



**UNIVERSIDAD TECNOLÓGICA INDOAMÉRICA**  
**FACULTAD DE INGENIERÍAS Y TECNOLOGÍAS DE LA**  
**INFORMACIÓN Y LA COMUNICACIÓN**  
**CARRERA DE INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN**

**TEMA**

---

**DESARROLLO DE UN DISPOSITIVO DE ASISTENCIA DE VOZ PARA  
PACIENTES CON DISCAPACIDAD MOTRIZ EN EL CENTRO DE  
REHABILITACIÓN “BENDICIONES” EN LA CIUDAD DE AMBATO**

---

Trabajo de titulación previo a la obtención del título de Ingeniero en Ciencias de la Computación.

**Autor:** López Granizo Kevin Joel

**Tutor:** Ing. José Luis Varela Aldás

AMBATO-ECUADOR

2022

## APROBACIÓN DEL ASESOR

En mi calidad de catedrático Asesor del Proyecto de grado previo a la obtención del título de Ingeniero en Sistemas, titulado “Desarrollo de un dispositivo de asistencia voz para pacientes con discapacidad motriz en el Centro de Rehabilitación Bendiciones en la ciudad de Ambato”, elaborado por el señor estudiante: López Granizo Kevin Joel. Certifico que dicho proyecto ha sido revisado en todas sus partes y considero que reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del tribunal examinador que se designe.



)LUPDGR  
GLJLWDOPHQWH  
SRU-26( /8,6  
9\$5(/\$ \$/'\$6  
)HFKD

Ing. José Luis Varela Aldás

ASESOR

## DECLARACIÓN DE AUTENTICIDAD

El abajo firmante, declara que los contenidos y resultados obtenidos en el presente proyecto, como requerimiento previo para la obtención del título de Ingeniero en Ciencias de la Computación, son absolutamente originales, auténticos, personales y de exclusiva responsabilidad legal y académica del autor.

A handwritten signature in black ink, appearing to read 'Kevin Joel López Granizo', written over a horizontal line.

López Granizo Kevin Joel

C.I: 1207909142

## APROBACIÓN DEL TRIBUNAL

Los Miembros del Tribunal Examinador designado por la Comisión Académica aprueban el trabajo de titulación cuyo tema es: “Desarrollo de un dispositivo de asistencia voz para pacientes con discapacidad motriz en el Centro de Rehabilitación Bendiciones en la ciudad de Ambato”, de acuerdo con las disposiciones reglamentarias emitidas por la Universidad Tecnológica Indoamérica para la obtención del Título de Ingeniero en Ciencias de la Computación. Por lo tanto, autorizamos al postulante para su impresión, empastado y sustentación pública.

Ambato, marzo 2022



Firmado electrónicamente por:  
**FRANKLIN ADRIAN  
CASTILLO  
LEDESMA**



Firmado electrónicamente por:  
**MARIO ERNESTO  
MIRANDA SANCHEZ**

---

**Ing. Franklin Castillo**

LECTOR 1

---

**Ing. Mario Miranda**

LECTOR 2

## DEDICATORIA

En primer lugar, quiero dedicar este trabajo a Dios por ser el autor de mi vida y permitirme llegar hasta el final de esta etapa y sostenerme en los momentos que todo se derrumbaba con fuerza y valentía para continuar. *Todo lo puedo en Cristo que me fortalece. Filipenses 4:13*

De una manera muy especial lo dedico a mis padres, Manuel mi padre, el cual ha sido mi guía y apoyo en todo momento, Olga mi madre por ser la mujer más importante en mi vida y mi ejemplo a seguir. Sin ellos no tendría el valor de escribir esta dedicatoria.

A mis hermanos Gissela y Josue por siempre brindarme su apoyo y ser un modelo de perseverancia y constancia ante una meta.

*Kevin Joel*

## **AGRADECIMIENTO**

Agradezco primeramente a Dios por darme la sabiduría, el conocimiento y las fuerzas en aquellos momentos cuando creí que no podía lograrlo y haberme permitido llegar hasta este punto, es gracias a él que ahora soy lo que soy.

Agradezco de la manera más profunda a toda mi familia que a pesar de todas las adversidades que pudieron haber llegado al hogar, supieron brindarme su apoyo incondicional en todo momento.

De una forma especial al Ingeniero José Luis Varela el cual fue mi mentor y la persona que me brindó el apoyo necesario, confiando en mí a pesar de que llegaban momentos en que creí que no podía lograrlo y a la vez transmitiendo sus conocimientos para que pueda concluir este trabajo con éxito.

A todos esos docentes que fueron un pilar fundamental desde mi niñez y que ahora gracias a ellos me encuentro donde estoy.

A mis amigos con los que compartí muchas experiencias en todo este camino y pudimos formar lazos de amistad que hasta el día de hoy perduran.

A la Universidad Tecnológica Indoamérica que me abrió las puertas para poder realizar mi formación y haber confiado en mí.

A todos, Dios les bendiga.

*Kevin Joel*

## ÍNDICE DE CONTENIDOS

TEMA .....	i
APROBACIÓN DEL ASESOR .....	ii
DECLARACIÓN DE AUTENTICIDAD .....	iii
APROBACIÓN DEL TRIBUNAL .....	iv
DEDICATORIA .....	v
AGRADECIMIENTO .....	vi
ÍNDICE DE CONTENIDOS .....	vii
ÍNDICE DE TABLAS .....	ix
ÍNDICE DE FIGURAS .....	x
ÍNDICE DE ANEXOS .....	xi
RESUMEN .....	xii
SUMMARY .....	xiii
CAPITULO I .....	1
INTRODUCCIÓN .....	1
EL PROBLEMA .....	1
CONTEXTUALIZACIÓN .....	3
Macro .....	3
Meso .....	3
Micro .....	4
ANTECEDENTES .....	4
CONTRIBUCIÓN TEÓRICA .....	6
Discapacidad .....	6
Sistema de reconocimiento de voz .....	7
Arquitectura de un sistema de reconocimiento de voz .....	8
M5Stack Core2 .....	8
MicroPython .....	9
Aplicaciones Móviles .....	10
Base de Datos .....	10
CONTRIBUCIÓN ECONÓMICA .....	11
CONTRIBUCIÓN SOCIAL .....	11
CONTRIBUCIÓN TECNOLÓGICA .....	11
JUSTIFICACIÓN .....	12
OBJETIVOS .....	13
General .....	13
Específicos .....	13
CAPÍTULO II .....	14
METODOLOGÍA .....	14
DISEÑO DEL TRABAJO .....	14
ÁREA DE ESTUDIO .....	14
MODALIDAD DE INVESTIGACIÓN .....	15
Bibliográfica o documental .....	15
De campo .....	15
Proyectos Especiales .....	15
TÉCNICAS E INSTRUMENTOS .....	16
Entrevista .....	16
ANÁLISIS E INTERPRETACIÓN DE RESULTADOS .....	18

CAPÍTULOS III .....	19
PROPUESTA Y RESULTADOS ESPERADOS .....	19
METODOLOGÍA DE DESARROLLO .....	19
Selección de la Metodología de desarrollo .....	19
Planificación .....	20
Diseño .....	21
Codificar .....	21
Pruebas .....	21
ANÁLISIS DEL SISTEMA ACTUAL .....	21
ESPECIFICACIÓN DE REQUERIMIENTOS .....	22
FASE I: PLANIFICACIÓN .....	23
PLAN DE DESARROLLO DE SOFTWARE .....	23
PLANEACIÓN DEL PROYECTO .....	27
ANÁLISIS DE RESTRICCIONES .....	28
ESTUDIO DE FACTIBILIDAD .....	29
ANÁLISIS COSTO BENEFICIO .....	32
ANÁLISIS ORIENTADO A OBJETOS .....	32
Fase II: DISEÑO DE LA PROPUESTA .....	33
DIAGRAMA DE CLASES .....	40
FASE III: CODIFICACIÓN .....	40
FASE IV: PRUEBAS .....	46
CAPITULO IV .....	51
CONCLUSIONES Y RECOMENDACIONES .....	51
CONCLUSIONES .....	51
RECOMENDACIONES .....	51
BIBLIOGRAFÍA .....	53
ANEXOS .....	55

## ÍNDICE DE TABLAS

<b>Tabla 1:</b> Requerimientos .....	22
<b>Tabla 2:</b> Herramientas para el desarrollo de la propuesta.....	24
<b>Tabla 3:</b> Historia de usuario dispositivo de voz .....	25
<b>Tabla 4:</b> Historia de usuario aplicación móvil.....	25
<b>Tabla 5:</b> Roles y Responsabilidades .....	26
<b>Tabla 6:</b> Plan de Fases.....	27
<b>Tabla 7:</b> Cronograma de Actividades.....	27
<b>Tabla 8:</b> Herramientas existentes .....	29
<b>Tabla 9:</b> Recursos tecnológicos requeridos .....	30
<b>Tabla 10:</b> Costos Directos .....	31
<b>Tabla 11:</b> Costos Indirectos.....	31
<b>Tabla 12:</b> Componentes del M5Stack .....	39
<b>Tabla 13:</b> Lista de Validación .....	49

## ÍNDICE DE FIGURAS

<b>Figura 1:</b> Árbol de problemas .....	2
<b>Figura 2.</b> Dispositivo M5Stack.....	9
<b>Figura 3:</b> Valor del módulo de administrador COCOMO II.....	30
<b>Figura 4:</b> Valor del proyecto COCOMO II.....	31
<b>Figura 5:</b> Diagrama de Caso de Uso.....	33
<b>Figura 6:</b> Interfaz preliminar M5Stack. ....	34
<b>Figura 7:</b> Interfaz de inicio de la app. ....	35
<b>Figura 8:</b> Interfaz segunda pantalla app. ....	36
<b>Figura 9:</b> Diagrama Interacción base de datos .....	36
<b>Figura 10:</b> Diagrama de Clase. ....	40
<b>Figura 11:</b> Función para generar un espectrograma .....	42
<b>Figura 12:</b> Espectrograma de una palabra.....	42
<b>Figura 13:</b> Comando para convertir el modelo tflite en código c.....	43
<b>Figura 14:</b> Codificación pantalla de inicio .....	44
<b>Figura 15:</b> Codificación pantalla principal .....	46
<b>Figura 16:</b> Conexión Wifi .....	47
<b>Figura 17:</b> Funcionamiento final .....	48

## ÍNDICE DE ANEXOS

<b>ANEXOS 1:</b> Fotografías M5Stack.....	55
<b>ANEXOS 2:</b> Interfaz de Inicio App .....	56
<b>ANEXOS 3:</b> Codificación de la Interfaz M5stack y conexión Wifi.....	56
<b>ANEXOS 4:</b> Análisis de Audio de la base de datos.....	62
<b>ANEXOS 5:</b> Creación del modelo de la red neuronal .....	69
<b>ANEXOS 6:</b> Conversión del modelo a código C (TensorFlowLite) .....	73
<b>ANEXOS 7:</b> Configuración del micrófono por entrada pdm .....	74
<b>ANEXOS 8:</b> Inicialización de la M5Stack y Comunicación Serial.....	75
<b>ANEXOS 9:</b> Procesamiento de audio y Comparación de la red neuronal .....	76
<b>ANEXOS 10:</b> Pruebas de Integración .....	78
<b>ANEXOS 11:</b> Pruebas de Validación.....	78
<b>ANEXOS 12:</b> Acta de Conformidad .....	79
<b>ANEXOS 13:</b> Manual de Usuario.....	81

**UNIVERSIDAD TECNOLÓGICA INDOAMÉRICA**  
**FACULTAD DE INGENIERÍAS Y TECNOLOGÍAS DE LA**  
**INFORMACIÓN Y COMUNICACIÓN**  
**CARRERA DE INGENIERÍA EN SISTEMAS**

**TEMA:** “DESARROLLO DE UN DISPOSITIVO DE ASISTENCIA DE VOZ PARA PACIENTES CON DISCAPACIDAD MOTRIZ EN EL CENTRO DE REHABILITACIÓN “BENDICIONES” EN LA CIUDAD DE AMBATO”

**AUTOR:** López Granizo Kevin Joel

**ASESOR:** Ing. José Luis Varela Aldás

**RESUMEN**

El centro de Rehabilitación Bendiciones de la ciudad de Ambato es una entidad privada que brinda tratamientos a niños y adultos con lesiones neurológicas, física y demás. Sin embargo, existe un problema latente en cuanto a la dependencia de los pacientes, ya que los cuidadores de un paciente con problemas motrices no pueden estar las 24 horas del día supliendo sus necesidades. Debido a esto, el objetivo del proyecto es el desarrollo de un dispositivo de asistencia de voz para personas con discapacidad motriz. La metodología utilizada es XP. Se utilizó Micropython para la programación del dispositivo de reconocimiento de voz (M5StackCore2) y Tensorflow como una biblioteca para el aprendizaje mediante archivos de audio como base de datos. Además, para el desarrollo de la aplicación móvil en donde se recibirá todas las peticiones del paciente se lo diseñó mediante App Inventor. Por otro lado, se realizó una conexión inalámbrica entre el dispositivo de reconocimiento de voz y la app. Se obtuvo un dispositivo capaz de reconocer la voz del paciente y notificar las necesidades solicitadas al cuidador a través de la aplicación móvil. Finalmente, gracias al desarrollo de este proyecto fue posible que el niño tenga dependencia al sentirse útil y poder solicitar las diferentes necesidades que tiene a través de un dispositivo de asistencia de voz, mejorando su calidad de vida.

DESCRIPTORES: App Inventor, Dispositivo de Asistencia de voz, MicroPython, M5StackCore2, TensorFlow.

**UNIVERSIDAD TECNOLÓGICA INDOAMÉRICA**  
**FACULTAD DE INGENIERÍAS Y TECNOLOGÍAS DE LA**  
**INFORMACIÓN Y COMUNICACIÓN**  
**CARRERA DE INGENIERÍA EN SISTEMAS**

**SUBJECT:** “DEVELOPMENT OF A VOICE ASSISTANCE DEVICE FOR PATIENTS WITH MOTOR DISABILITIES IN THE REHABILITATION CENTER "BENDICIONES" IN THE CITY OF AMBATO”

**AUTHOR:** López Granizo Kevin Joel

**TUTOR:** Ing. José Luis Varela Aldás

**SUMMARY**

The Rehabilitation Center “Bendiciones” in Ambato is a private institution that treats children and adults with serious neurological and physical injuries. However, there is a big issue when patients depend on themselves since for his caregivers it is impossible to meet all his needs during the whole day. This research aims to develop a voice assistance device for people with motor disabilities. It was used the methodology XP. Micropython for the programming of the speech recognition device (M5StackCore2) and TensorFlow as a library for learning using audio files as a database. In addition, App Inventor was used to develop the mobile application so it will be easier to receive all the commands from the patient. On the other hand, a wireless connection was made between the speech recognition device and the app. A device capable of recognizing the patient's voice and notifying the caregiver of the needs requested through the mobile application was obtained. Finally, thanks to the development of this project, it was possible for the child to become independent because he is able to ask and request different things or necessities by using the voice assistance device, significantly improving his quality of life.

DESCRIPTORS: App inventor, MicroPython, M5stack, Voice assist device,  
TensorFlow

# **CAPITULO I**

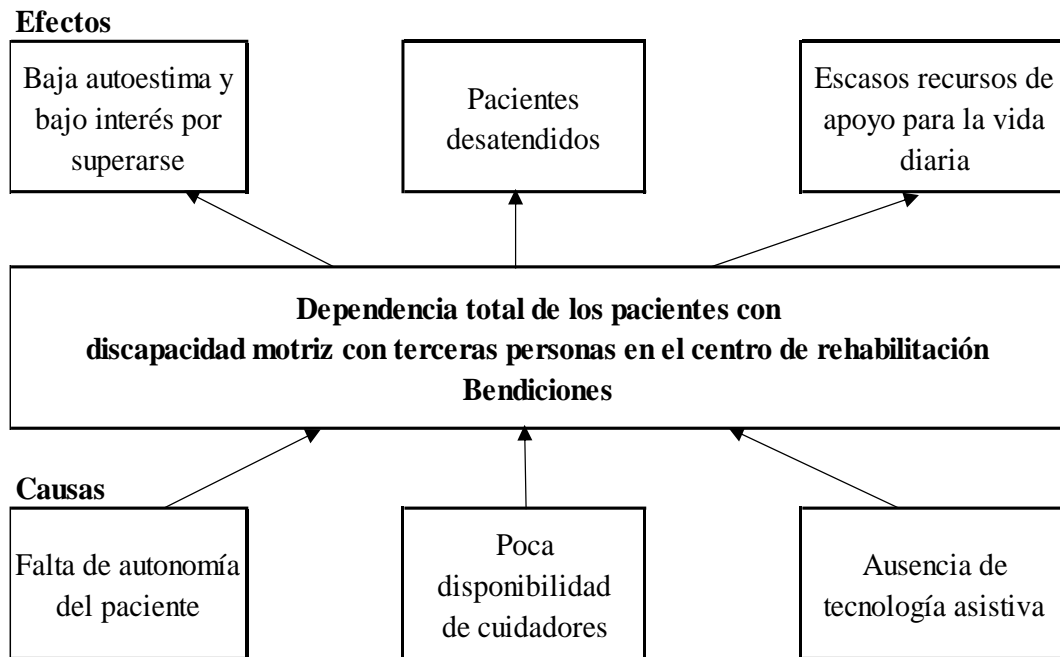
## **INTRODUCCIÓN**

### **EL PROBLEMA**

Hoy en día las personas con alguna discapacidad física, y específicamente con problemas motrices enfrentan diferentes dificultades en su diario vivir. Según una encuesta realizada por la Organización Mundial de la Salud sobre la discapacidad, más de 1000 millones de personas padecen algún tipo de discapacidad, y de estos un aproximado de 190 millones tienen la mayor parte de su cuerpo paralizada. Es por esto que en el centro de rehabilitación Bendiciones en la ciudad de Ambato se ha podido identificar un problema latente en cuanto a sus pacientes. La dependencia total de los pacientes con discapacidad motriz con terceras personas, ha sido una problemática que afecta al entorno de este centro de rehabilitación, ya que los pacientes luego de su terapia no pueden realizar actividades por sí solos, debido a que la persona que lo padece presenta inmovilización de los miembros superiores e inferiores, es decir está paralizada, requiriendo la ayuda de otra persona, sin embargo, en ocasiones estos cuidadores no pueden estar presentes las 24 horas del día a lado de la persona con discapacidad.

Por otro lado, el problema identificado en relación a la dependencia de los pacientes de otras personas, además de todo esto, también, los pacientes al tener una falta de autonomía, puede provocar una baja autoestima y a la vez un bajo interés de superación, ya que se conforman con la terapia realizada y la ayuda de otras personas las 24 horas del día. Además, existe una limitada asistencia para los pacientes, debido a que los profesionales que realizan la terapia no pueden estar las 24 horas del día con ellos ayudándoles, lo que provoca una incapacidad de realizar actividades. Otra razón muy importante por la que se da este problema dentro del centro de rehabilitación es que existe ausencia de tecnología asistiva, siendo esta una tecnología ayudada de cualquier tipo de producto que sirve de asistencia y apoyo para mejorar las capacidades de personas con problemas motrices y en general con discapacidad, es por esto que a través de esta ausencia

hay escasos recursos que sirven de apoyo para la vida diaria del paciente. En la Figura 1 se muestra el árbol de problemas con sus causas y efectos.



**Figura 1:** Árbol de problemas

**Elaborado por:** Kevin López

## **CONTEXTUALIZACIÓN**

### **Macro**

Ecuador es un país situado en América del Sur, el cual es uno de los más pequeños de acuerdo a su geografía en comparación a los demás países. De acuerdo a datos estadísticos encontrados, Ecuador cuenta con 17.684.536 de habitantes en el año 2021 de acuerdo a los primeros meses. En el Ecuador se creó el programa Misión Solidaria Manuela Espejo en el año 2007 encargado de registrar y hacer un diagnóstico de las personas con alguna discapacidad. De esta manera se pudo tener una idea más clara del número de personas que tienen discapacidad y así brindarles ayuda necesaria en su diario vivir mediante este programa [1].

En la página web de CONADIS (Consejo Nacional Para La Igualdad De Discapacidades) se encuentran publicados datos estadísticos relacionados con la discapacidad. Existe un número de 470,820 personas con discapacidad, de las cuales 5,52% tiene discapacidad psicológica, 11,54% visual, 14,13% auditiva, 23,11% intelectual y con el 45,69% la discapacidad física, siendo esta la de mayor porcentaje. Por otro lado, la mayor prevalencia se encuentra en los hombres con aproximadamente 56% y las mujeres el 44%. Cabe recalcar que las personas con un rango de edad de 36 a 64 años son las que más tienen alguna discapacidad, y entre los 0 a 3 años existe un bajo porcentaje de discapacidad [2].

### **Meso**

En la provincia de Tungurahua, de acuerdo a datos publicados en la página web del Consejo Nacional Para La Igualdad De Discapacidades, existen 13,296 habitantes con discapacidad de los cuales 4,807 presentan discapacidad física, representando el 36,15%. Dado esto en la provincia se ha evidenciado que no existe una inclusión total hacia estas personas, por ende, hay pocos lugares que les ayuden a una buena rehabilitación en relación a su discapacidad motriz, es por esto que, de acuerdo a una investigación se pudo evidenciar un caso en la provincia de Tungurahua. El Ministerio de Salud Pública otorgó un carné de discapacidad física del 76% a una niña. Dado, esto la madre de la niña fue a

presentar una queja contra el Centro Ortopédico Vinicio en Ambato, en la Coordinación Zonal 3 de la Defensoría del Pueblo ya que este centro no cumplió con la realización de una prótesis para la niña con discapacidad, la cual necesitaba para seguir con su tratamiento. Con el pasar de los meses la niña seguía sin recibir la prótesis, y esto hizo que afectara su rehabilitación, tratamiento y mejora [2].

### **Micro**

En la ciudad de Ambato se encuentra el Centro de Rehabilitación Física y Neurológica Bendiciones. Este centro brinda atención a las personas, especialmente a niños con discapacidad física y neurológica. Actualmente, se encuentra en constante crecimiento con el fin de mejorar los métodos de rehabilitación y técnicas de recuperación garantizando la salud de los pacientes. Sin embargo, se ha podido evidenciar un problema en cuanto a sus pacientes, debido a que luego de las distintas terapias que se realizan, los profesionales no pueden estar las 24 horas del día con ellos cuidándoles, al igual que sus cuidadores no pueden estar todo el tiempo a lado de la persona con discapacidad motriz, aun mas cuando es parapléjica. Dado esto surge la necesidad de contribuir mediante el desarrollo de un dispositivo de asistencia de voz para que el cuidador pueda conocer las necesidades del niño cuando no se encuentra presente; como principal beneficiario es un niño paciente del Centro de Rehabilitación para mejorar su calidad de vida. Dado esto, se pretende dar una solución a la problemática encontrada.

### **ANTECEDENTES**

Debido a que la discapacidad física y en general se presenta de diferentes formas y niveles, se han creado diversos prototipos, métodos, aplicaciones como ayuda y asistencia para las personas que necesitan de un apoyo, cada uno teniendo en cuenta el nivel de movilidad de la persona discapacitada. En este contexto a continuación se presentan obras de relevancia asociadas a la temática abordada.

En el trabajo de graduación con tema “SISTEMA DOMÓTICO DE APOYO PARA PERSONAS CON DISCAPACIDAD MOTRIZ MEDIANTE TECNOLOGÍA MÓVIL Y RECONOCIMIENTO DE VOZ” de Geesela del

Rosario Alban Mollocana, desarrollado en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, UTA, Ambato 2018, implementó un sistema de reconocimiento de voz para personas con discapacidad motriz. Para la realización de este proyecto se creó una aplicación móvil desarrollada en App Inventor para Android, además del uso de Raspberry PI. Este sistema permite a las personas con discapacidad controlar varias partes de la vivienda, como la seguridad, iluminación y otras áreas específicas [3].

En el trabajo de grado titulado “SISTEMA DE RECONOCIMIENTO DE VOZ PARA PERSONAS CON DISCAPACIDAD MOTRIZ” de Jorge Alejandro Gaitán Albarracín, Luis Ernesto Velasco Waltero, Juan Camilo Valencia Hurtado y Luis Alejandro Tocarruncho Ariza, desarrollado en la Facultad de Ingeniería, Unipanamericana, Bogotá 2019, diseñaron un sistema de reconocimiento de voz para personas con discapacidad motriz. Para su desarrollo crearon una aplicación móvil con fines domóticos haciendo uso de la placa ESP32 y AI2 como software de desarrollo para la aplicación móvil para Android. La app da la posibilidad de solicitar diferentes acciones mediante Google Assistant a través del reconocimiento de voz para de esa manera tener un control dentro de la casa y generar dependencia en las personas con discapacidad [4].

En el trabajo de grado que tiene como título “CONTROL DE MANDO POR VOZ PARA PERSONAS CON DISCAPACIDAD MOTRIZ” de Edgar Oswaldo Garnica Cárdenas, realizado en la Facultad de Ingeniería, Universidad Católica de Colombia, Bogotá 2018, desarrolló un dispositivo de reconocimiento de voz. Este dispositivo brinda la posibilidad de controlar la vivienda de una persona a través de dispositivos eléctricos. El módulo utilizado fue Elechouse Voice recognition V3, el cual sirve para reconocimiento de voz e incluye una placa que almacena muchos comandos. Por otro lado, la programación fue realizada en Arduino con la utilización de las librerías necesaria para hacer uso del módulo ya mencionado. [5].

En el trabajo de titulación con tema “SISTEMA DOMÓTICO CONTROLADO POR VOZ PARA PERSONAS CON DISCAPACIDAD EN EXTREMIDADES SUPERIORES, UTILIZANDO TARJETA RASPBERRY PI” de Daniel

Gerardo Tomala Cuenca, Universidad Politécnica Salesiana, Guayaquil 2018, implementó un sistema domótico controlado por voz a través de Raspberry Pi 3. Este está comunicado mediante Wifi a diferentes actuadores como el sistema de iluminación en diferentes áreas de la vivienda, seguridad de la puerta eléctrica, y otros dispositivos. El reconocimiento de voz es a través de un micrófono que permite tomar todas las solicitudes por parte de la persona con discapacidad motriz y de esa manera ayudar a la misma para que pueda superarse en su diario vivir [6].

## **CONTRIBUCIÓN TEÓRICA**

### **Discapacidad**

En cuanto a la teoría y la aplicación del conocimiento para el presente proyecto, este tiene un amplio campo de estudio acorde a las temáticas planteadas, conociendo todo lo relacionado al tema en estudio, para poder llegar a sustentar el mismo. En primer lugar, es necesario conocer que es el término discapacidad. A la discapacidad se la puede definir como una restricción que tiene una persona, imposibilitándole realizar ciertas tareas, acciones o algún tipo de actividad. Dado esto, las causas pueden ser física, sensoriales o más, produciendo un problema en la vida diaria, provocando una baja autoestima a nivel familiar, psicológico y social [7], [8].

De acuerdo a la afectación que se tenga se puede identificar diversos tipos de discapacidad, los cuales se plasman a continuación.

La discapacidad sensorial auditiva hace referencia a la falta de ciertas partes y estructuras que conforman todo el sistema auditivo. Las personas con esta discapacidad tienen limitaciones al escuchar normalmente, unas son completamente sordas, provocando problemas al momento de realizar una acción considerada normal [14].

La discapacidad sensorial visual se la puede definir como las deficiencias de ciertas estructuras fundamentales para que una persona pueda ver con normalidad, es decir funciones que abarcan el sistema nervioso. Una persona con este tipo de discapacidad pierde parcial o completamente la visión, limitándola a realizar una

tarea en específico [9], [10].

La discapacidad mental hace referencia a la falta considerable del funcionamiento normal en cuanto a la intelectualidad causada por alteraciones en el embrión, errores o factores genéticos y enfermedades que han provocado que la persona no tenga la capacidad de aprender, razonar con normalidad y a la vez desenvolverse en el ámbito social [10], [11].

En cuando a la discapacidad motriz, en la cual se centra el presente trabajo, este tipo de discapacidad hace referencia al déficit de la capacidad que tienen las personas del movimiento, manipulación de objetos, imposibilitando que realicen todo tipo de actividad que requiera movimiento, ya sea de sus extremidades superiores e inferiores. Esta discapacidad se origina cuando existen problemas en el área motriz de la persona, involucrando diferentes trastornos que pueden tener [12].

El trastorno físico periférico se puede definir como la afección a las extremidades, músculos y huesos. Este problema puede ser causado desde el nacimiento por la mal formación de los huesos u extremidades o también por lesiones que pudieron tener y que afectaron su sistema motriz.

Otro trastorno es el neurológico, el cual se basa en una lesión dentro del cerebro en una parte importa la cual envía y recibe toda la información necesaria para el ser humano la cual mueve al cuerpo, llamada corteza motora cerebral. Esto produce afectaciones en el movimiento de la persona como por ejemplo mediante tumores, traumatismos, enfermedades y más.

### **Sistema de reconocimiento de voz**

Al hablar de reconocimiento de voz y específicamente del sistema, a este se lo puede definir como un dispositivo, herramienta o instrumento que es capaz de reconocer la voz que emite la persona y de esta manera aceptar esa información enviada para posteriormente plasmarla en texto, imagen o video, o también realizando una acción en específico.

Cabe recalcar que una de las características principales que tienen los sistemas de

reconocimiento de voz como se dijo anteriormente, es que son capaces de identificar las señales, órdenes que se da mediante la voz y poder convertirlas en algo visible [13].

### **Arquitectura de un sistema de reconocimiento de voz**

El funcionamiento de un sistema de reconocimiento de voz consta de dos partes muy importantes tales como el extractor de características y el clasificador. Su funcionamiento se basa primeramente en la emisión de ondas mediante la voz, las cuales son recibidas por el analizador reconociendo la orden. Seguidamente el dispositivo de reconocimiento de voz realiza la representación y la plasma mediante una acción o mensajes en diferentes formas. Dado esto, en el presente trabajo se pretende desarrollar un dispositivo de asistencia de voz, es decir que el usuario, en este caso el paciente, tendrá que pronunciar una palabra y así el dispositivo la reconocerá y enviará la información que se desea transmitir en forma de mensaje [14].

### **M5Stack Core2**

El M5Core2 es el dispositivo de segunda generación de la serie de kits de desarrollo M5Stack, que mejora aún más las funciones de la generación original.

El MCU es un modelo ESP32. Dispone de interfaz WiFi y Bluetooth e incluye una memoria flash de 16 MB y PSRAM de 8 MB, interfaz USB TYPE-C para cargar, descargar programas y comunicación en serie, una pantalla táctil capacitiva integrada de 2.0 pulgadas y un motor de vibración incorporado.

M5Core2 también cuenta con un módulo reloj RTC incorporado que puede proporcionar una sincronización precisa. La fuente de alimentación es administrada por un chip de administración de energía AXP192, que puede controlar el consumo de energía de la base y un indicador de energía LED verde incorporado ayuda a notificar al usuario del nivel de la batería. La capacidad de la batería se ha actualizado a 390 mAh, que puede alimentar el núcleo durante mucho más tiempo que el modelo anterior.

El M5Core2 conserva la ranura de la tarjeta TF (microSD) y los altavoces. Sin

embargo, para garantizar una salida de sonido de mayor calidad, el chip amplificador de potencia de la interfaz de audio digital I2S se utiliza para prevenir eficazmente la distorsión de la señal. Hay botones de encendido y reinicio independientes en el lado izquierdo y en la parte inferior de la base.

Los pequeños círculos de la parte frontal de la pantalla son botones capacitivos que son programables. Hay una pequeña placa de expansión en la parte posterior de la base con un sensor IMU de 6 ejes y un micrófono. La plataforma de desarrollo y el lenguaje de programación soportados por M5Stack Core2: Arduino, UIFlow usando Blockly, lenguaje MicroPython. En la figura 2 se muestra el dispositivo M5Stack [15],



**Figura 2.** Dispositivo M5Stack

**Tomado de:** Mouser.ec

### **MicroPython**

MicroPython se trata de un compilador completo y a la vez es intérprete de Python, siendo este un lenguaje de programación, con el objeto de ser creado para poder trabajar en microcontroladores. Este tiene la función de ejecutar todo el código de programación realizado y hacer posible que el hardware, en este caso el dispositivo de reconocimiento de voz ejecute acciones. A continuación, se mencionan algunas de las características más resaltantes de MicroPython que lo diferencian de otros:

- Tiene varias librerías para la ejecución de tareas.
- Tiene un editor de código propio muy sencillo.
- Es extensible [16].

### **Aplicaciones Móviles**

El término de aplicación móvil se lo puede definir como un software informático que fue diseñado para que este se ejecute mediante un dispositivo móvil como smartphones, tabletas, teléfonos inteligentes y más, permitiendo al usuario efectuar un variado conjunto de tareas. Cabe recalcar que hoy en día las aplicaciones móviles son mejor conocidas como apps, Es por esto que, su principal objetivo es mejorar la productividad personal mediante el dispositivo. Dentro del proyecto, la aplicación móvil tendrá importancia ya que servirá como una interfaz de visualización en forma de mensajes de las necesidades que el paciente pida a través del dispositivo de reconocimiento de voz.

Existe una clasificación para las aplicaciones móviles: nativas e híbridas. Las aplicaciones nativas son aquellas que están desarrolladas bajo un lenguaje específico que hace que la funcionalidad de esta sea estable para el sistema operativo para el que fue desarrollado, pero recalando que solo servirán exclusivamente para ese sistema operativo. Las aplicaciones web usan lenguaje y tecnología que se usa exclusivamente en la web y son arrancados en un explorador independientemente del sistema operativo. Por último, están las aplicaciones híbridas que son una combinación de las dos que se mencionan anteriormente, con una herramienta adecuada permite que la aplicación pueda ser utilizada en cualquier sistema operativo [17].

### **Base de Datos**

Las bases de datos hacen referencia a un conjunto de datos en donde se almacena toda la información que se requiera, manteniéndola de forma más organizada y por ende de fácil acceso. En el presente trabajo, se usa una base de datos almacenada en carpetas, los cuales son audios para entrenamiento del reconocimiento de palabras claves. Los audios que sirven para entrenamiento son de las distintas palabras que se desea reconocer. [18].

## **CONTRIBUCIÓN ECONÓMICA**

Desde el punto de vista económico es necesario que el Ecuador se convierta en un país que genere dispositivos y herramientas tecnológicas que sirven como ayuda a todas las personas que tienen discapacidad motriz. Mediante el dispositivo de reconocimiento de voz (M5stackCore2) se estaría contribuyendo al paciente con atrofia muscular espinal ya que no tiene que realizar gastos, además que al utilizar el M5stackCore2 existe un gran ahorro en hardware, ya que

Por otro lado, existe un ahorro considerable en el cuidado del paciente debido a que gracias a este dispositivo de reconocimiento de voz los cuidadores podrán ahorrarse varias herramientas de alto precio que puedan ayudar en cierto modo a dar independencia al paciente.

El uso de software libre es un punto a favor debido a que no se generan costos adicionales en software, ya que hoy en día existen varias herramientas gratuitas las cuales permiten realizar desarrollo y programación de una manera libre.

## **CONTRIBUCIÓN SOCIAL**

Desde el punto de vista social el proyecto tiene una gran contribución ya que se orienta a las personas con discapacidad motriz y especialmente un paciente del Centro de Rehabilitación Bendiciones en la ciudad de Ambato, el cual mediante el dispositivo de reconocimiento de voz puede pedir sus necesidades y sentirse útil, a la vez que aumenta su estado emocional, social y psicológico.

Además, gracias a la implementación de este dispositivo el paciente mejora su calidad de vida ya que gracias a diversos factores como el poder hacer algo por su cuenta, genera un poco de independencia y bienestar para su vida.

## **CONTRIBUCIÓN TECNOLÓGICA**

Desde el punto de vista tecnológico la implementación del proyecto permite repotenciar la tecnología en el ámbito de la salud física y motriz dentro de un hogar, ya que no todas las personas tienen la posibilidad de contar con dispositivos de tecnología asistiva. Además, permite abrir nuevos caminos para el

desarrollo de nuevas tecnologías que ayuden a potenciar diferentes procesos tecnológicos en la salud y a futuro que la generación actual esté dispuesta y comprometida en involucrarse mucho más en la tecnología como beneficio mas no como un desperdicio de recursos y tiempo.

Por otro lado, el presente proyecto tiene contribución tecnológica ya que gracias a esta se tiene una mejor calidad de vida, además de que se conoce nuevo hardware y software como el M5StackCore2, el cual permite tener mejores resultados gracias a sus múltiples características en cuanto a otros dispositivos o elementos, teniendo diversos usos. Gracias a este tipo de tecnología también existe una contribución, pues se trata de una creación original, innovadora la cual tiene un interés y utilidad para personas con discapacidad motriz.

## **JUSTIFICACIÓN**

A lo largo de la historia y hoy en día la tecnología ha tenido un avance considerable. Gracias a estos avances en la actualidad se pueden utilizar medios tecnológicos para ayudar, facilitar y satisfacer necesidades de las personas en su diario vivir. De acuerdo al problema encontrado con respecto a los pacientes y personas con discapacidad motriz, es necesario el desarrollo de un dispositivo de reconocimiento de voz para que de esa manera lo usuarios puedan solicitar las necesidades que tienen a sus cuidadores. De esta forma se estaría ayudando y optimizando el proceso de pedir una necesidad, a la vez que se mejora la calidad de vida de las mismas y se genera un cierto grado de dependencia.

La presente investigación mediante el uso de un dispositivo de reconocimiento de voz pretende mejorar la calidad de vida de las personas con discapacidad motriz. Este dispositivo trae beneficios, ya que permite a las personas pedir alguna necesidad únicamente con su voz, es decir el usuario dice una palabra y esta se plasma en una aplicación móvil para de esa manera satisfacer, conocer la petición que el usuario está solicitando y llegar a una buena toma de decisiones.

Este dispositivo es orientado para personas con discapacidad motriz en general, sin embargo, también puede ser usado por personas sin ninguna discapacidad, ya que es un dispositivo que reconoce la voz del usuario y mediante una palabra

clave llega a una app solicitando algo. Esta propuesta a futura será implementada y podrán beneficiarse varias personas como un método para pedir una necesidad de una manera más y eficaz.

La investigación es factible debido a que todos los materiales utilizados para el desarrollo del proyecto se los puede adquirir sin ningún problema. También, se cuenta con suficientes fuentes bibliográficas confiables y verídicas que sirven como apoyo y guía para el desarrollo del presente proyecto.

Por otro lado, esta investigación es factible debido a que es de mucho interés por parte de la propietaria del Centro de Rehabilitación Bendiciones y especialmente por los principales beneficiarios que son los pacientes y con discapacidad motriz. Además de que este proyecto sirve como base para investigaciones e implementaciones futuras.

Finalmente, este proyecto mediante el prototipo planteado, tiene además como finalidad un término muy importante, el cual es inclusión social. Dado esto, se puede afirmar que el objetivo principal de esta investigación, es brindar una ayuda a las personas con discapacidad y población en general, mejorando su calidad de vida, para así brindarles una mayor autonomía dentro de sus hogares.

## **OBJETIVOS**

### **General**

Desarrollar un dispositivo de asistencia de voz para pacientes con discapacidad motriz en el centro de rehabilitación “Bendiciones” en la ciudad de Ambato.

### **Específicos**

- Analizar las características de autonomía de los pacientes con discapacidad motriz para determinar los requerimientos de asistencia mediante observación de campo.
- Diseñar un sistema de asistencia para facilitar el trabajo de los cuidadores de pacientes con discapacidad motriz a través de tecnología de bajo costo.
- Construir un dispositivo portátil controlado mediante comandos de voz para atender las necesidades de los pacientes con discapacidad motriz.

## **CAPÍTULO II**

### **METODOLOGÍA**

#### **DISEÑO DEL TRABAJO**

- Una entrevista para la recolección de información acerca de los requerimientos y necesidades que tiene el paciente con atrofia muscular espinal.
- Programación del dispositivo de reconocimiento de voz M5stackCore2
- Diseño de una aplicación móvil que permita recibir al cuidador las necesidades que tenga el paciente en forma de mensajes.
- Conexión ente el dispositivo de reconocimiento de voz y la aplicación móvil.
- Pruebas finales con el paciente.

#### **ÁREA DE ESTUDIO**

El área de estudio de este proyecto se basa principalmente en las ciencias de la computación, las cuales se pueden definir como un conjunto de disciplinas basadas en la tecnología. Además, de un término muy importante como es la programación y sus lenguajes que ayudan a diseñar diferentes programas, modelos para hoy en día resolver diferentes problemáticas y así lograr una automatización computacional.

Además, el área de estudio también se centra directamente al término dispositivo de reconocimiento de voz, ya que es el producto que se lo va a desarrollar con la ayuda de varios, métodos, herramientas, lenguajes de programación, y así tener un buen producto final.

**Aplicaciones móviles:** Se utilizará para la realización de la aplicación móvil que se encargará de recibir en forma de mensaje las necesidades del paciente.

**Base de datos:** Se utilizará una base de datos para obtener la información para la creación de la aplicación móvil, así como también diferentes procesos

## **MODALIDAD DE INVESTIGACIÓN**

La modalidad para el desarrollo de este proyecto será aplicada debido a que se centra en un dispositivo de reconocimiento de voz, el cual se lo va a programar y desarrollar de acuerdo a las necesidades existentes, con el propósito de satisfacer y dar un poco de independencia al paciente con discapacidad motriz, específicamente con atrofia muscular espinal.

### **Bibliográfica o documental**

Gracias al método bibliográfico el presente proyecto se fortalece en la búsqueda de información en diferentes documentos, tesis, artículos científicos y así recolectar, analizar, seleccionar información coherente y necesaria; además de la utilización de trabajos de titulación del área en estudio que sirvieron como base para la realización del presente proyecto, permitiendo construir un marco teórico robusto, cumpliendo con las normativas de citas bajo el formato IEE que sustente y respalde la implementación del proyecto.

### **De campo**

La investigación de campo es adoptada en el proyecto ya que se realiza la visita a las instalaciones del Centro de Rehabilitación, así como al hogar del paciente, lo cual permitirá dialogar e interactuar con las personas con el objetivo de conocer la problemática y el proceso de automatización, es necesario tener contacto directo con los involucrados en cada una de las tareas; la observación es parte fundamental para que los resultados de las visitas sean más exitosos.

### **Proyectos Especiales**

Se utilizó la Modalidad de Proyecto Especial pues se trata de una creación original, innovadora la cual tiene un interés y utilidad dentro de la empresa. Se trata de dar una solución tecnológica como un dispositivo de reconocimiento de

voz para necesidades que será usado por el paciente con discapacidad motriz. Los proyectos especiales hacen referencia a proponer diseños de utilidad e innovación ante un problema o una necesidad existe dentro de una organización, empresa o una persona.

## **TÉCNICAS E INSTRUMENTOS**

Se detalla las técnicas que se utilizan para la obtención de la información necesaria que será usada como base a la toma de decisiones:

### **Entrevista**

En primer lugar, se realizará una entrevista ya que es una técnica que se orienta a establecer contacto directo con la persona que provee información verídica y confiable para la realización del proyecto. La persona entrevistada fue la directora del Centro de Rehabilitación Bendiciones, la Licenciada María Belén Camino, con el propósito de recolectar información y las especificaciones necesarias en cuanto a un paciente en específico, un niño con atrofia muscular espinal, así como los diferentes requerimientos que se desea para poder ayudar al paciente.

Esta técnica tiene como instrumento el Cuestionario, el cual es un conjunto de preguntas para generar datos necesarios para de esa manera alcanzar los objetivos del proyecto de investigación. Para el proyecto, en el cuestionario se presentaron preguntas abiertas semiestructuradas, es decir, alternando preguntas estructuradas y con preguntas espontáneas, fáciles y breves, así como preguntas que se fueron originando a lo largo de la entrevista lo cual fue de mucha ayuda para la realización del proyecto.

### **Entrevista**

1.- ¿Cuál es su nombre y su cargo?

Soy la Licenciada María Belén Camino, directora del centro de rehabilitación bendiciones

2.- ¿Qué tiempo funciona la empresa / organización?

Ya lleva 13 años que el centro de rehabilitación funciona, el cual fue fundado en el año 2008

3.- ¿Qué tiempo trabaja en la empresa / organización?

Desde la fundación de este centro de rehabilitación

4.- ¿A qué se dedica la empresa / organización?

Es una entidad privada destinada a la rehabilitación física ubicado en la ciudad de Ambato que ofrece los tratamientos profesionales en Neurorehabilitación, Neuropsicología, Estimulación de praxia fina y cognitiva Terapia de lenguaje, Masaje terapéutico, Fisioterapia acuática, Rehabilitación Física. Además de la atención a niños con Secuelas de lesiones neurológicas, parálisis cerebral, accidente cerebro vascular, traumatismo craneoencefálico, síndromes, lesiones traumatológicas y deportivas, estrés laboral.

5.- ¿Cuántas personas trabajan en la empresa / organización?

9 fisioterapeutas, 2 estimuladoras tempranas

6.- ¿Cuáles son los procesos centrales de la organización?

El proceso central de la empresa es el de la rehabilitación, en donde se encuentra inmerso la Neurorehabilitación, Rehabilitación Física, Fisioterapia acuática, Estimulación de lenguaje, Equinoterapia, etc.

7.- ¿Cuál es el principal problema a resolver?

Existen varias necesidades dentro del centro de rehabilitación, pero un caso especial es de un niño de 8 años llamado Benjamín, el cual tiene atrofia muscular espinal I. Los cuidadores de Benjamín no pueden estar las 24 horas del día junto a él ante una necesidad, ya que también el habla de Benjamín es escasa, es por esto que lo ideal sería que Benjamín en cierto modo se sienta útil al poder plasmar sus necesidades mediante un dispositivo que informe a los cuidadores sus necesidades.

8.- ¿Si se cuenta con el apoyo de la dirección de empresa / organización para la resolución del problema?

El centro de rehabilitación bendiciones está dispuesto a brindar el apoyo necesario para la solución y ayuda del paciente.

9.- ¿Qué resultados quisiera obtener con la implementación de la solución?

Que el paciente pueda tener en cierto sentido independencia en algunas cosas y que se sienta útil y entusiasmado al poder usar el dispositivo de reconocimiento de voz para informar a sus cuidadores las necesidades que él tenga.

## **ANÁLISIS E INTERPRETACIÓN DE RESULTADOS**

De acuerdo a lo manifestado por la directora del Centro de Rehabilitación Bendiciones, existe la necesidad de brindar una solución para los pacientes que presentan atrofia muscular espinal I, mediante la implementación de un dispositivo de reconocimiento de voz, como una ayuda para sentirse útiles dentro del hogar y la sociedad, y que la enfermedad no sea un obstáculo para seguir adelante.

Existe la predisposición de la directora del Centro de Rehabilitación, así como también de los padres del paciente para la implementación del dispositivo de reconocimiento de voz, ya que será de gran utilidad para poder conocer las necesidades que tiene el paciente.

Por los resultados obtenidos se puede mencionar que la implementación de este dispositivo resulta ser eficiente en lo que se refiere a su funcionalidad; evidenciándolo mediante el grado de aceptación y nivel de satisfacción puesta de manifiesto por parte del paciente y sus cuidadores.

## **CAPÍTULOS III**

### **PROPUESTA Y RESULTADOS ESPERADOS**

#### **METODOLOGÍA DE DESARROLLO**

##### **Selección de la Metodología de desarrollo**

Para la selección de la metodología de desarrollo se debe tener en cuenta que tipo de proyecto se está realizando, y de acuerdo a eso seleccionar la que más se adapte. Dado esto, en el presente capítulo se abarca el análisis, desarrollo y el funcionamiento del producto final, así también los parámetros que se necesitan.

Es necesario realizar un proyecto orientado al desarrollo de un dispositivo o prototipo de asistencia de voz para personas con discapacidad motriz. Es por esto que la metodología cumple un papel muy importante dentro del desarrollo del mismo.

Gracias al uso de una metodología se puede realizar todo el proceso de análisis y estudio que requiere la investigación a través de las diferentes técnicas, pautas y fases que componen una metodología para desarrollar de mejor manera un trabajo. Por otro lado, una metodología permite que las personas que desarrollan un proyecto sean más proactivas y ágiles en la ejecución de acciones, a la vez que ayuda a llevar una mejor planificación en general.

Existen varios tipos de metodologías de desarrollo, las más comunes son las tradicionales y las ágiles. Las metodologías tradicionales definen al inicio del proyecto todos los requerimientos de una manera muy rígida y absoluta en donde no se pueden hacer cambios posteriores. Las metodologías ágiles se caracterizan por su flexibilidad, permitiendo adaptar cambios y necesidades que surgen a lo largo del desarrollo del proyecto, es decir se añaden nuevas funcionalidades al producto final.

Dado esto, en el presente proyecto de acuerdo a las necesidades del desarrollo de un prototipo de asistencia de voz, se usa una metodología ágil, la metodología XP (Xtreme Programming).

Esta metodología se fundamenta principalmente en la recopilación de información y requerimientos brindados por el cliente mediante relaciones interpersonales que son la clave de éxito. XP permite que el producto que se vaya a desarrollar, en este caso el prototipo de asistencia de voz, sea de alta calidad y se adapte a los requisitos cambiantes que se pueden producir a lo largo de su desarrollo.

Xtreme Programming es una de las metodologías más viables ya que se puede dar seguimiento por partes de cómo se está desarrollando el producto. Además, consta con cuatro fases fundamentales que permiten seguir un proceso metódico y sistemático a lo largo del proyecto, brindando una mejor visión en cuanto a los pasos que se debe seguir.

Por otro lado, el desarrollo de este proyecto a través de la metodología en estudio tiene muchos beneficios, ya que primeramente impulsa el trabajo en equipo y las relaciones entre usuario y desarrollador. Esta metodología al ser flexible brinda mucho beneficio al proyecto ya que ayudará a que se puedan hacer cambios necesarios en el dispositivo de asistencia de voz cuando este lo requiera. Se caracteriza por permitir pruebas a lo largo del proyecto, en donde existe un ahorro de tiempo y por ende de economía al poder solucionar posibles problemas existentes a tiempo.

### **Planificación**

En la fase de planificación se recopila toda la información necesaria para la realización del proyecto. Se establecen los objetivos, arquitectura, características, componentes, tecnologías y lo más importante la historia del usuario para identificar los requerimientos del mismo en cuanto al producto final. Además, se identifica todas las entidades que se van a involucrar en el proyecto y así se delimita el alcance del mismo.

## **Diseño**

En la fase de diseño es en donde se intenta buscar la manera más sencilla de realizar el código para satisfacer los requerimientos dados en la historia del usuario. Es donde se incluye el diseño aplicativo tanto del dispositivo como del software y las herramientas para la respectiva programación. Además, si en esta fase se diseña correctamente, servirá como base para la siguiente fase.

## **Codificar**

Esta fase tiene como finalidad la funcionalidad del sistema, en donde se profundiza el proceso de programación. El desarrollo físico del prototipo y del software se inicia mediante los algoritmos a utilizar, permitiendo tener un código de forma colectiva, que sirva como base para investigaciones futuras. Se desarrolla el código fuente en general para el correcto funcionamiento del producto final, con posibles cambios a realizar.

## **Pruebas**

Luego de la codificación, se realizan pruebas de funcionamiento tanto del prototipo como del software para poder identificar posibles errores existentes dentro del producto final. Es aquí donde se solucionan los errores, en algún caso que los haya al momento de haber realizado las pruebas de aceptación. Además, se capacita al usuario y se provee el soporte técnico necesario. Y por último se verifica que el producto final cumpla con los requerimientos dados.

## **ANÁLISIS DEL SISTEMA ACTUAL**

El método de solicitar una necesidad por parte de los pacientes con discapacidad motriz, específicamente del paciente con atrofia muscular espinal se basa únicamente en que los cuidadores deben estar al lado del paciente la mayoría del tiempo para poder escuchar con claridad lo que él requiere.

Dado esto, se puede decir que en cierto modo no existe un método efectivo para solicitar una necesidad dado que el niño con discapacidad no tiene la voz suficientemente alta para que se le escuche a una distancia larga. Esto provoca que en ciertas ocasiones no se pueda atender las necesidades del niño debido a que los

cuidadores, en este caso su familia, no pueden estar las veinticuatro horas del día a su lado, imposibilitando satisfacer al cien por ciento las peticiones.

Por otro lado, el uso de tecnología asistiva dentro del hogar del paciente es escasa o inexistente. Dado esto, no les permite aprovechar las nuevas tecnologías y no constan con una herramienta que sea capaz de recibir las necesidades del paciente.

Por tanto, la acción de solicitar cierta necesidad es limitada. Esto da como resultado la importancia del desarrollo del dispositivo de asistencia de voz para el paciente con discapacidad motriz. Este dispositivo y el software servirán como una herramienta de comunicación entre cuidador y paciente ya que podrán estar en aviso ante cualquier petición que el mismo requiera.

## ESPECIFICACIÓN DE REQUERIMIENTOS

En la tabla 1 se presenta la especificación de requerimientos del proyecto.

**Tabla 1.** Requerimientos

<b>Componente</b>	<b>Descripción</b>	<b>Cantidad</b>
Dispositivo	Hardware en donde se realiza la programación y que permitirá reconocer las peticiones del paciente	1
Micrófono	Servirá para que el paciente pida la necesidad y reconozca su voz	1
Wifi	Módulo de comunicación entre el dispositivo de asistencia de voz y la aplicación móvil	1
Aplicación Móvil	Interfaz de visualización en donde llegarán las peticiones del paciente en forma de mensaje	1

**Elaborado por:** Kevin López

Un dispositivo de asistencia de voz basados en principios de programación permite en cierto grado la dependencia del paciente y la comunicación entre cuidador y paciente. Este dispositivo servirá como hardware en donde se realizará toda la programación necesaria para que el mismo pueda reconocer mediante un micrófono la voz del niño, que en este caso es el paciente. Además, tendrá varios componentes dentro de la programación que ayudarán a que el paciente pueda pedir sus necesidades mediante palabras claves.

El dispositivo de reconocimiento de voz debe ser de fácil uso y manipulación para que el paciente con discapacidad pueda solicitar las necesidades fácilmente. Además de ello de impulsar el uso de las nuevas tecnologías dentro de un área muy importante como es el de la salud, indicando que son de suma importancia hoy en día.

Por otro lado, el presente proyecto también consta con una aplicación móvil. Esta sirve como interfaz de visualización de las diferentes necesidades que el niño solicita a través del dispositivo de reconocimiento de voz y llegan a esta app en forma de mensaje.

Además, se tendrá una comunicación inalámbrica entre el dispositivo de asistencia de voz y la aplicación móvil. Esta permitirá que los dispositivos permanezcan conectados a la red sin usar algún cableado en específico, es decir mediante Wifi. Todos estos componentes están inmersos dentro del producto final para que se pueda cumplir el a cabalidad lo dicho anteriormente a través de la metodología en desarrollo.

## **FASE I: PLANIFICACIÓN**

### **PLAN DE DESARROLLO DE SOFTWARE**

#### **Visión General del Proyecto**

El propósito principal del presente proyecto es que los pacientes con discapacidad motriz se vean involucrados con herramientas tecnológicas que les ayuden a generar cierto grado de dependencia en cuanto a sus cuidadores. El dispositivo de asistencia de voz está diseñado con principios de ingeniería lo cual permitirá que

el paciente solicite una necesidad.

La utilización de un micrófono dentro del dispositivo permitirá que este pueda reconocer la voz y ejecutar una acción. Una aplicación móvil servirá como una herramienta para el cuidador. En esta se podrá visualizar las necesidades solicitadas por el paciente y a la vez enviar un mensaje al mismo.

### **Propuesta del Nuevo Sistema**

Desarrollar un dispositivo de asistencia de voz como herramienta de apoyo para pacientes con discapacidad motriz que permita solicitar necesidades es de gran importancia. Esto permitirá que los pacientes y específicamente el niño con atrofia muscular espinal se sienta útil y genere dependencia.

En la presente propuesta se ha seleccionado un dispositivo de gran importancia como es el M5StackCore2. Dentro de este se encuentran varios componentes que sirven para el desarrollo de un asistente de voz mediante la programación. Este fue adquirido desde una empresa de los Estados Unidos de América, dando a notar que no es un producto de fácil acceso.

El diseño de una aplicación móvil a través de app inventor también es parte de la propuesta. App inventor es una herramienta que encaja perfectamente con la metodología escogida. Debido a esto, utilizar un lenguaje esquematizado de bloques, permite el entendimiento y comprensión de cómo funcionan los algoritmos de procesos. Dicho esto, mediante la programación se puede obtener una aplicación móvil funcional e intuitiva como producto final.

### **Análisis del Sistema Propuesto**

En la tabla 2 se detalla las herramientas que se van a usar en la propuesta.

**Tabla 2:** Herramientas para el desarrollo de la propuesta

<b>ID DE DESARROLLO APP</b>	App Inventor
<b>BASE DE DATOS</b>	Firestore
<b>ID DE PROGRAMACIÓN (M5Stack)</b>	Visual Studio Code
<b>LENGUAJE DE PROGRAMACION (M5Stack)</b>	MicroPython

<b>APLICACIÓN MÓVIL</b>	Android
-------------------------	---------

**Elaborado por:** Kevin López

En la tabla 3 se presenta la historia del usuario que va a controlar la aplicación móvil.

**Tabla 3:** Historia de usuario dispositivo de voz

Historia de Usuario	
Numero: 1	Usuario: Paciente (niño)
Nombre Historia: <b>Control dispositivo de reconocimiento de voz</b>	
Prioridad: Alta	Riesgo en desarrollo: Media
Puntos Estimados:1	Interacción Asignada:1
Programador Responsable: Kevin López	
Descripción: El usuario que podrá usar este dispositivo de asistencia de voz será exclusivamente el paciente, es decir el niño. Él tendrá la capacidad de solicitar una necesidad a través de este dispositivo programado para reconocer su voz.	
Observaciones: No se requiere de un inicio de sesión.	

**Elaborado por:** Kevin López

En la tabla 4 se presenta la historia del usuario que va a controlar la aplicación móvil.

**Tabla 4:** Historia de usuario aplicación móvil

Historia de Usuario	
Numero: 2	Usuario: Cuidador
Nombre Historia: <b>Control aplicación móvil</b>	
Prioridad: Alta	Riesgo en desarrollo: Media
Puntos Estimados:1	Interacción Asignada:1

Programador Responsable: Kevin López
Descripción: Los usuarios que podrán acceder al uso de la aplicación móvil, serán únicamente los cuidadores del paciente, ya que esta app será exclusivamente implementada en el lugar en donde se realice las pruebas.
Observaciones: No se requiere de un inicio de sesión.

**Elaborado por:** Kevin López

A la persona que cuida al paciente se le implementará la aplicación móvil en su teléfono celular para que de esa manera pueda mantenerse comunicada en cuanto a las peticiones del paciente.

### **Organización del Proyecto**

La tabla 5 presenta los cargos que se deben desempeñar para que la propuesta planteada se realice exitosamente.

**Tabla 5:** Roles y Responsabilidades

<b>Cargo</b>	<b>Funciones</b>
Jefe de Proyecto Kevin López	Realiza la recopilación de información y la adquisición de los implementos necesarios para el diseño del sistema Se encarga de hacer cumplir con los plazos propuestos para el sistema Asigna tareas específicas para cada uno de los participantes del sistema
Analista de Sistemas Kevin López	Realiza el estudio de los requerimientos de los procesos Identifica las restricciones que presenta el sistema Diseña la aplicación móvil
Programador Kevin López	Desarrolla la codificación para el dispositivo de asistencia de voz Realiza la programación estructurada en bloques para la aplicación móvil Se encarga de hacer la conexión entre el dispositivo de asistencia de voz y la aplicación móvil

**Elaborado por:** Kevin López

## Plan de Fases

En la tabla 6 se muestra las fases a realizar en el proyecto.

**Tabla 6:** Plan de Fases

Planeación del proyecto	Recopilación de información Cronograma de actividades Análisis de restricciones Estudio de factibilidad
Diseño	Diseño de interfaz del dispositivo de asistencia de voz Diseño del funcionamiento de la base de datos Diseño de la interfaz de la aplicación móvil Diseño del Diagrama de Clase
Codificación	Programación del dispositivo (M5Stack) Programación de la Aplicación Móvil Conexión del dispositivo y la aplicación móvil
Pruebas e implementación	Pruebas de unidad Pruebas de integración Pruebas de validación

## PLANEACIÓN DEL PROYECTO

Se muestra el cronograma de actividades del presente proyecto en la tabla 7.

**Tabla 7:** Cronograma de Actividades

Actividades	MESES 2021 – 2022					
	Oct	Nov	Dic	Ene	Feb	Mar
Formulación del tema	X					
Recopilación de material bibliográfico	X					
Desarrollo Capítulo I		x				
Redacción Objetivos del Proyecto		x				
Revisión Capítulo I		x				

Desarrollo Capitulo II			X			
Recopilación de información mediante entrevista			X			
Revisión del Capítulo II			X			
Interpretación de resultados de la entrevista			X	x		
Desarrollo Capitulo III				x		
Selección de Metodología por utilizar en el proyecto				x		
Análisis de Requerimientos				x		
Análisis del Sistema Por Implementar					x	
Diseño de la interfaz del dispositivo de reconocimiento					x	
Diseño de la interfaz de la aplicación móvil					x	
Codificación del dispositivo					x	
Codificación de la aplicación móvil					x	
Pruebas funcionales						x
Verificación de Resultados						x

**Elaborado por:** Kevin López

### **ANÁLISIS DE RESTRICCIONES**

1. El dispositivo de asistencia de voz es a carga, batería por lo que cada determinado tiempo deberá ser conectado para que se pueda almacenar nuevamente energía.
2. La aplicación móvil y el dispositivo de asistencia de voz deben estar conectados inalámbricamente mediante Wifi.

3. Para el mantenimiento preventivo y correctivo del dispositivo de asistencia de voz se necesita de personal que esté capacitado y conozca el funcionamiento total del mismo.
4. Las actualizaciones de la aplicación móvil se realizarán cada vez que el cuidador disponga de actualizaciones nuevas.

## **ESTUDIO DE FACTIBILIDAD**

### **Factibilidad Operativa**

El presente proyecto es operativamente factible ya que se tienen todos los conocimientos y capacidades necesarias para el desarrollo del mismo. El dispositivo de asistencia de voz y la aplicación móvil serán implementados en el hogar del paciente. Para el desarrollo de estos, es necesario tener conocimientos de ingeniería principalmente de programación, diseño de interfaces, base de datos y programación estructurada en bloques para la aplicación móvil.

Los usuarios serán capacitados para que puedan hacer un uso correcto tanto de la aplicación móvil como del dispositivo de asistencia de voz. Dado esto, el nivel de aceptación del producto final por parte del paciente y del cuidador, así como por la directora del centro de rehabilitación es satisfactoria.

### **Factibilidad Técnica**

#### **Recursos Tecnológicos Disponibles**

En la tabla 8 se muestra los recursos tecnológicos disponibles en el hogar del paciente.

**Tabla 8:** Herramientas existentes

Conexión a internet (Wifi)	Equipos portátiles (PC) Teléfonos móviles
----------------------------	--

**Elaborado por:** Kevin López

## Recurso Tecnológico Necesario

En la tabla 9 se plasman los recursos tecnológicos necesarios para el correcto funcionamiento del sistema propuesto.

**Tabla 9:** Recursos tecnológicos requeridos

Componentes que se requiere para el funcionamiento del sistema	
Hardware	Software
<ul style="list-style-type: none"><li>• Se requiere un dispositivo móvil (Funcionabilidad completa)</li><li>• Se requiere de un dispositivo de reconocimiento de voz</li><li>• Micrófono</li></ul>	<ul style="list-style-type: none"><li>• Sistema Operativo Android</li><li>• Conexión a Internet (Wifi)</li></ul>

**Elaborado por:** Kevin López

## Factibilidad Económica

En la figura 3 y figura 4 se muestra el valor obtenido por el módulo COCOMO II

SLOC Input Dialog - KitRobotSTEAM

Sizing Method:  SLOC  Function Points  Adaptation and Reuse

Breakage: % of code thrown away due to requirements evolution and volatility. REVL: 0.00

Module Size in Function Points: Language: JAVA, Change Multiplier: 53

Ratio Type:  Jones  David

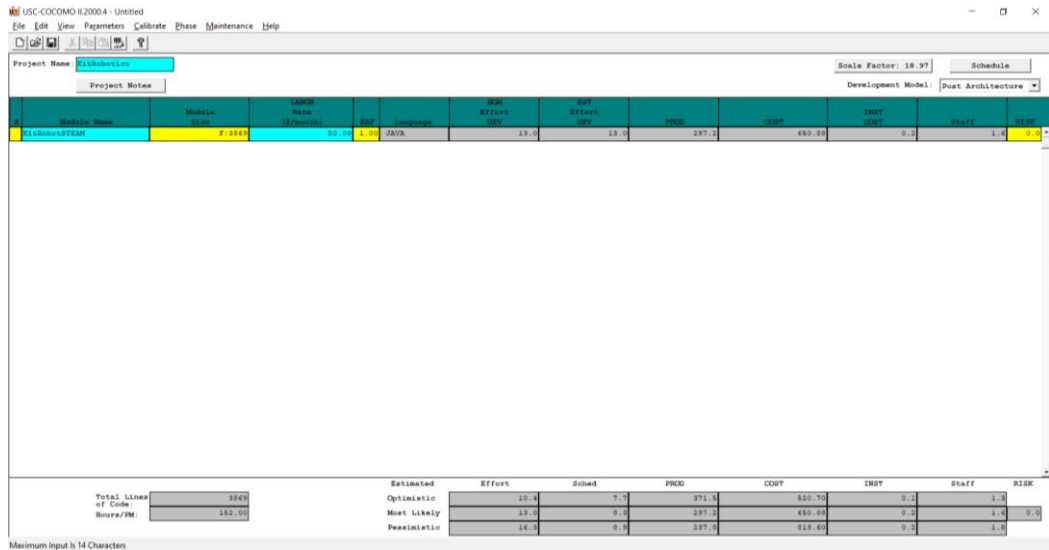
Calculation Method:  Using Table  Input Calculated Function Point

Function Type	# of Function Points			SubTotal
	Low	Average	High	
Inputs	4	4	0	28
Outputs	0	6	0	30
Files	0	0	1	15
Interfaces	0	0	0	0
Queries	0	0	0	0
Total Unadjusted Function Points				73
Equivalent Total in SLOC				3869

Buttons: OK, Cancel, Help

**Figura 3:** Valor del módulo de administrador COCOMO II.

**Elaborado por:** Kevin López



**Figura 4:** Valor del proyecto COCOMO II.

**Elaborado por:** Kevin López

### Costos Directos

Se detallan los costos directos a realizarse en la tabla 10.

**Tabla 10:** Costos Directos

Rubro	Valor
Dispositivo M5stack	\$ 88.50
Desarrollo de Aplicación móvil	\$ 120
Total	\$ 208.50

**Elaborado por:** Kevin López

Se detallan los costos indirectos a realizarse en la tabla 10.

**Tabla 11:** Costos Indirectos

Rubro	Valor
Envío del dispositivo	\$15
Movilización	\$20
Total	\$ 35

**Elaborado por:** Kevin López

Costo total de proyecto= costos directos + costos indirectos.

Costo total= \$ 243.50

La suma total de los costos directos como indirectos es 243.50 \$. Se realiza la comparativa con el costo un dispositivo de asistencia de voz como es el caso de Alexa. Además, estos son los valores de adquisición de cada componente, siendo los desarrolladores adquieran los mismos, por lo tanto, es económicamente factible de realizarse.

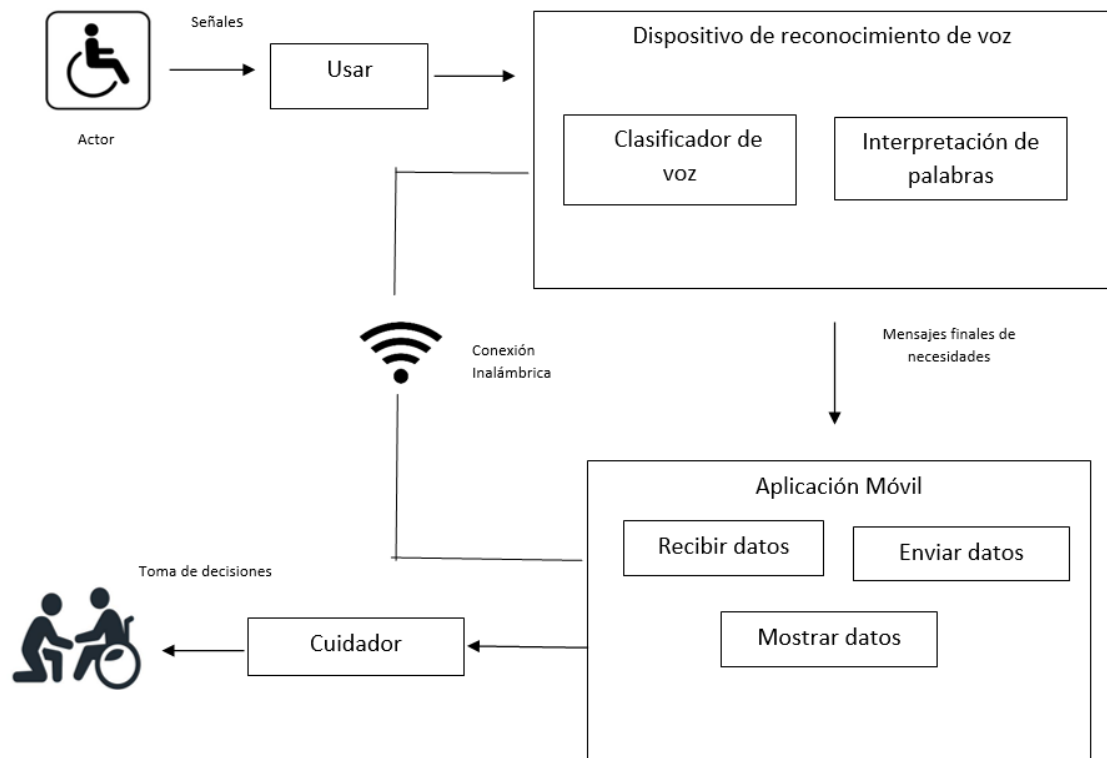
### **ANÁLISIS COSTO BENEFICIO**

Como referencia se toma a un dispositivo de asistencia de voz con características funcionales similares. Alexa desarrollado por Amazon es un asistente virtual que se controla mediante la voz con varias funcionalidades, con el que se puede realizar una gran variedad de peticiones. El precio de adquisición de Alexa supera los \$100 aproximadamente en mercados internacionales, Dado esto se pone a disposición un dispositivo de similares características con el fin de adentrarnos al campo de la salud y ayudar a pacientes con discapacidad motriz.

### **ANÁLISIS ORIENTADO A OBJETOS**

#### **Diagramas de Casos de Uso**

La figura 5 muestra el funcionamiento del presente sistema.



**Figura 5:** Diagrama de Caso de Uso

**Elaborado por:** Kevin López

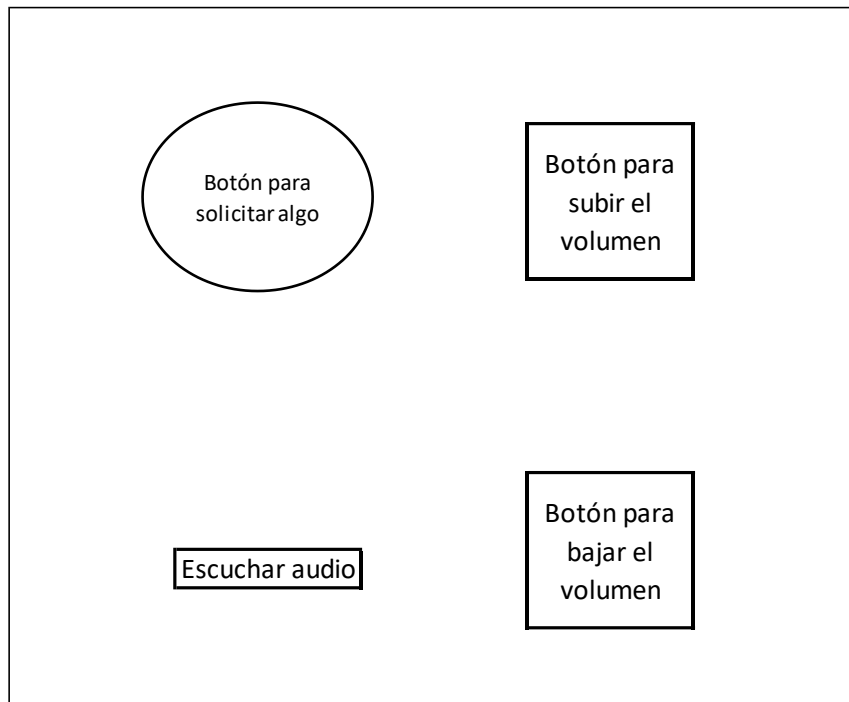
## **Fase II: DISEÑO DE LA PROPUESTA.**

### **Diseño De La Interfaz De Usuario**

A continuación, se presenta un diseño preliminar de la interfaz de uso que llevará tanto el dispositivo de asistencia de voz como la aplicación móvil para el control de todos los componentes. El dispositivo permitirá que el paciente solicite necesidades y ejecute acciones. Es una interfaz de inicio y una interfaz principal dividida en diferentes secciones.

En la primera sección, el paciente podrá solicitar una necesidad mediante su voz. Esta puede ser mediante un icono o un botón que permita realizar esta acción. Además, se tendrá dos botones de volumen que le permita al paciente subir y bajar la intensidad de lo que está escuchando. Si el cuidador necesita algún otro requerimiento conociendo la necesidad del paciente, posteriormente se lo estará implementando. En la figura 6 se muestra un diseño preliminar de la interfaz del

dispositivo de reconocimiento de voz.

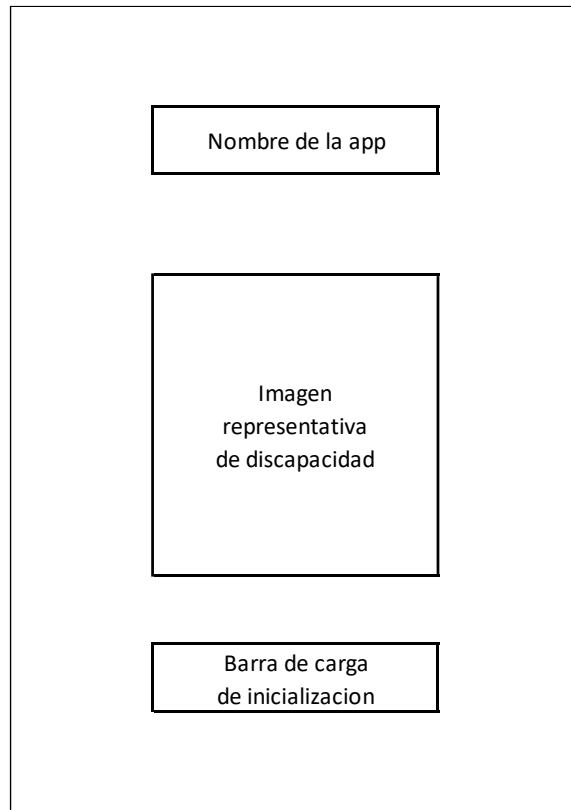


**Figura 6:** Interfaz preliminar M5Stack.

**Elaborado por:** Kevin López

Por otro lado, la aplicación móvil tendrá una interfaz que se dividirá en dos secciones. En la primera sección se encuentra la pantalla de inicio en donde únicamente hay información del nombre de la app. En la segunda sección habrá un botón para poder enviar un mensaje mediante audio al dispositivo de asistencia. También, se encuentra una opción en donde llegarán todos los mensajes de las solicitudes hechas por parte del paciente.

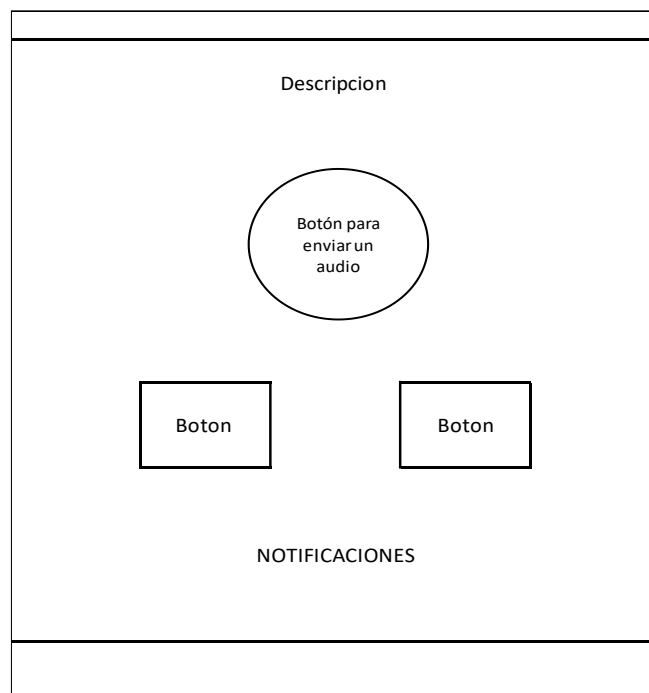
En la figura 7 se muestra un diseño preliminar de la pantalla de inicio de la aplicación móvil.



**Figura 7:** Interfaz de inicio de la app.

**Elaborado por:** Kevin López

En la Figura 8 se presenta un diseño preliminar de la pantalla principal de la app.

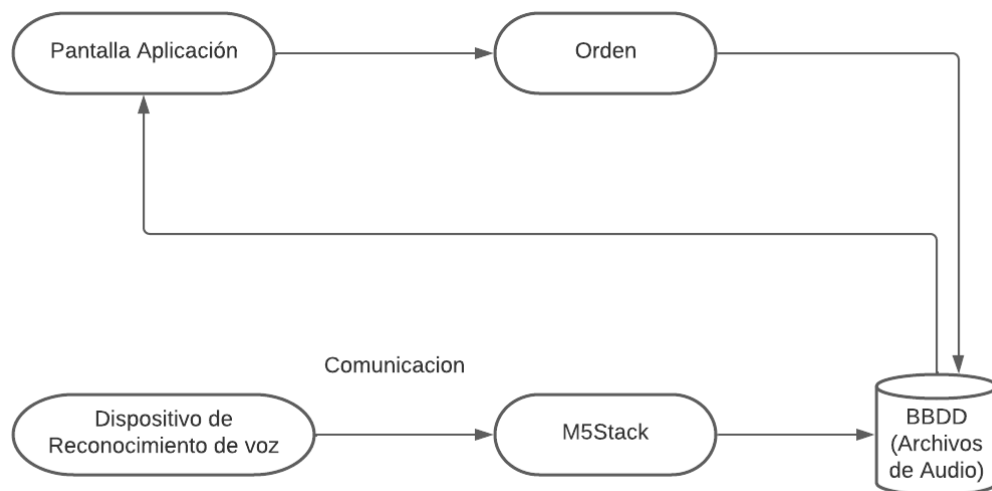


**Figura 8:** Interfaz segunda pantalla app.

**Elaborado por:** Kevin López

**Diagrama de uso de Base de Datos.**

En la siguiente figura 9 se muestra como es la interacción entre el dispositivo de asistencia de voz y la aplicación móvil, así como la incidencia que tienen los archivos de audio como base de datos que almacena toda la información necesaria.



**Figura 9:** Diagrama Interacción base de datos

**Elaborado por:** Kevin López

**Materiales**

**M5Stack**

M5Stack es un robusto kit de desarrollo de código abierto con módulos apilables y un IDE fácil de usar, que permite crear prototipos rápidos y de alta calidad. El objetivo de M5Stack es hacer que los prototipos no solo sean útiles, sino también hermosos gracias a su elegante carcasa de aluminio. El sistema consiste en un módulo central con una pantalla LCD que puede apilarse sobre una plétora de placas de expansión, entre las que se incluyen osciloscopios, sensores ambientales, PLC, GPS, baterías y mucho más. La plataforma es completamente

de código abierto y es compatible con Wi-Fi, LoRa y Arduino. Cuando se combinan, estos módulos pueden crear una amplia gama de productos, desde enchufes e interruptores hasta multímetros y controladores de automatización.

Los microprocesadores ESP32 de Espressif han ganado rápidamente popularidad y ya se pueden encontrar siendo utilizados en muchos proyectos de IoT que requieren conectividad Wi-Fi o BLE. Aunque estos procesadores tienen mucha potencia, a menudo es difícil programarlos en el entorno nativo para los principiantes y los entusiastas del IoT. Para hacer frente a este problema y acelerar el desarrollo del IoT, M5 Stack ha presentado su nuevo kit de desarrollo M5Stack Core2, un módulo de desarrollo basado en ESP32 con muchas funciones que le permite crear un prototipo de sus ideas de IoT nada más sacarlo de la caja. Este kit de desarrollo tiene una pantalla táctil capacitiva de 2 pulgadas integrada, una batería incorporada y muchos otros sensores y módulos interesantes. Y además de todo esto, se puede programar fácilmente con Arduino IDE o micro python.

M5Stack se centra en la creación de kits de desarrollo de IoT modulares y apilables todo en uno, basados en ESP32. M5Stack ha desarrollado una marca de boca en boca en el espacio de la placa de desarrollo en el mundo en los últimos años. Sus productos son queridos por la mayoría de los aficionados en Japón y se venden en más de 100 países como Japón, Estados Unidos, Reino Unido, Alemania, Australia, Bélgica, etc. Sus productos han empleado varios escenarios de aplicación como Smart Home, Smart Office, STEM Education, AI, Robotics, Industry4.0, y etc.

### **Vistazo más de cerca**

Mirando de cerca el kit, podemos ver que tiene una forma cuadrada limpia con una pantalla en la parte superior y botones y ranuras en el lado.

La pantalla, es una pantalla táctil capacitiva de 2 pulgadas con una resolución de 300x240 píxeles. Justo debajo de la pantalla, también se pueden observar tres botones táctiles capacitivos que se pueden programar para que funcionen según nuestras necesidades. Tenemos el botón de encendido, un puerto de interfaz USB tipo C para cargar y programar el dispositivo, y un conector de interfaz de ranura

que puede utilizar para conectar otros sensores y módulos si es necesario. Seguidamente, en la parte inferior, podemos ver un botón de reset, un LED verde como indicador de encendido, y una ranura para tarjetas SD que puede soportar tarjetas de hasta 16G.

La placa se vuelve más interesante cuando echamos un vistazo a la parte trasera. La pegatina de la parte trasera contiene una breve explicación de las características y especificaciones de los circuitos integrados utilizados en esta placa. Veámoslo en la imagen siguiente.

El cerebro detrás del kit de desarrollo es el microprocesador ESP32 D0WDQ6 y un chipset Xtensa de doble núcleo y 32 bits que funciona a 240Mhz con 16MB Flash y 8MB PSRAM. Y ni que decir tiene que el ESP32 soporta tanto el protocolo Wi-Fi como el Bluetooth (BLE). A la derecha, podemos ver a qué pines está conectada la pantalla y el nombre del IC del controlador de la pantalla que es ILI9342, luego podemos ver la marca para el botón de encendido, si lo mantenemos durante 6 segundos, el dispositivo se apagará. Luego tenemos el IC BM8563 RTC y a continuación el conector USB tipo C conectado al IC controlador USB CP2104 y el IC de gestión de energía AXP192, que controla la carga de nuestra batería y también regula los 3,3V necesarios para la placa. Continuando, tenemos el IC convertidor DC/DC SY7088, que se utiliza para configurar el voltaje de la batería a 5V.

A continuación, tenemos el IC amplificador NS4168 I2C que se conecta a un altavoz incorporado para reproducir audio. Luego, la placa de expansión a la izquierda, la imagen de abajo muestra la placa de expansión desmontada de la placa principal. La placa de expansión consiste en un micrófono incorporado y un sensor MPU886 de 6 ejes IMU. Una vez que se retira la placa de cabecera, los pines de cabecera de la placa principal quedarán expuestos y podrán utilizarse para la interconexión con otros módulos. La definición de los pines de la cabecera se menciona en el propio Stricker.

## **Especificaciones del hardware del M5Stack Core2**

Hay que comprobar lo que tiene adentro para echar un vistazo a nuestro hardware. A continuación, en la tabla 12 se indican las especificaciones técnicas y componentes completos del kit de desarrollo.

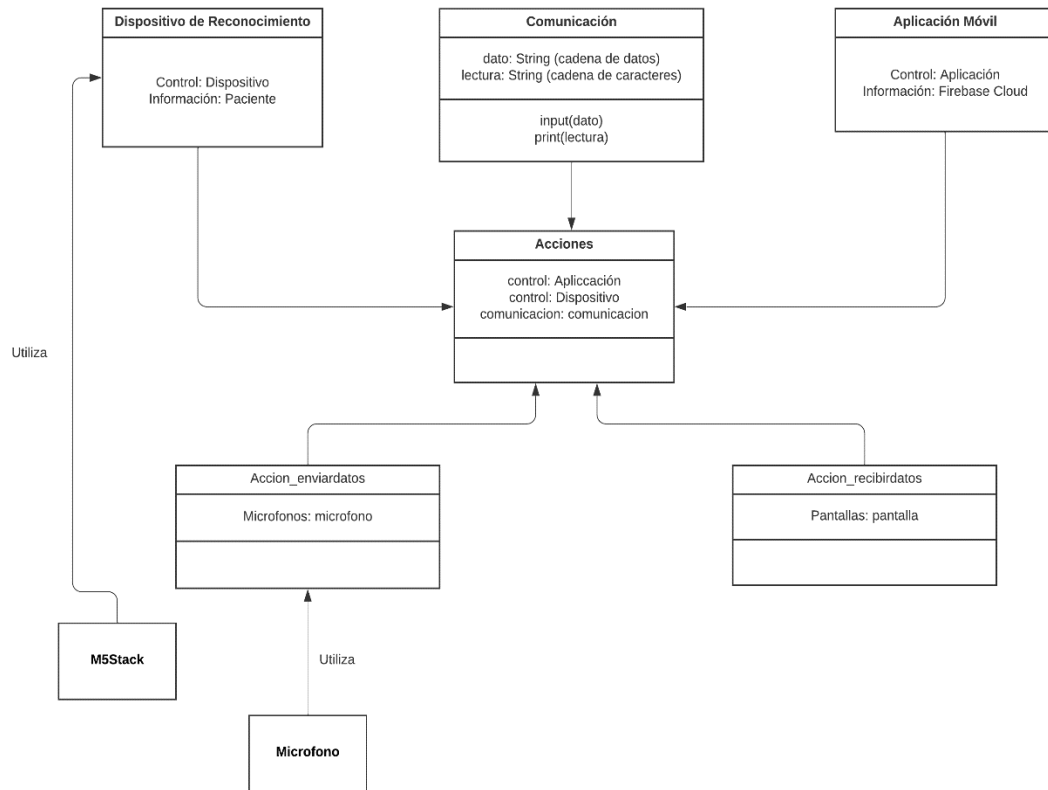
**Tabla 12:** Componentes del M5Stack

<b>Componente</b>	<b>Característica</b>
ESP32-D0WD-V3	240 MHz dual core, 600 DMIPS, 520 kb SRAM, Wi-Fi, dual mode Bluetooth
Flash	16 MB
PSRAM	8 MB
Voltaje de Entrada	5V @ 500mA
Interfaz	Type C x 1, Grove ( I2C+I/O+UART) x 1
Pantalla LCD IPS	2.0" @ 320*240 ILI9342C
Pantalla táctil	FT6336U
Altavoz	1W-0928
LED	Indicador luminoso verde de alimentación
Botón	Botón de encendido, botón RST, botón de pantalla virtual
Recordatorio de vibración	Motor de vibración
Micrófono	SPM1423
Amplificador de potencia I2C	NS4168
Tamaño del producto	54 x 54 x 16mm

**Elaborado por:** Kevin López

## DIAGRAMA DE CLASES

En la figura 10 se muestra diagrama de clases que compete a las acciones que realizará tanto el dispositivo de reconocimiento de voz y la aplicación, así también a que acción corresponde cada componente en general



**Figura 10:** Diagrama de Clase.

**Elaborado por:** Kevin López

## FASE III: CODIFICACIÓN

### M5StackCore2

#### Comandos de Voz

En el presente proyecto se incluye un conjunto de datos de audio de palabras habladas diseñado para ayudar a entrenar y evaluar los sistemas de detección de palabras clave. Su objetivo principal es proporcionar una forma de construir y probar pequeños modelos que detecten cuando se pronuncia una sola palabra, a

partir de un conjunto de palabras objetivo, con la menor cantidad posible de falsos positivos del ruido de fondo o del habla no relacionada.

Hay que tener en cuenta que, en el conjunto de entrenamiento y validación, la etiqueta "desconocido" prevalece mucho más que las etiquetas de las palabras objetivo o el ruido de fondo. Una diferencia con la versión de lanzamiento es el manejo de segmentos silenciosos. Mientras que en el conjunto de prueba los segmentos de silencio son archivos normales de 1 segundo, en el entrenamiento se proporcionan como segmentos largos. Seguidamente, se dividen estos ruidos de fondo en clips de 1 segundo y también guardamos uno de los archivos para el conjunto de validación.

### **Datos de Entrenamiento**

En primer lugar, es necesario encontrar algunos datos con los cuales entrenar un modelo. En este proyecto se tiene un conjunto de datos almacenados en carpetas con varios archivos de audio por cada palabra que consisten en un conjunto de 9 palabras de comandos principales como "Comida", "Baño", "Ventana", "Ayuda" y un conjunto de palabras adicionales. Cada una de las muestras tiene una duración de 1 a 2 segundos de distintas personas.

Con los datos de entrenamiento en su lugar, debemos pensar en qué características vamos a entrenar nuestra red neuronal. Es poco probable que introducir una forma de onda de audio sin procesar en nuestra red neuronal nos dé buenos resultados.

Necesitamos convertir nuestras muestras de audio en algo que parezca una imagen; para hacer esto, podemos tomar un espectrograma. Para obtener un espectrograma de una muestra de audio, dividimos la muestra en pequeñas secciones y luego realizamos una transformada de Fourier discreta en cada sección. Esto nos dará las frecuencias que están presentes en ese segmento de audio. Juntar estos cortes de frecuencia nos da un espectrograma como muestra. Todo esto se puede evidenciar en el Anexo 4 y 5.

La siguiente función que se muestra en la figura 11 se puede utilizar para generar un espectrograma a partir de una muestra de audio:

```

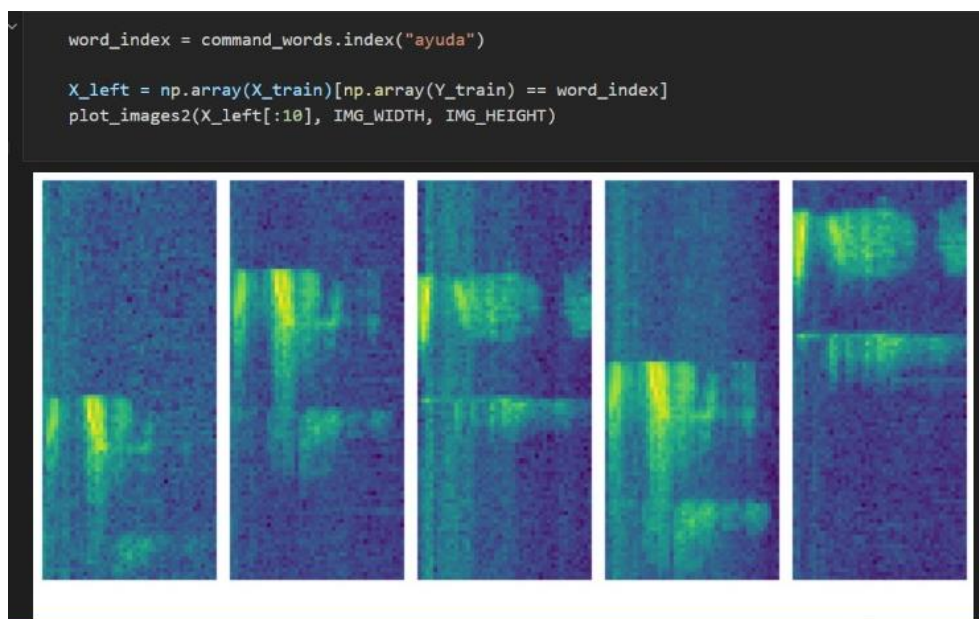
def get_spectrogram(audio):
    # normalise the audio
    audio = audio - np.mean(audio)
    audio = audio / np.max(np.abs(audio))
    # create the spectrogram
    spectrogram = audio_ops.audio_spectrogram(audio,
                                              window_size=320,
                                              stride=160,
                                              magnitude_squared=True).numpy()
    # reduce the number of frequency bins in our spectrogram to a more sensible level
    spectrogram = tf.nn.pool(
        input=tf.expand_dims(spectrogram, -1),
        window_shape=[1, 6],
        strides=[1, 6],
        pooling_type='AVG',
        padding='SAME')
    spectrogram = tf.squeeze(spectrogram, axis=0)
    spectrogram = np.log10(spectrogram + 1e-6)
    return spectrogram

```

**Figura 11:** Función para generar un espectrograma

**Elaborado por:** Kevin López

Esta función primero normaliza la muestra de audio para eliminar cualquier variación en el volumen de nuestras muestras. Luego calcula el espectrograma: hay una gran cantidad de datos en el espectrograma, por lo que reducimos esto aplicando la agrupación promedio. En la figura 12 se muestra un ejemplo del espectrograma generado de una palabra.



**Figura 12:** Espectrograma de una palabra

**Elaborado por:** Kevin López

Finalmente, tomamos el registro del espectrograma para no introducir valores extremos en nuestra red neuronal, lo que podría dificultar el entrenamiento.

### **Modelo Red Neuronal**

Una vez que se analice los audios de la base de datos, es necesario crear el modelo de la red neuronal. Con nuestros datos de entrenamiento en su lugar, debemos pensar en qué características vamos a entrenar nuestra red neuronal. Es poco probable que introducir una forma de onda de audio sin procesar en nuestra red neuronal nos dé buenos resultados. Se puede evidenciar en el Anexo 5 la codificación para la creación del modelo de la red neuronal.

### **Conversión del modelo a TensorFlow Lite**

Una vez que el modelo sea entrenado, ahora debemos convertirlo para usarlo en TensorFlow Lite. Este proceso de conversión toma nuestro modelo completo y lo convierte en una versión mucho más compacta que se puede ejecutar de manera eficiente en nuestro microcontrolador.

Dentro del proyecto existe un cuaderno que pasa nuestro modelo entrenado TFLiteConverter junto con ejemplos de datos de entrada. Proporcionar los datos de entrada de la muestra permite que el convertidor cuantifique nuestro modelo con precisión.

Una vez que el modelo se ha convertido, podemos ejecutar una herramienta de línea de comandos para generar código C que podemos compilar en nuestro proyecto, tal y como se muestra en la figura 13.

```
!xxd -i converted_model.tflite > model_data.cc
```

**Figura 13:** Comando para convertir el modelo tflite en código c

**Elaborado por:** Kevin López

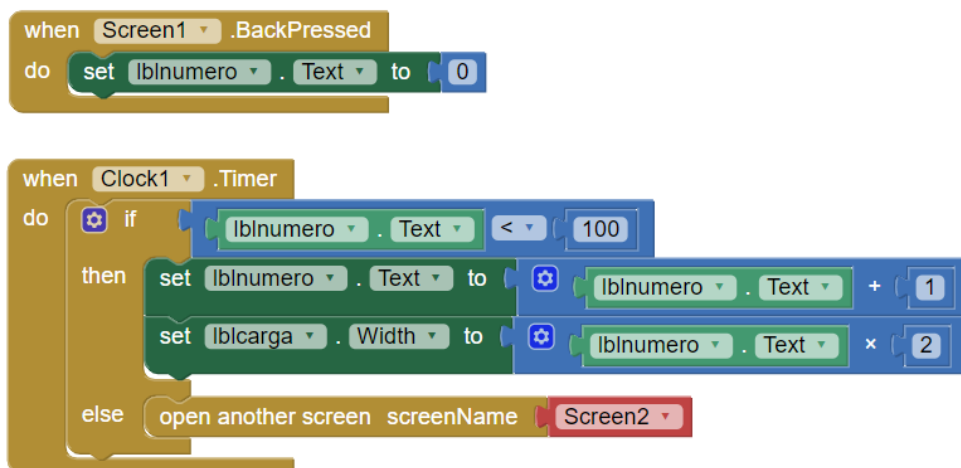
## Aplicación Móvil

### Diagrama de Bloques

#### Pantalla de Inicio

Cuando se realiza un previo diseño seguidamente se procede a la codificación. En este caso se realiza la pantalla de inicio la cual consta con el nombre de la aplicación, una imagen representativa a la definición de discapacidad y una barra de carga. Para el título del nombre de la aplicación, barra de carga, número de carga se utilizará etiquetas de texto. Cabe recalcar que cada label estará nombrado de acuerdo con la acción que realice.

El lblnumero se encargará de ejecutar la barra de carga que consta también con un reloj, es decir una duración de carga. Cuando el número sea menor a 100, este se va ir aumentando en 1 y luego se multiplicará x2. De esa manera, cuando lblnumero llegue a 100 se abrirá la siguiente pantalla. Todo esto se muestra en la figura 14.



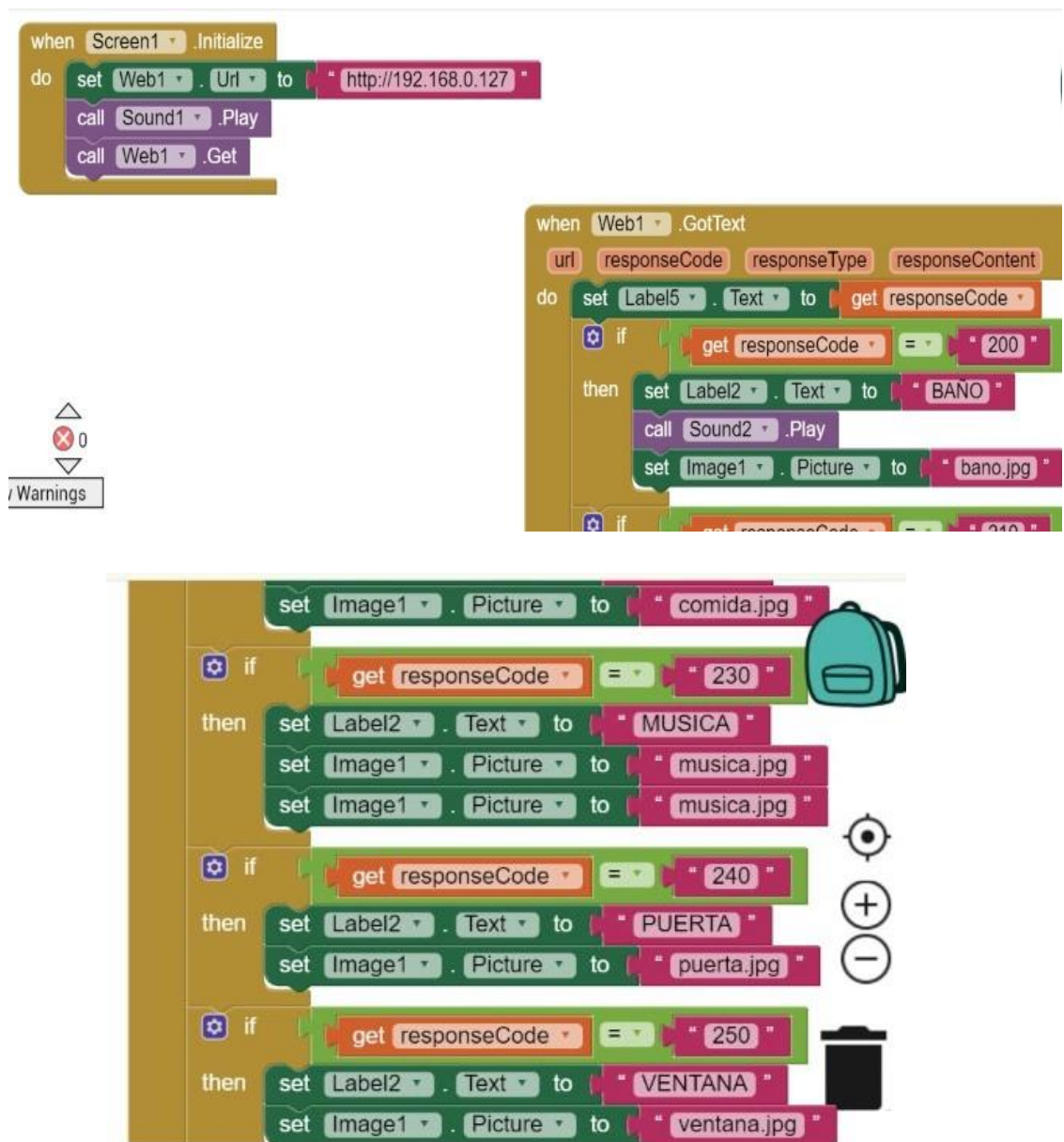
**Figura 14:** Codificación pantalla de inicio

**Elaborado por:** Kevin López

## Pantalla Principal

En esta pantalla en la parte superior se encuentra el logo de la universidad. Además, se encuentra un texto como notificación que muestra la necesidad solicitada. Por ejemplo, si se reconoció a través del dispositivo la palabra “agua”, en esta pantalla se podrá visualizar un mensaje como este “El paciente está solicitando agua”. También existe un botón de conexión.

Cabe recalcar que toda esta interfaz fue realizada con label, imágenes y diferentes funciones que ayudaron para el funcionamiento eficaz. Todo el diagrama de bloques se muestra en la figura 15.





**Figura 15:** Codificación pantalla principal

**Elaborado por:** Kevin López

## **FASE IV: PRUEBAS**

### **Pruebas de Unidad**

Para realizar las pruebas de unidad, se procedió a hacerlo con cada uno de los componentes utilizados. Se verificó su correcto funcionamiento y la eficiencia al responder a los requerimientos planteados en el proyecto.

Para la programación del dispositivo M5stack se usó el software Visual Studio Code. En este se fue comprobando el funcionamiento individual de cada función y así también la asignación de los identificadores que poseen cada uno para su uso.

Lo primero que vamos a necesitar es algún tipo de sistema de detección de palabras de activación. Esto escuchará continuamente el audio, esperando una frase o palabra desencadenante.

Cuando escuche la palabra, activará el resto del sistema para capturar las instrucciones que tenga el usuario. Una vez que se haya capturado el audio, lo

enviará a un servidor para que lo reconozca. El servidor procesa el audio y resuelve lo que pide el usuario.

### **Detección de palabras de activación**

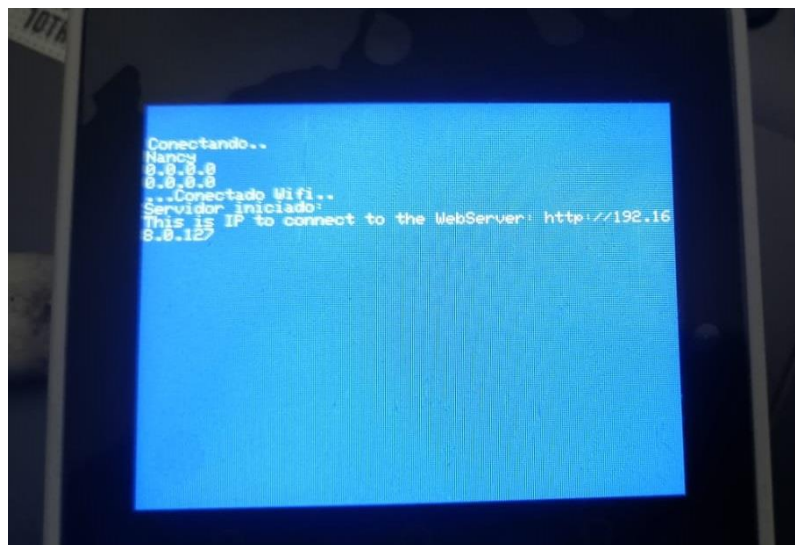
Para esto se necesita crear algo que indique el uso cuando el sistema escuche una palabra en específico.

Esto deberá ejecutarse en nuestros dispositivos integrados; una opción ideal para esto es usar TensorFlow y TensorFlow Lite

### **Pruebas de Integración**

Una vez que la prueba de unidad se haya culminado con éxito, es aquí donde se inicia la prueba de integración. Se procede a conectar la aplicación móvil con la M5Stack. Se comprueba que la conexión sea exitosa.

La conexión mediante conexión Wifi se la muestra en la figura 16 y en el Anexo 10.



**Figura 16:** Conexión Wifi

**Elaborado por:** Kevin López

### **Pruebas de Validación**

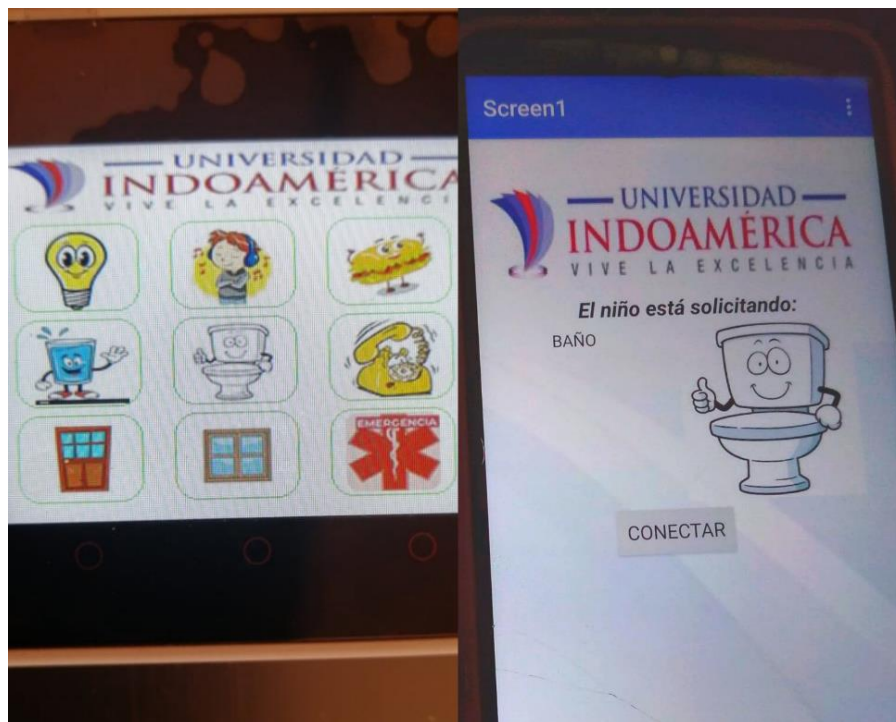
Las pruebas de validación se dieron una vez que se integró la aplicación móvil con el dispositivo de reconocimiento de voz, M5stack. Es aquí donde se

comprueba que el funcionamiento tanto del dispositivo como de la aplicación móvil sea el correcto.

Se realiza una prueba previa en laboratorio, es decir con una persona que no sea el usuario final, para de esa manera comprobar las diferentes funcionalidades, acciones, procesos que debe realizar el dispositivo de reconocimiento de voz y la aplicación móvil. Esto se puede evidenciar en el Anexo 11

### **Puesta En Marcha**

Luego que se realizaron todas las pruebas anteriores, se presentan los resultados funcionales en cuanto a la interfaz final de la M5stack y la aplicación móvil. También se verifica el correcto funcionamiento con el usuario. Todo esto se puede verificar en la figura 17 y el anexo 12.



**Figura 17:** Funcionamiento final

**Elaborado por:** Kevin López

## Lista de Validación

En la tabla 13 se muestra la lista de todas las acciones y módulos que componen el sistema en general del presente proyecto.

**Tabla 13:** Lista de Validación

<b>ID</b>	<b>Requerimiento</b>	<b>Estado</b>
1	Interfaz del M5Stack	Funciona
2	Reconocimiento de voz	Funciona
3	Conexión Wifi	Funciona
4	Interfaz aplicación móvil	Funciona
5	Recepción de palabras en la app	Funciona

**Elaborado por:** Kevin López

## RESULTADOS ESPERADOS

Luego de haber desarrollado la propuesta planteada mediante la metodología de investigación, se espera que los actores y los participantes que hacen parte del presente proyecto puedan desarrollar nuevas habilidades. Estas habilidades permiten ser más competentes e investigativos para que de esa manera los participantes puedan buscar soluciones a las diferentes problemáticas y puedan dar respuesta a las interrogantes existentes en la actualidad.

Como resultado se presenta un dispositivo capaz de reconocer la voz de una persona con discapacidad motriz y usuarios en general. Este prototipo de dispositivo de asistencia de voz tiene la capacidad de procesar la palabra que mencione la persona. El dispositivo realiza con éxito el reconocimiento de voz ya que las palabras que se pueden solicitar o reconocer tuvieron un previo

entrenamiento mediante audios pregrabados en la base de datos. Cabe recalcar que existe otra forma de solicitar una necesidad, esta es a través de la pantalla del dispositivo. Como la interfaz del dispositivo cuenta con unos gráficos relacionados con las necesidades que se pueden pedir, el usuario puede también presionar en uno de estos y el dispositivo lo reconocerá.

Por otro lado, también se plasma una aplicación móvil como medio tecnológico. Luego de que el dispositivo reconozca la palabra dicha por el usuario, mediante conexión wifi a través de las ip esta palabra llega a la aplicación móvil del cuidador en forma de mensaje. Además, se muestra el gráfico relacionado a la necesidad que solicita el usuario.

La ingeniería en ciencias de la computación se podrá verificar en todo el código desarrollado que permite que el dispositivo pueda reconocer la voz de una persona. Por último, la tecnología se verá implícita al usar la aplicación móvil, que servirá como interfaz de recepción de las necesidades solicitadas.

## **CAPITULO IV**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **CONCLUSIONES**

- Se analizó las características de autonomía de los pacientes con discapacidad motriz para determinar los requerimientos de asistencia mediante observación de campo. Toda la información recopilada en cuanto a los requerimientos se la obtuvo mediante una entrevista para de esa manera conocer las necesidades de las personas con discapacidad motriz y la problemática a resolver.
- Se diseñó un sistema de asistencia para facilitar el trabajo de los cuidadores de pacientes con discapacidad motriz a través de tecnología de bajo costo. El trabajo de los cuidadores fue facilitado también mediante el desarrollo de una aplicación móvil usando lenguaje de programación en bloques, ya que esta permite recibir por mensajes todas las necesidades que el paciente solicita. Se realizaron 4 pruebas del funcionamiento de la app en general.
- Se pudo construir un dispositivo portátil capaz de reconocer la voz de las personas con discapacidad motriz y en general, para atender las necesidades de las mismas, realizando 4 pruebas de su funcionamiento. Este dispositivo fue capaz de reconocer 5 palabras como comandos de voz cuando el usuario solicite algo mediante las mismas, las cuales al ser detectadas por el dispositivo llegan a la aplicación móvil del cuidador.

#### **RECOMENDACIONES**

- Se recomienda que la recopilación de información se lo haga de manera presencial ya que de esa forma se puede tener un mejor entendimiento en cuanto a los requerimientos planteados por los usuarios. Además, utilizar técnicas de obtención de datos como entrevistas, encuestas ayudará a que toda la información recopilada se la tenga de una manera más organizada.
- Es recomendable tener una buena estabilidad en el internet porque tanto el dispositivo de reconocimiento de voz como la aplicación móvil estarán

conectadas mediante Wifi. Sin embargo, si existe problemas de conexión, por ende, se puede retardar las necesidades solicitadas por el paciente en cuanto al momento de llegada a la aplicación móvil.

- Si se desea desarrollar un dispositivo de reconocimiento de voz se lo debe realizar con la finalidad de que el usuario lo maneje con facilidad. La interfaz tanto del dispositivo como de la aplicación móvil debe ser accesible, interactiva y de fácil uso, tanto para el paciente y los cuidadores. Además de desarrollarlo en diferentes lenguajes de programación para así tener un conocimiento más amplio.
- Se debe prestar mucha atención si en algún momento se acaba la batería del dispositivo, es por eso que se recomienda en un tiempo dado hacer el cambio respectivo de batería, para posteriormente no tener inconvenientes.
- Se recomienda que la aplicación móvil se la desarrolle en herramientas que sean de fácil uso y de acuerdo a los requerimientos dados por el usuario.
- Es recomendable realizar las pruebas que sean necesarias para de esa manera tener un cierto grado de aceptación por parte de los usuarios. Es importante que al dispositivo antes de su implementación se realice una previa prueba para así poder corregir los posibles errores existentes, es decir realizar ciertas modificaciones.

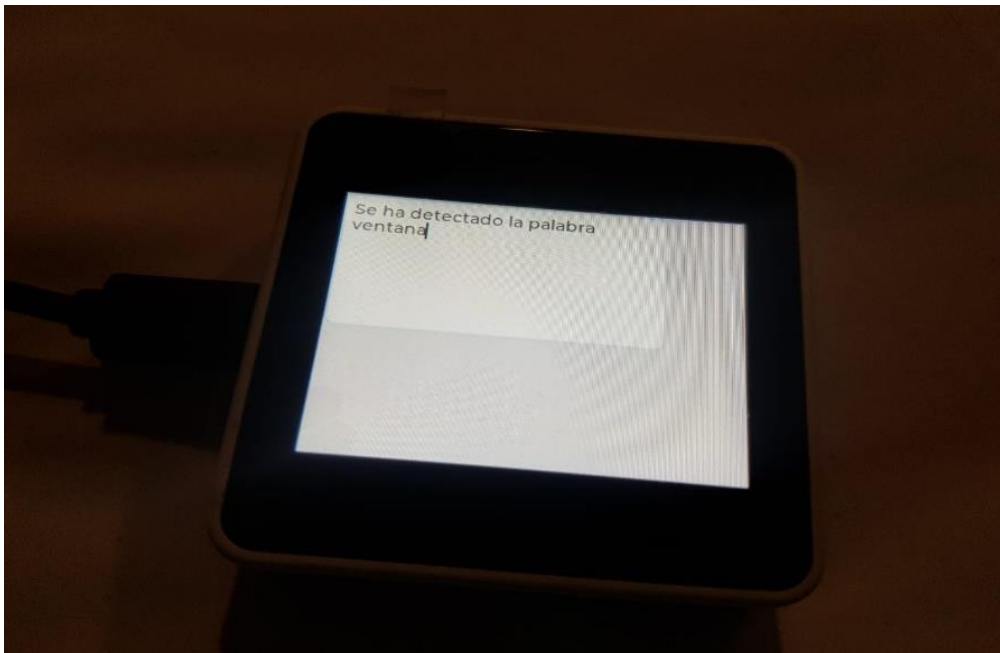
## BIBLIOGRAFÍA

- [1] C. Valencia y M. Bernal, «Cepal,» Marzo 2016. [En línea]. Available: [https://repositorio.cepal.org/bitstream/handle/11362/39995/S1600203\\_es.pdf?sequence=1&isAllowed=y](https://repositorio.cepal.org/bitstream/handle/11362/39995/S1600203_es.pdf?sequence=1&isAllowed=y).
- [2] CONADIS, «Consejo Discapacidades,» 2020. [En línea]. Available: <https://www.consejodiscapacidades.gob.ec/estadisticas-de-discapacidad/>.
- [3] G. Alban, «SISTEMA DOMÓTICO DE APOYO PARA PERSONAS CON DISCAPACIDAD MOTRIZ MEDIANTE TECNOLOGÍA MÓVIL Y RECONOCIMIENTO DE VOZ,» Ambato, 2018.
- [4] J. Gaitán, J. Velasco, J. Valencia y L. Tocarruncho, «SISTEMA DE RECONOCIMIENTO DE VOZ PARA PERSONAS CON DISCAPACIDAD MOTRIZ,» Bogotá, 2019.
- [5] E. Garnica, «CONTROL DE MANDO POR VOZ PARA PERSONAS CON DISCAPACIDAD,» Bogotá, 2018.
- [6] D. Tomala, «SISTEMA DOMÓTICO CONTROLADO POR VOZ PARA PERSONAS CON DISCAPACIDAD EN EXTREMIDADES SUPERIORES, UTILIZANDO TARJETA RASPBERRY PI,» Guayaquil, 2018.
- [7] OMS, «Organización Mundial de la Salud,» 4 Junio 2017. [En línea]. Available: [http://www.who.int/disabilities/world\\_report/2011/es/](http://www.who.int/disabilities/world_report/2011/es/).
- [8] U. D. PALERMO, «Diseño interior, una vivienda para personas con discapacidades físicas (discapacidades motoras: silla de rueda) y sensoriales (ceguera, sordomudo, hipoacusia) para lograr que puedan acceder a una mayor autonomía en su vida cotidiana,» 2018.
- [9] J. Cantón, La discapacidad (Características y necesidades de las personas en situación..., Google Books, 2015.
- [10] Servicio Nacional de Rehabilitación, «Discapacidad,» [En línea]. Available: <http://www.plb.gba.gov.ar/gba/plb/pdf/DISCAPACIDAD.pdf>.
- [11] Ouanono, «La justicia en la discapacidad: sepa cuáles son sus derechos.»  
]

- [12 C. N. d. F. Educativo, «Conafe,» [En línea]. Available:  
] [http://www.conafe.gob.mx/educacioncomunitaria/programainclusioneducativa/di scapacidad-motriz.pdf](http://www.conafe.gob.mx/educacioncomunitaria/programainclusioneducativa/di%20scapacidad-motriz.pdf).
- [13 Histornet, «Histornet,» 2017. [En línea]. Available:  
] <https://histornet.wordpress.com/2017/10/26/software-de-reconocimiento-de-voz/>.
- [14 A. Esparza, «Reconocimiento de voz. Ciencia Tecnológica,» 2019, pp. 1-3.  
]
- [15 BricoGeek, «M5Stack - ESP32 Core2 IoT,» [En línea]. Available:  
] <https://tienda.bricogeek.com/m5stack-esp32/1509-m5stack-esp32-core2-iot.html>.
- [16 Dani No, «MICROPYTHON ESP32 – ¿Qué son Python y MicroPython?,»  
] [En línea]. Available: [esploradores.com/python\\_y\\_micropython\\_que\\_son/](http://esploradores.com/python_y_micropython_que_son/).
- [17 J. Cuello y J. Vittone, Diseñando apps para móviles, Argentina: Catalina  
] Duque Giraldo, 2017.
- [18 Google, «Firebase,» 2021. [En línea]. Available: <https://firebase.google.com>.  
]

## ANEXOS

### ANEXOS 1: Fotografías M5Stack



## ANEXOS 2: Interfaz de Inicio App



## ANEXOS 3: Codificación de la Interfaz M5stack y conexión Wifi

```
// nombre y contraseña del internet
const char* ssid = "NANCY REINOSO";
const char* password = "0502234073";
int var=0;
int16_t ven_f=0;
int16_t lec=0;
// CONFIGURACION DE LA IP ESTÁTICA
IPAddress local_IP(192, 168, 0, 125);
IPAddress gateway(192, 168, 1, 1);
IPAddress subnet(255, 255, 255, 0);

WiFiServer server(80); // PUERTO 80

String estado = "";
int wait30 = 30000; // TIEMPO DE RECONEXION CUANDO SE PIERDE.
```

```

ButtonColors on_clrs = {TFT_TRANSPARENT, RED, RED};
ButtonColors off_clrs = {TFT_TRANSPARENT, WHITE, GREEN};
//TFT_TRANSPARENT
//DECLARACION DE BOTONES
//(X,Y,DX,DY)
int px=10,py=50,mx=80,my=56,ax=20,ay=5;
int ix=30,iy=55,dix=90,diy=60;

Button luces(px, py, mx, my, false, "", off_clrs, on_clrs, BC_DATUM);
Button musica(px+mx+ax, py, mx, my, false, "", off_clrs, on_clrs, CC_DATUM);
Button comida(px+2*mx+2*ax, py, mx, my, false, "", off_clrs, on_clrs,
CC_DATUM);
Button agua(px, py+my+ay, mx,my, false, "", off_clrs, on_clrs, CC_DATUM);
Button bano(px+mx+ax, py+my+ay, mx, my, false, "", off_clrs, on_clrs,
CC_DATUM);
Button telefono(px+2*mx+2*ax,py+my+ay, mx, my, false, "", off_clrs, on_clrs,
CC_DATUM);
Button puerta(px, py+2*my+2*ay, mx,my, false, "", off_clrs, on_clrs,
CC_DATUM);
Button ventana(px+mx+ax, py+2*my+2*ay, mx, my, false, "", off_clrs, on_clrs,
CC_DATUM);
Button ayuda(px+2*mx+2*ax,py+2*my+2*ay, mx, my, false, "", off_clrs,
on_clrs, CC_DATUM);

//FUNCION PARA CONECTAR EL WIFI
void conectWifi(){
  M5.Lcd.clear();
  M5.Lcd.fillScreen(BLUE);
  M5.lcd.setCursor(1,20);
  M5.Lcd.setTextColor(WHITE);
  M5.Lcd.println("Conectando..");
  M5.Lcd.println(ssid);
  M5.Lcd.println(WiFi.localIP());
  M5.Lcd.println(WiFi.networkID());
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    M5.Lcd.print(".");
  }
  M5.Lcd.println("Conectado Wifi..");
  server.begin();
  M5.Lcd.println("Servidor iniciado: ");
}

```

```

// Esta es la IP
M5.Lcd.print("This is IP to connect to the WebServer: ");
M5.Lcd.print("http://");
M5.Lcd.println(WiFi.localIP());
// If disconnected, try to reconnect every 30 seconds.
if ((WiFi.status() != WL_CONNECTED) && (millis() > wait30)) {
  M5.Lcd.println("Trying to reconnect WiFi...");
  WiFi.disconnect();
  WiFi.begin(ssid, password);
  wait30 = millis() + 30000;
}
// Check if a client has connected..
WiFiClient client = server.available();
if (!client) {
  return;
}

M5.Lcd.println("New client: ");
M5.Lcd.println(client.remoteIP());
client.println("HTTP/1.1 200 OK");
// Espera hasta que el cliente envíe datos.
// while(!client.available()){ delay(1); }

////////////////////////////////////
// Read the information sent by the client.
String req = client.readStringUntil('\r');
M5.Lcd.println(req);

////////////////////////////////////
// Página WEB. //////////////////////////////////
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println(""); // Comillas importantes.
client.println(estado); // Return status.

client.flush();
//client.stop();
}

void botones(){
  M5.Lcd.clear();
  M5.update();
}

```

```

M5.Lcd.fillScreen(WHITE);
M5.Lcd.drawJpgFile(SPIFFS, "/indoamerica1.jpg", 0, 0);
M5.Lcd.drawJpgFile(SPIFFS, "/foco1.jpg", ix, iy);
M5.Lcd.drawJpgFile(SPIFFS, "/musica.jpg", ix+dix, iy);
M5.Lcd.drawJpgFile(SPIFFS, "/comida.jpg", ix+2*dix+10, iy);
M5.Lcd.drawJpgFile(SPIFFS, "/agua.jpg", ix-8, iy+diy);
M5.Lcd.drawJpgFile(SPIFFS, "/bano.jpg", ix+dix, iy+diy);
M5.Lcd.drawJpgFile(SPIFFS, "/telefono.jpg", ix+2*dix+10, iy+diy);
M5.Lcd.drawJpgFile(SPIFFS, "/puerta.jpg", ix-8, iy+2*diy);
M5.Lcd.drawJpgFile(SPIFFS, "/ventana.jpg", ix+dix, iy+2*diy);
M5.Lcd.drawJpgFile(SPIFFS, "/emergencia.jpg", ix+2*dix+10, iy+2*diy);

}
void fail(){
  M5.Lcd.clear();
  M5.Lcd.fillScreen(RED);
}

void setup() {
  M5.begin(
    true, // LCDEnable
    false, // SDEnable
    false, // SerialEnable !important
    false // I2CEnable
  );
  M5.Axp.SetLcdVoltage(3000);

  WiFi.begin(ssid, password);
  M5.Lcd.fillScreen(WHITE);

  Serial.begin(115200);
  if(!SPIFFS.begin(true))
  {
    // Serial.println("SPIFFS Mount Failed");
    return;
  }
}

void loop() {

  M5.update();

```

```

if (M5.BtnA.wasReleased() || M5.BtnA.pressedFor(1000, 200)) {
  conectWifi();ven_f=1;
} else if (M5.BtnB.wasReleased() || M5.BtnB.pressedFor(1000, 200)) {
  if(ven_f==1){ ven_f=2;botones();M5.Buttons.draw();}
  if(ven_f==0){ fail();}
} else if (M5.BtnC.wasReleased() || M5.BtnC.pressedFor(1000, 200)) {

}

```

```
//LECTURA ESTADO SE BOTONES
```

```

if(bano.wasReleased() ||bano.pressedFor(1000, 200)){ var=1; }
if(luces.wasReleased() ||luces.pressedFor(1000, 200)){ var=2; }
if(comida.wasReleased() ||comida.pressedFor(1000, 200)){ var=3; }
if(musica.wasReleased() ||musica.pressedFor(1000, 200)){ var=4; }
if(puerta.wasReleased() ||puerta.pressedFor(1000, 200)){ var=5; }
if(ventana.wasReleased() ||ventana.pressedFor(1000, 200)){ var=6; }
if(telefono.wasReleased() ||ayuda.pressedFor(1000, 200)){ var=7; }
if(ayuda.wasReleased() ||ayuda.pressedFor(1000, 200)){ var=8; }

```

```
// If disconnected, try to reconnect every 30 seconds.
```

```

if ((WiFi.status() != WL_CONNECTED) && (millis() > wait30)) {
  M5.Lcd.println("Trying to reconnect WiFi...");
  WiFi.disconnect();
  WiFi.begin(ssid, password);
  wait30 = millis() + 30000;
}

```

```
// Check if a client has connected..
```

```

WiFiClient client = server.available();
if (!client) {
  return;M5.Lcd.println(". ");
}
//client.println("HTTP/1.1 800 OK");
//client.println("Content-Type: text/html");
//client.println(""); // Comillas importantes.
//client.println(estado); // Return status.
//5.Lcd.println("New client: ");
//M5.Lcd.println(client.remoteIP());

```

```
//BAÑO
```

```

if(var==1){
  //M5.Lcd.println("1");
  client.println("HTTP/1.1 200 OK");
}

```

```

    client.println("Content-Type: text/html");
    client.println(""); // Comillas importantes.
    client.println(estado);
}
//LUCES
if(var==2){
    //M5.Lcd.println("2");
    client.println("HTTP/1.1 210 OK");
    client.println("Content-Type: text/html");
    client.println(""); // Comillas importantes.
    client.println(estado);
}
//COMIDA
if(var==3){
    //M5.Lcd.println("3");
    client.println("HTTP/1.1 220 OK");
    client.println("Content-Type: text/html");
    client.println(""); // Comillas importantes.
    client.println(estado);
}
//MUSICA
if(var==4){
    client.println("HTTP/1.1 230 OK");
    client.println("Content-Type: text/html");
    client.println(""); // Comillas importantes.
    client.println(estado);
}
// PUERTA
if(var==5){
    client.println("HTTP/1.1 240 OK");
    client.println("Content-Type: text/html");
    client.println(""); // Comillas importantes.
    client.println(estado);
}
//VENTANA
if(var==6){
    client.println("HTTP/1.1 250 OK");
    client.println("Content-Type: text/html");
    client.println(""); // Comillas importantes.
    client.println(estado);
}
//TELEFONO

```

```

if(var==7){
    client.println("HTTP/1.1 260 OK");
    client.println("Content-Type: text/html");
    client.println(""); // Comillas importantes.
    client.println(estado);
}
//AYUDA
if(var==8){
    client.println("HTTP/1.1 270 OK");
    client.println("Content-Type: text/html");
    client.println(""); // Comillas importantes.
    client.println(estado);
}
//var=0;
// Return status.}
// Espera hasta que el cliente envíe datos.
// while(!client.available()){ delay(1); }

////////////////////////////////////
// Read the information sent by the client.
String req = client.readStringUntil('\r');
client.flush();
//client.stop();

}

```

#### ANEXOS 4: Análisis de Audio de la base de datos

```

command_words = [
    'AGUA',
    'AYUDA',
    'BANO',
    'COMIDA',
    'LUCES',
    'MUSICA',
    'TELEFONO',
    'PUERTA',
    'VENTANA',
    '_invalid',
]
nonsense_words = [
    'BANO',

```

```

]

a = gfile.glob('C:/audios/'+ 'comida'+ '/*.wav')
# get all the files in a directory
def get_files(word):
    return gfile.glob('C:/audios/'+ word+ '/*.wav')

# get the location of the voice
def get_voice_position(audio, noise_floor):
    audio = audio - np.mean(audio)
    audio = audio / np.max(np.abs(audio))
    return tfio.audio.trim(audio, axis=0, epsilon=noise_floor)

# Work out how much of the audio file is actually voice
def get_voice_length(audio, noise_floor):
    position = get_voice_position(audio, noise_floor)
    return (position[1] - position[0]).numpy()

# is enough voice present?
def is_voice_present(audio, noise_floor, required_length):
    voice_length = get_voice_length(audio, noise_floor)
    return voice_length >= required_length

# is the audio the correct length?
def is_correct_length(audio, expected_length):
    return (audio.shape[0]==expected_length).numpy()
def is_valid_file(file_name):
    # load the audio file
    audio_tensor = tfio.audio.AudioIOTensor(file_name)
    # check the file is long enough
    if not is_correct_length(audio_tensor, EXPECTED_SAMPLES):
        return False
    # convert the audio to an array of floats and scale it to between -1 and 1
    audio = tf.cast(audio_tensor[:], tf.float32)
    audio = audio - np.mean(audio)
    audio = audio / np.max(np.abs(audio))
    # is there any voice in the audio?
    if not is_voice_present(audio, NOISE_FLOOR,
MINIMUM_VOICE_LENGTH):
        return False
    return True
def get_spectrogram(audio):

```

```

# normalise the audio
audio = audio - np.mean(audio)
audio = audio / np.max(np.abs(audio))
# create the spectrogram
spectrogram = audio_ops.audio_spectrogram(audio,
                                           window_size=320,
                                           stride=160,
                                           magnitude_squared=True).numpy()
# reduce the number of frequency bins in our spectrogram to a more sensible
level
spectrogram = tf.nn.pool(
    input=tf.expand_dims(spectrogram, -1),
    window_shape=[1, 6],
    strides=[1, 6],
    pooling_type='AVG',
    padding='SAME')
spectrogram = tf.squeeze(spectrogram, axis=0)
spectrogram = np.log10(spectrogram + 1e-6)
return spectrogram

# process a file into its spectrogram
def process_file(file_path):
    # load the audio file
    audio_tensor = tfio.audio.AudioIOTensor(file_path)
    # convert the audio to an array of floats and scale it to between -1 and 1
    audio = tf.cast(audio_tensor[:], tf.float32)
    audio = audio - np.mean(audio)
    audio = audio / np.max(np.abs(audio))
    # randomly reposition the audio in the sample
    voice_start, voice_end = get_voice_position(audio, NOISE_FLOOR)
    end_gap=len(audio) - voice_end
    random_offset = np.random.uniform(0, voice_start+end_gap)
    audio = np.roll(audio,-random_offset+end_gap)
    # add some random background noise
    background_volume = np.random.uniform(0, 0.1)
    # get the background noise files
    background_files = get_files('_background_noise_')
    background_file = np.random.choice(background_files)
    background_tensor = tfio.audio.AudioIOTensor(background_file)
    background_start = np.random.randint(0, len(background_tensor) - 16000)
    # normalise the background noise

```

```

    background = background - np.mean(background)
    background = background / np.max(np.abs(background))
    # mix the audio with the scaled background
    audio = audio + background_volume * background
    # get the spectrogram
    return get_spectrogram(audio)

train = []
validate = []
test = []

TRAIN_SIZE=0.8
VALIDATION_SIZE=0.1
TEST_SIZE=0.1

def process_files(file_names, label, repeat=1):
    file_names = tf.repeat(file_names, repeat).numpy()
    return [(process_file(file_name), label) for file_name in tqdm(file_names,
desc=f"{word} ({label})", leave=False)]

# process the files for a word into the spectrogram and one hot encoding word
value
def process_word(word, label, repeat=1):
    # get a list of files names for the word
    file_names = [file_name for file_name in tqdm(get_files(word),
desc="Checking", leave=False) if is_valid_file(file_name)]
    # randomly shuffle the filenames
    np.random.shuffle(file_names)
    # split the files into train, validate and test buckets
    train_size=int(TRAIN_SIZE*len(file_names))
    validation_size=int(VALIDATION_SIZE*len(file_names))
    test_size=int(TEST_SIZE*len(file_names))
    # get the training samples
    train.extend(
        process_files(
            file_names[:train_size],
            label,
            repeat=repeat
        )
    )

```

```

# and the validation samples
validate.extend(
    process_files(
        file_names[train_size:train_size+validation_size],
        label,
        repeat=repeat
    )
)
# and the test samples
test.extend(
    process_files(
        file_names[train_size+validation_size:],
        label,
        repeat=repeat
    )
)

# process all the command words
for word in tqdm(command_words, desc="Processing words"):
    if '_' not in word:
        repeat = 40 if word in ('TELEFONO','VENTANA') else 20
        process_word(word, command_words.index(word), repeat=repeat)

# all the nonsense words
for word in tqdm(nonsense_words, desc="Processing words"):
    if '_' not in word:
        process_word(word, command_words.index('_invalid'), repeat=1)

print(len(train), len(test), len(validate))

# process the background noise files
def process_background(file_name, label):
    # load the audio file
    audio_tensor = tfio.audio.AudioIOTensor(file_name)
    audio = tf.cast(audio_tensor[:, :], tf.float32)
    audio_length = len(audio)
    samples = []
    for section_start in tqdm(range(0, audio_length-EXPECTED_SAMPLES,
16000), desc=file_name, leave=False):
        section_end = section_start + EXPECTED_SAMPLES
        section = audio[section_start:section_end]
        # get the spectrogram

```

```

spectrogram = get_spectrogram(section)
samples.append((spectrogram, label))

# simulate random utterances
for section_index in tqdm(range(1000), desc="Simulated Words", leave=False):
    section_start = np.random.randint(0, audio_length -
EXPECTED_SAMPLES)
    section_end = section_start + EXPECTED_SAMPLES
    section = np.reshape(audio[section_start:section_end],
(EXPECTED_SAMPLES))

    result = np.zeros((EXPECTED_SAMPLES))
    # create a pseudo bit of voice
    voice_length = np.random.randint(MINIMUM_VOICE_LENGTH/2,
EXPECTED_SAMPLES)
    voice_start = np.random.randint(0, EXPECTED_SAMPLES - voice_length)
    hamming = np.hamming(voice_length)
    # amplify the voice section
    result[voice_start:voice_start+voice_length] = hamming *
section[voice_start:voice_start+voice_length]
    # get the spectrogram
    spectrogram = get_spectrogram(np.reshape(section, (16000, 1)))
    samples.append((spectrogram, label))

np.random.shuffle(samples)

train_size=int(TRAIN_SIZE*len(samples))
validation_size=int(VALIDATION_SIZE*len(samples))
test_size=int(TEST_SIZE*len(samples))

train.extend(samples[:train_size])

validate.extend(samples[train_size:train_size+validation_size])

test.extend(samples[train_size+validation_size:])

for file_name in tqdm(get_files('_background_noise_'), desc="Processing
Background Noise"):
    process_background(file_name, command_words.index("_invalid"))

print(len(train), len(test), len(validate))

```

```

def process_problem_noise(file_name, label):
    samples = []
    # load the audio file
    audio_tensor = tfio.audio.AudioIOTensor(file_name)
    audio = tf.cast(audio_tensor[:, :], tf.float32)
    audio_length = len(audio)
    samples = []
    for section_start in tqdm(range(0, audio_length-EXPECTED_SAMPLES, 800),
desc=file_name, leave=False):
        section_end = section_start + EXPECTED_SAMPLES
        section = audio[section_start:section_end]
        # get the spectrogram
        spectrogram = get_spectrogram(section)
        samples.append((spectrogram, label))

np.random.shuffle(samples)

train_size=int(TRAIN_SIZE*len(samples))
validation_size=int(VALIDATION_SIZE*len(samples))
test_size=int(TEST_SIZE*len(samples))

train.extend(samples[:train_size])
validate.extend(samples[train_size:train_size+validation_size])
test.extend(samples[train_size+validation_size:])

for file_name in tqdm(get_files("_problem_noise_"), desc="Processing problem
noise"):
    process_problem_noise(file_name, command_words.index("_invalid"))

print(len(train), len(test), len(validate))
# randomise the training samples
np.random.shuffle(train)

X_train, Y_train = zip(*train)
X_validate, Y_validate = zip(*validate)
X_test, Y_test = zip(*test)

# save the computed data
np.savez_compressed(
    "training_spectrogram.npz",

```

```

    X=X_train, Y=Y_train)
print("Saved training data")
np.savez_compressed(
    "validation_spectrogram.npz",
    X=X_validate, Y=Y_validate)
print("Saved validation data")
np.savez_compressed(
    "test_spectrogram.npz",
    X=X_test, Y=Y_test)
print("Saved test data")

# get the width and height of the spectrogram "image"
IMG_WIDTH=X_train[0].shape[0]
IMG_HEIGHT=X_train[0].shape[1]

def plot_images2(images_arr, imageWidth, imageHeight):
    fig, axes = plt.subplots(2, 5, figsize=(10, 10))
    axes = axes.flatten()
    for img, ax in zip(images_arr, axes):
        ax.imshow(np.reshape(img, (imageWidth, imageHeight)))
        ax.axis("off")
    plt.tight_layout()
    plt.show()

word_index = command_words.index("BANO")
X_left = np.array(X_train)[np.array(Y_train) == word_index]
plot_images2(X_left[:10], IMG_WIDTH, IMG_HEIGHT)

```

### **ANEXOS 5:** Creación del modelo de la red neuronal

```

command_words = [
    'AGUA',
    'AYUDA',
    'BANO',
    'COMIDA',
    'LUCES',
    'MUSICA',
    'TELEFONO',
    'PUERTA',
    'VENTANA',

```

```

    '_invalid',
]
# Load up the spectrograms and labels
training_spectrogram = np.load('training_spectrogram.npz')
validation_spectrogram = np.load('validation_spectrogram.npz')
test_spectrogram = np.load('test_spectrogram.npz')

# extract the data from the files
X_train = training_spectrogram['X']
Y_train_cats = training_spectrogram['Y']
X_validate = validation_spectrogram['X']
Y_validate_cats = validation_spectrogram['Y']
X_test = test_spectrogram['X']
Y_test_cats = test_spectrogram['Y']

# get the width and height of the spectrogram "image"
IMG_WIDTH=X_train[0].shape[0]
IMG_HEIGHT=X_train[0].shape[1]

# plot a distribution of the words
plt.hist(Y_train_cats, bins=range(0,len(command_words)+1), align='left')

unique, counts = np.unique(Y_train_cats, return_counts=True)
print(unique, counts)
dict(zip([command_words[i] for i in unique], counts))

Y_train = tf.one_hot(Y_train_cats, len(command_words))
Y_validate = tf.one_hot(Y_validate_cats, len(command_words))
Y_test = tf.one_hot(Y_test_cats, len(command_words))

# create the datasets for training
batch_size = 32

train_dataset = Dataset.from_tensor_slices(
    (X_train, Y_train)
).repeat(
    count=-1
).shuffle(
    len(X_train)
).batch(
    batch_size
)

```

```
validation_dataset = Dataset.from_tensor_slices((X_validate,
Y_validate)).batch(X_validate.shape[0]//10)
```

```
test_dataset = Dataset.from_tensor_slices((X_test, Y_test)).batch(len(X_test))
```

```
model = Sequential([
    Conv2D(4, 3,
        padding='same',
        activation='relu',
        kernel_regularizer=regularizers.l2(0.001),
        name='conv_layer1',
        input_shape=(IMG_WIDTH, IMG_HEIGHT, 1)),
    MaxPooling2D(name='max_pooling1', pool_size=(2,2)),
    Conv2D(4, 3,
        padding='same',
        activation='relu',
        kernel_regularizer=regularizers.l2(0.001),
        name='conv_layer2'),
    MaxPooling2D(name='max_pooling3', pool_size=(2,2)),
    Flatten(),
    Dropout(0.1),
    Dense(
        80,
        activation='relu',
        kernel_regularizer=regularizers.l2(0.001),
        name='hidden_layer1'
    ),
    Dropout(0.1),
    Dense(
        len(command_words),
        activation='softmax',
        kernel_regularizer=regularizers.l2(0.001),
        name='output'
    )
])
model.summary()
```

### **Modelo de Tren**

```
model_checkpoint_callback = tf.keras.callbacks.ModelCheckpoint(
    filepath="checkpoint.model",
```

```

    monitor='val_accuracy',
    mode='max',
    save_best_only=True)

history = model.fit(
    train_dataset,
    steps_per_epoch=len(X_train) // batch_size,
    epochs=epochs,
    validation_data=validation_dataset,
    validation_steps=10,
    callbacks=[tensorboard_callback, model_checkpoint_callback]
)
model.save("trained.model")

```

### Prueba del modelo

```

model.save("trained.model")
model2 = keras.models.load_model("trained.model")
results = model2.evaluate(X_test, tf.cast(Y_test, tf.float32), batch_size=128)
predictions = model2.predict(X_test, 128)

```

```
import itertools
```

```
def plot_confusion_matrix(cm, class_names):
```

```
    """
```

Returns a matplotlib figure containing the plotted confusion matrix.

Args:

cm (array, shape = [n, n]): a confusion matrix of integer classes

class\_names (array, shape = [n]): String names of the integer classes

```
    """
```

```
    cm = cm.numpy()
```

```
    # Normalize the confusion matrix.
```

```
    cm = np.around(cm.astype("float") / cm.sum(axis=1)[:], np.newaxis,
decimals=2)
```

```
    figure = plt.figure(figsize=(8, 8))
```

```
    plt.imshow(cm, interpolation="nearest", cmap=plt.cm.Blues)
```

```
    plt.title("Confusion matrix")
```

```
    plt.colorbar()
```

```
    tick_marks = np.arange(len(class_names))
```

```

plt.xticks(tick_marks, class_names, rotation=45)
plt.yticks(tick_marks, class_names)

# Use white text if squares are dark; otherwise black.
threshold = cm.max() / 2.0
for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
    color = "white" if cm[i, j] > threshold else "black"
    plt.text(j, i, cm[i, j], horizontalalignment="center", color=color)

plt.tight_layout()
plt.ylabel("True label")
plt.xlabel("Predicted label")
plt.show()
# return figure

cm = tf.math.confusion_matrix(
    labels=tf.argmax(Y_test, 1), predictions=tf.argmax(predictions, 1)
)
plot_confusion_matrix(cm, command_words)

```

### ANEXOS 6: Conversión del modelo a código C (TensorFlowLite)

```

training_spectrogram = np.load('training_spectrogram.npz')
validation_spectrogram = np.load('validation_spectrogram.npz')
test_spectrogram = np.load('test_spectrogram.npz')

X_train = training_spectrogram['X']
X_validate = validation_spectrogram['X']
X_test = test_spectrogram['X']

complete_train_X = np.concatenate((X_train, X_validate, X_test))
# complete_train_X = X_validate

converter2 = tf.lite.TFLiteConverter.from_saved_model("fully_trained.model")
converter2.optimizations = [tf.lite.Optimize.DEFAULT]
def representative_dataset_gen():
    for i in range(0, len(complete_train_X), 100):
        # Get sample input data as a numpy array in a method of your choosing.
        yield [complete_train_X[i:i+100]]
converter2.representative_dataset = representative_dataset_gen
# converter.optimizations = [tf.lite.Optimize.OPTIMIZE_FOR_SIZE]

```

```

converter2.target_spec.supported_ops
[tf.lite.OpsSet.TFLITE_BUILTINS_INT8]
tflite_quant_model = converter2.convert()
open("converted_model.tflite", "wb").write(tflite_quant_model)

```

### ANEXOS 7: Configuración del micrófono por entrada pdm

```

// i2s config for reading from both channels of I2S
i2s_config_t i2sMemsConfigBothChannels = {
    .mode = (i2s_mode_t)(I2S_MODE_MASTER | I2S_MODE_RX|
I2S_MODE_PDM),
    .sample_rate = 16000,
    .bits_per_sample = I2S_BITS_PER_SAMPLE_32BIT,
    .channel_format = I2S_MIC_CHANNEL,
    .communication_format = i2s_comm_format_t(I2S_COMM_FORMAT_I2S),
    .intr_alloc_flags = ESP_INTR_FLAG_LEVEL1,
    .dma_buf_count = 4,
    .dma_buf_len = 64,
    .use_apll = false,
    .tx_desc_auto_clear = false,
    .fixed_mclk = 0};

// i2s microphone pins
i2s_pin_config_t i2s_mic_pins = {

    .bck_io_num = GPIO_NUM_12,
    .ws_io_num = GPIO_NUM_0,
    .data_out_num = I2S_PIN_NO_CHANGE,
    .data_in_num = GPIO_NUM_34};

// This task does all the heavy lifting for our application
void applicationTask(void *param)
{
    CommandDetector *commandDetector = static_cast<CommandDetector
*>(param);

    const TickType_t xMaxBlockTime = pdMS_TO_TICKS(100);
    while (true)
    {
        // wait for some audio samples to arrive
        uint32_t ulNotificationValue = ulTaskNotifyTake(pdTRUE, xMaxBlockTime);
        if (ulNotificationValue > 0)
        {

```

```

    commandDetector->run();
  }
}
}

```

## ANEXOS 8: Inicialización de la M5Stack y Comunicación Serial

```

//inicializacion M5STACK - comunicacion serial
void setup()
{
  M5.begin(
    true, // LCDEnable
    false, // SDEnable
    false, // SerialEnable !important
    false // I2CEnable
  );
  M5.Axp.SetSpkEnable(true); // Enable speaker power. 启用扬声器电源
  M5.Lcd.fillScreen(WHITE);

  Serial.begin(115200);
  delay(1000);
  Serial.println("EMPEZANDO...");

  // make sure we don't get killed for our long running tasks
  esp_task_wdt_init(10, false);

  I2SSampler *i2s_sampler = new I2SMicSampler(i2s_mic_pins, false);
  // start up the I2S input (from either an I2S microphone or Analogue microphone
  via the ADC)

  // the command processor
  CommandProcessor *command_processor = new CommandProcessor();

  // create our application
  CommandDetector *commandDetector = new CommandDetector(i2s_sampler,
  command_processor);

  // set up the i2s sample writer task
  TaskHandle_t applicationTaskHandle;

```

```
xTaskCreatePinnedToCore(applicationTask, "Command Detect", 8192,
commandDetector, 1, &applicationTaskHandle, 0);
```

```
// start sampling from i2s device - use I2S_NUM_0 as that's the one that supports
the internal ADC
```

```
#ifdef USE_I2S_MIC_INPUT
```

```
    i2s_sampler->start(I2S_NUM_0, i2sMemsConfigBothChannels,
applicationTaskHandle);
```

```
#else
```

```
    i2s_sampler->start(I2S_NUM_0, adcI2SConfig, applicationTaskHandle);
```

```
#endif
```

```
}
```

```
void loop()
```

```
{
```

```
    //vTaskDelay(pdMS_TO_TICKS(1000));
```

```
}
```

**ANEXOS 9:** Procesamiento de audio y Comparación de la red neuronal

```
const char *words[] = {
```

```
    'AGUA',
```

```
    'AYUDA',
```

```
    'BANO',
```

```
    'COMIDA',
```

```
    'LUCES',
```

```
    'MUSICA',
```

```
    'TELEFONO',
```

```
    'PUERTA',
```

```
    'VENTANA',
```

```
};
```

```
void commandQueueProcessorTask(void *param)
```

```
{
```

```
    CommandProcessor *commandProcessor = (CommandProcessor *)param;
```

```
    while (true)
```

```
    {
```

```
        uint16_t commandIndex = 0;
```

```
        if (xQueueReceive(commandProcessor->m_command_queue_handle,
&commandIndex, portMAX_DELAY) == pdTRUE)
```

```
        {
```

```
            commandProcessor->processCommand(commandIndex);
```

```

    }
  }
}

int calcDuty(int ms)
{
  // 50Hz = 20ms period
  return (65536 * ms) / 20000;
}

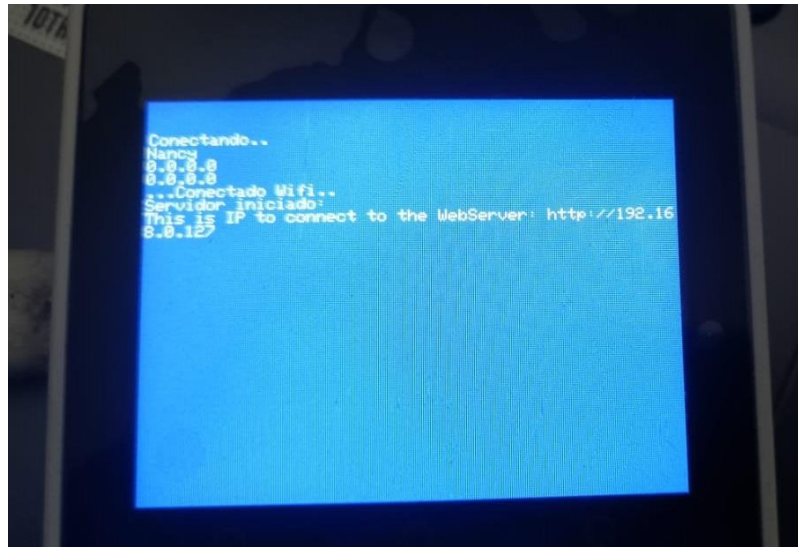
//Para saber si la palabra se detectó
CommandProcessor::CommandProcessor()
{
  // allow up to 5 commands to be in flight at once
  m_command_queue_handle = xQueueCreate(5, sizeof(uint16_t));
  if (!m_command_queue_handle)
  {
    Serial.println("Failed to create command queue");
  }
  // kick off the command processor task
  TaskHandle_t command_queue_task_handle;
  xTaskCreate(commandQueueProcessorTask, "Command Queue Processor",
1024, this, 1, &command_queue_task_handle);
}

void CommandProcessor::queueCommand(uint16_t commandIndex, float
best_score)
{
  // unsigned long now = millis();
  Serial.printf("leyo");
  if (commandIndex != 5 && commandIndex != -1)
  {
    Serial.printf("***** %ld Detected command %s(%f)\n", millis(),
words[commandIndex], best_score);
    M5.Lcd.setTextColor(WHITE);
    M5.Lcd.setCursor(10, 26);
    M5.Lcd.printf(words[commandIndex]);
    if (xQueueSendToBack(m_command_queue_handle, &commandIndex, 0) !=
pdTRUE)
    {
      Serial.println("No more space for command");
    }
  }
}

```

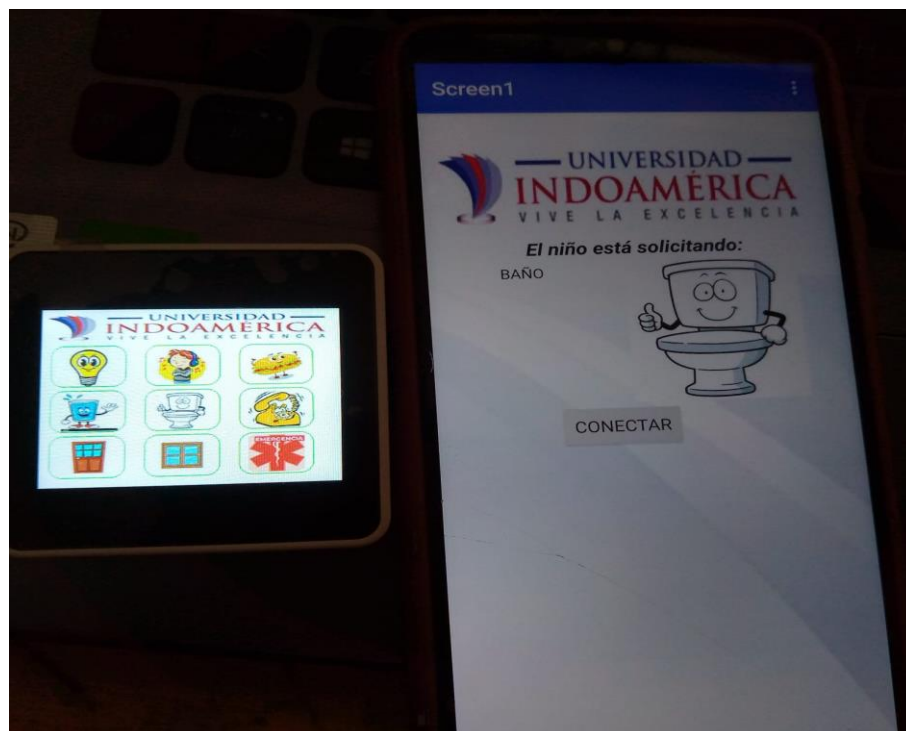
## ANEXOS 10: Pruebas de Integración

Se presenta la conexión de Wifi del dispositivo de reconocimiento de voz y la aplicación móvil mediante las IP.



## ANEXOS 11: Pruebas de Validación

Se puede verificar el correcto funcionamiento tanto del dispositivo y de la aplicación móvil.



## ANEXOS 12: Acta de Conformidad

Ambato 01 de abril, 2022

Licenciada

María Belén Camino

**Directora del Centro de Rehabilitación Bendiciones**

De mis consideraciones

Con referencia a la petición brindada por el centro para desarrollo e implementación de un "Dispositivo de asistencia de voz para pacientes con discapacidad motriz en el Centro de Rehabilitación "Bendiciones" en la ciudad de Ambato", por este medio, realizo la entrega del dispositivo el cual cumple con los requerimientos expuestos por los beneficiarios. Dicho software cumplirá con el propósito de generar una aceptación a las nuevas tecnologías actuales, brindando ayuda e innovación dentro del área de la salud.

Por la atención a la presente, anticipo mi agradecimiento.

Atentamente,





Kevin Joel López Granizo

1207909142

<b>Ciente</b>	
Lic. María Belén Camino Directora del Centro de Rehabilitación Bendiciones	

<b>Entrega:</b>	Dispositivo de asistencia de voz para pacientes con discapacidad motriz en el Centro de Rehabilitación "Bendiciones" en la ciudad de Ambato
<b>Fecha:</b>	01-04-2022

<b>Elementos entregados</b>	
Dispositivo de asistencia de voz para pacientes con discapacidad motriz en el Centro de Rehabilitación "Bendiciones" en la ciudad de Ambato	

<b>El cliente recibe conforme</b>	<b>Entrega por el estudiante</b>
 Lic. María Belén Camino	 Kevin Joel López Granizo



## ANEXOS 13: Manual de Usuario

### M5stack

Se presenta el manual de usuario del dispositivo de reconocimiento de voz, así como de la aplicación móvil.

El dispositivo de reconocimiento de voz, M5stack consta con una interfaz sencilla en donde se encuentran inmersos gráficos relacionados con las necesidades que se puede solicitar como de baño, ventana, comida, etc.

Para poder solicitar una necesidad se puede realizar de dos maneras. La primera forma consiste en que el dispositivo desde que se conecta a la red va a estar reconociendo la voz y el usuario puede decir una de las palabras clave, pronunciándolas con su voz para poder pedir esa necesidad.

La segunda forma es presionando uno de los gráficos configurados como botones que hacen relación a las necesidades que se puede solicitar, por ejemplo, presionamos la imagen del inodoro, esto quiere decir que se solicita la necesidad de ir al baño.



## Aplicación Móvil

La aplicación móvil es una interfaz sencilla e intuitiva que consta de dos pantallas. La primera pantalla es la de bienvenida y carga en donde se encuentra inmersa un logo representativo y la barra de carga de inicio para entrar a la pantalla principal.



En la segunda pantalla se encuentra un botón para conectarnos con el dispositivo de reconocimiento de voz. En esta pantalla únicamente llegarán todas las peticiones que el usuario solicite en forma de mensaje, a la vez se puede visualizar el gráfico relacionado con la necesidad solicitada.



Ambato 19 de abril de 2022

Ing. Belén Ruales  
**Decana FITIC**

Presente.-

De mi consideración:

Reciba un cordial y muy respetuoso saludo, a la vez me permito augurarle el mejor de los éxitos en tan delicadas funciones.

Por medio de la presente, informo que se ha concluido el presente trabajo desarrollado por Kevin López como parte del proyecto de titulación que tiene como tema “DESARROLLO DE UN DISPOSITIVO DE ASISTENCIA DE VOZ PARA PACIENTES CON DISCAPACIDAD MOTRIZ EN EL CENTRO DE REHABILITACIÓN “BENDICIONES” EN LA CIUDAD DE AMBATO”.

Atentamente,



Firmado digitalmente  
por JOSE LUIS  
VARELA ALDAS  
Fecha: 2022-04-20 09:58:05:00



Ing. José Varela  
**Docente FITIC**



AMBATO:

Dirección: Bolívar 20-35 y Guayaquil

Telfs: (03) 2421 452 / 2421 713 / 2421 985



[www.uti.edu.ec](http://www.uti.edu.ec)

QUITO:

Dirección: Machala y Sabanilla

Telfs: (02) 3998 200/3998 201/3998 203