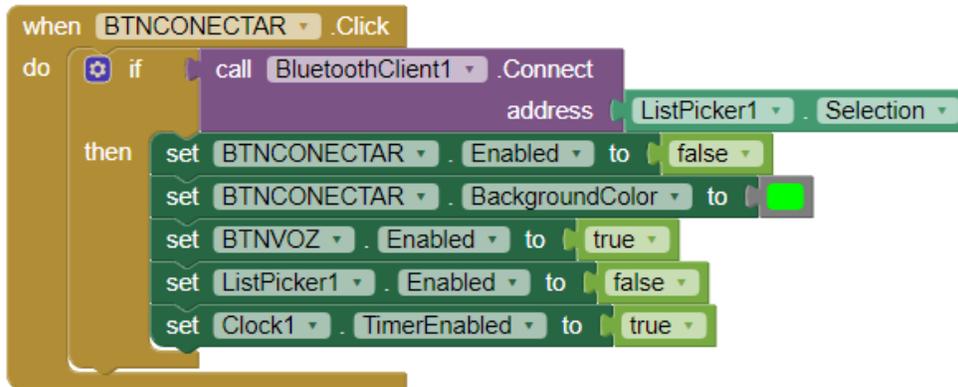
```
when Screen1.Initialize
do
  if not BluetoothClient1.Enabled
  then call ActivityStarter1.StartActivity
  set BTNCONECTAR.Enabled to false
  set Clock1.TimerEnabled to false
  set Clock2.TimerEnabled to false
  set BTNVOZ.Enabled to false
  set BTNDETENER.Enabled to false
  set ListPicker1.Elements to BluetoothClient1.AddressesAndNames
```

El código muestra una lista de los dispositivos Bluetooth encontrados dentro el radio de alcance, una vez seleccionado el Bluetooth a utilizar, modifica el estado del botón conectar a true.

```
when ListPicker1.BeforePicking
do set ListPicker1.Elements to BluetoothClient1.AddressesAndNames

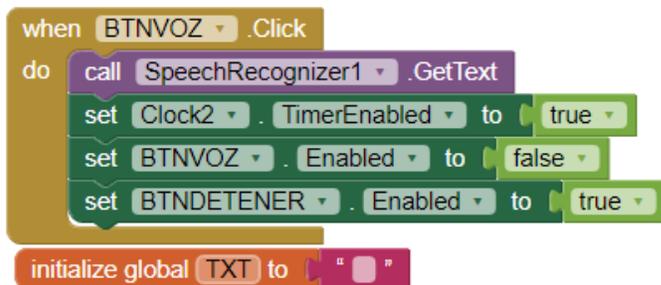
when ListPicker1.AfterPicking
do set BTNCONECTAR.Enabled to true
```

Con el boton conectar en true, inicializa los estados de las variables, el boton conectar cambia a color verde, habilitamos el boton de voz, iniciamos el timer de espera, y borramos la variable de escucha.



```
when BTNCONECTAR .Click
do
  if
  then
    call BluetoothClient1 .Connect
    address ListPicker1 . Selection
    set BTNCONECTAR . Enabled to false
    set BTNCONECTAR . BackgroundColor to #00FF00
    set BTNVOZ . Enabled to true
    set ListPicker1 . Enabled to false
    set Clock1 . TimerEnabled to true
```

Al elegir el botón voz, habilitamos la app para escuchar los comandos, modificamos la variable de escucha para informar que tiene información, y cambiamos de estado la bandera en el botón detener, e iniciamos la variable txt para poder mostrar lo dictado a la app.



```
when BTNVOZ .Click
do
  call SpeechRecognizer1 .GetText
  set Clock2 . TimerEnabled to true
  set BTNVOZ . Enabled to false
  set BTNDETENER . Enabled to true

initialize global TXT to ""
```

La siguiente parte del código, muestra los comandos de voz para el manejo del televisor, como se puede ver los seis comandos son Encender TV, Apagar TV, Subir Canal, Bajar Canal, Subir Volumen y Bajar Volumen, los mismos que se muestran en la parte de la recepción en la aplicación.

```

when SpeechRecognizer1 .AfterGettingText
  result partial
do
  set global TXT to uppercase get result
  if compare texts get global TXT = " ENCENDER TV "
  then call BluetoothClient1 .SendText
    text " a "
  else if compare texts get global TXT = " APAGAR TV "
  then call BluetoothClient1 .SendText
    text " f "
  else if compare texts get global TXT = " SUBIR CANAL "
  then call BluetoothClient1 .SendText
    text " b "
  else if compare texts get global TXT = " BAJAR CANAL "
  then call BluetoothClient1 .SendText
    text " c "
  else if compare texts get global TXT = " SUBIR VOLUMEN "
  then call BluetoothClient1 .SendText
    text " d "
  else if compare texts get global TXT = " BAJAR VOLUMEN "
  then call BluetoothClient1 .SendText
    text " e "
  call SpeechRecognizer1 .Stop

```

Pasado el tiempo programado en el timer, se muestra el txt cargado en la pantalla de la aplicación transmitimos y extraemos el valor escuchado.

```

when Clock1 .Timer
do
  if call BluetoothClient1 .BytesAvailableToReceive > 0
  then set TextBox1 .Text to call BluetoothClient1 .ReceiveText
    numberOfBytes call BluetoothClient1 .BytesAvailableToReceive

when Clock2 .Timer
do
  call SpeechRecognizer1 .GetText

```

Al salir, desconectamos el Bluetooth y limpiamos las variables iniciales junto con los timer, dejando listo las variables para esperar la apertura de la aplicación nuevamente.

```
when BTNSALIR .Click
do
  if BluetoothClient1 .IsConnected
  then call BluetoothClient1 .Disconnect
  call SpeechRecognizer1 .Stop
  set Clock1 .TimerEnabled to false
  set Clock2 .TimerEnabled to false
  close application
```

Al presionar el botón detener hacemos un stop en el grabador de escucha, y cambiamos los valores de las variables btndetener y btnvoz para poder iniciar nuevamente la grabación, sin tener datos grabados.

```
when BTNDETENER .Click
do
  call SpeechRecognizer1 .Stop
  set Clock2 .TimerEnabled to false
  set BTNDETENER .Enabled to false
  set BTNVOZ .Enabled to true
```

CAPITULO 4

4.1. Análisis de Costos.

El presente capítulo hace referencia a todos los gastos que se realizaron tanto en el desarrollo del sistema como en su implementación.

En el diseño e implementación del proyecto, en todo momento se tuvo en cuenta la economía tratando de utilizar herramientas y componentes que no tuvieran un gran costo, pero cuidando que los mismos sean lo más eficientemente posible.

A continuación, veremos los costos directos e indirectos que se utilizó para realizar dicho proyecto:

4.1.1. Costos Directos.

En este punto se menciona los programas y componentes que se utilizó en el desarrollo del proyecto.

Programas y/o Componentes	Precio Bs.-
Placa Arduino Uno	80
Modulo Bluetooth	50
Led infrarrojo Emisor	3
Led infrarrojo Receptor	7
Transistor 2N 3904	2
Cuatro Resistencias	4 x 0.30 = 1.20
Quemado y diseño de la placa	45
Fuente o cargador de 5V	10
Total=	198.20 Bs.-

En este caso ya que el código y diseño del proyecto fue realizado por mi persona no tuvo ningún costo, también los programas como ser el Arduino Uno y el Proteus son de tipo Free por lo cual no tuvieron ningún costo.

4.1.2. Costos Indirectos.

En este punto se menciona los costos extras que se utilizó en el desarrollo del proyecto.

Insumos	Precio Bs.-
Electricidad	60
Internet	180
Carcasa para el circuito (dependiendo el material)	60
Cable USB	15
Total=	315 Bs.-

En este punto existen varias cosas que podrían hacer variar los costos indirectos adicionales esto dependerá de las características que desee el usuario y el tiempo de ejecución de este.

4.1.3. Costo total del proyecto.

El costo total del proyecto lo calcularemos sumando costos directos y los costos indirectos, cabe aclarar que podría existir una pequeña variación en dichos costos.

Costos Directos: 198.20 bs

Costos Indirectos: 315 bs

Total = 513.20 Bs.-

CAPITULO 5

5.1. Conclusiones.

Al culminar el presente proyecto de grado se obtuvieron las siguientes conclusiones y recomendaciones:

- Se implemento con éxito el sistema de control con reconocimiento de voz para un equipo de televisión basándose en el lenguaje Arduino, mejorando la calidad de vida y la autonomía de las personas de la tercera edad,
- Se realizo el algoritmo de reconocimiento de voz por software desarrollando en este caso el algoritmo de Google en sus aplicaciones para sistema operativo Android.
- Se desarrolló una aplicación para sistema operativo Android con compatibilidad con el sistema propuesto de control de equipos de televisión para las personas de la tercera edad con el paquete MIT App inventor, con un alto grado de confiabilidad en espacios con poco nivel de ruido sonoro.
- Se diseño el sistema electrónico con integración con la aplicación móvil para enviar y recibir información con el equipo de televisión mediante el paquete de análisis y simulación de circuitos electrónicos ISIS Proteus.

5.2. Recomendaciones.

- El sistema desarrollado en el presente proyecto obtiene mejores resultados en espacios con poco nivel de ruido sonoro en el ambiente, el cual se puede mejorar con circuitos supresores de ruido.
- Si bien la calidad del dispositivo inteligente no afecta de manera considerable el rendimiento del sistema, la entonación de la voz y el dictado claro de los comandos suele ser un factor para considerar para que el sistema tenga mayor confiabilidad.

- Como muchos dispositivos electrónicos no se recomienda usar el sistema propuesto en entornos con temperaturas extremas, para evitar malos funcionamientos.

Referencias Bibliográficas.

- Alvarado, A., & Salazar, A. (2014). Análisis del concepto de envejecimiento. *Gerokomos*, 6.
- Carmona, J. L. (2009). *Reconocimiento de Voz Codificada sobre Redes IP*. Granada: S/N.
- Gonzales, J. (2010). Codificadores, Decodificadores y Comparadores. En *Curso Completo de Electrónica Digital* (pág. 11). Madrid: S/N.
- Jauregui, J., & Rubin, R. (2012). Fragilidad en el adulto mayor. *Revista del Hospital Italiano de Buenos Aires*, 6.
- Aguilar R., Druet N. (2020) Aprendizaje vivencial en la tercera edad [Figura] Recuperado de <https://observatorio.tec.mx/edu-bits-blog/aprendizaje-vivencial-tercera-edad>
- Mejía J. (2020). Celebremos el día de las personas con discapacidad. [Figura]. Recuperado de <https://www.habitantesiete.com/celebremos-el-dia-de-las-personas-con-discapacidad/>
- Matzner P. (2014). Tetraplejia: causas, tratamientos y cuidados. [Figura]. Recuperado de <https://www.sunrisemedical.es/blog/tetraplejia>
- De Pablos, Fernando;. (20 de octubre de 2008). *casadomo.com*. Recuperado el 15 de abril de 2022, de <https://www.casadomo.com/2008/10/20/control-por-voz>
- Stroustrup, Bjarne (1997), *The C++ Programming Language* (Third ed.). ISBN 0-201-88954-4. OCLC 59193992.
- Microchip. (s.f.). ww1.microchip.com. Obtenido de https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf
- Fernandez Y. (2022). Qué es Arduino, cómo funciona y qué puedes hacer con uno. [Figura]. Recuperado de <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>

Arduino. (naylampmechatronics, s.f.) ([blog/36_tutorial-arduino-y-control-remoto-infrarrojo](#), s.f.).

Lozano R. (2018). modulo Bluetooth hc-05. [Figura]. Recuperado de <https://www.taloselectronics.com/blogs/tutoriales/configurar-modulo-bluetooth-hc-5>

Modulo hc-05. (naylampmechatronics, s.f.) ([inalámbrico/43-modulo-bluetooth-hc05](#), s.f.).

Geekbot Electronics. (12 de abril de 2021). *Geekbot Electronics*. Obtenido de <https://geekbotelectronics.com/tienda/producto/led-infrarrojo-ir/>

Linares, R., Quijano, J., & Holguín, G. (2004). Implementación del protocolo bluetooth para la conexión inalámbrica de dispositivos electrónicos programables. *Scientia et Technica*, 7.

Mayo Foundation for Medical Education and Research. (22 de junio de 2022). *Mayo Clinic*. Obtenido de <https://www.mayoclinic.org/es-es/diseases-conditions/spinal-cord-injury/symptoms-causes/syc-20377890>

MCI Electronics. (15 de Julio de 2022). *Arduino.Cl*. Obtenido de <https://arduino.cl/que-es-arduino/>

Naylamp Mechatronics SAC. (16 de 07 de 2022). *Naylamp Mechatronics*. Obtenido de <https://naylampmechatronics.com/inalambrico/43-modulo-bluetooth-hc05.html>

naylampmechatronics. (s.f.). [blog/36_tutorial-arduino-y-control-remoto-infrarrojo](#).

Quality Unit, LLC. (18 de 06 de 2022). *LiveAgent*. Obtenido de <https://www.liveagent.es/glosario/interfaz-de-usuario-de-voz/>

Robledano, A. (22 de Junio de 2019). *OpenWebinars S.L*. Obtenido de <https://openwebinars.net/blog/que-es-cpp/>

Torres, L. (2005). Capitulo 2. Filtros. . En *S/N* (pág. 25). Mexico: N/A.