



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

CARRERA DE INGENIERÍA ELECTRÓNICA

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO ELECTRÓNICO**

PROYECTO TÉCNICO:

**DISEÑO DE UN GUANTE CON SENSORES DE FLEXIBILIDAD QUE
TRADUCEN LETRAS DEL ABECEDARIO DEL LENGUAJE SORDO MUDO
UTILIZANDO MICROPYTHON**

AUTORA:

ARIANNA LISBETH MORALES ITURRALDE

TUTOR:

ING. MONICA MIRANDA RAMOS

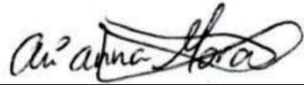
GUAYAQUIL – ECUADOR

2021

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA

Los conceptos desarrollados en el presente proyecto de titulación, el desarrollo del tema, el análisis y respectivas conclusiones de este, corresponden exclusivamente a **Arianna Lisbeth Morales Iturralde** con cédula de ciudadanía 0950395657 y el patrimonio intelectual del mismo a la Universidad Politécnica Salesiana.

Guayaquil, 17 de noviembre del 2021



Nombre: Arianna Lisbeth Morales Iturralde

Cédula 0950395657

CESIÓN DE DERECHOS DE AUTOR

Yo **Arianna Lisbeth Morales Iturralde** con cédula de ciudadanía 0950395657, manifiesto mi voluntad y cedo a la **Universidad Politécnica Salesiana** la titularidad sobre los derechos del actual proyecto de titulación en virtud de que soy la autora del tema: **“DISEÑO DE UN GUANTE CON SENSORES DE FLEXIBILIDAD QUE TRADUCEN LETRAS DEL ABECEDARIO DEL LENGUAJE SORDO MUDO UTILIZANDO MICROPYTHON”**, mismo que ha sido realizado para la obtención del título de **INGENIERO ELECTRÓNICO**, quedando la universidad facultada para ejercer y usar los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual del Ecuador, en nuestra condición de autores nos reservamos los derechos de la obra anteriormente citada.

En concordancia, suscribimos este documento en el momento que haga entrega del trabajo final en formato impreso y digital a la Biblioteca de la **Universidad Politécnica Salesiana**.

Guayaquil, 17 de noviembre del 2021



Nombre: Arianna Lisbeth Morales Iturralde

Cédula 0950395657

CERTIFICADO DE DIRECCIÓN DE TRABAJO DE TITULACIÓN

Yo declaro que bajo mi dirección y asesoría fue desarrollado el trabajo de titulación **“DISEÑO DE UN GUANTE CON SENSORES DE FLEXIBILIDAD QUE TRADUCEN LETRAS DEL ABECEDARIO DEL LENGUAJE SORDO MUDO UTILIZANDO MICROPYTHON”** con resolución de aprobación de Consejo de Carrea #354-082-2017 realizado por la estudiante **Arianna Lisbeth Morales Iturralde** con cédula de ciudadanía 0950395657, obteniendo un producto que cumple con los objetivos del diseño de aprobación, informe final y demás requisitos estipulados por la **Universidad Politécnica Salesiana**, para ser considerados como trabajo final de titulación.

Guayaquil, 17 de noviembre del 2021

A handwritten signature in blue ink that reads "Mónica Miranda". The signature is written in a cursive style and is positioned above a horizontal line.

Tutora del trabajo de titulación

Ing. Mónica Miranda

DEDICATORIA

Dedico esta tesis a Dios quien ha sido mi fortaleza, guía y ha estado conmigo en cada paso que doy. A mis padres Luis Morales y Nancy Iturralde que me han dado en todo momento en especial mi madre que ha depositado toda su confianza en cada reto que se me presentaba.

Finalmente quiero dedicar esta tesis a mis hermanos que supieron apoyarme en todo momento para que pueda terminar esta carrera.

Arianna Lisbeth Morales Iturralde

AGRADECIMIENTO

En primer lugar, estoy agradecida con mis familiares por haberme dado el apoyo en cada momento durante todo este proceso del proyecto de titulación

Agradecemos a mi tutora Ing. Mónica Miranda por su paciencia, apoyo, motivación en la realización de este proyecto.

Y finalmente doy las gracias a mis queridos compañeros y amigos de la Universidad por haber tenido el gusto de compartir toda nuestra etapa universitaria y por todos aquellos buenos y malos momentos que pasamos.

Arianna Lisbeth Morales Iturralde

RESUMEN

AÑO	TITULO	ALUMNA	DIRECTOR	TEMA DE TITULACIÓN
2020	Ingeniero Electrónico	Arianna Lisbeth Morales Iturralde	Ing. Mónica Miranda	DISEÑO DE UN GUANTE CON SENSORES DE FLEXIBILIDAD QUE TRADUCEN LETRAS DEL ABECEDARIO DEL LENGUAJE SORDO MUDO UTILIZANDO MICROPYTHON

La comunicación es un aspecto esencial en la vida diaria, la cual define a las personas como seres sociales. En este contexto, la discapacidad de las personas para hablar y escuchar constituye una barrera en el proceso comunicacional, la cual afecta cada vez más su inclusión en la sociedad.

Ello se debe a que pocas personas entienden el lenguaje de signos y señas. No obstante, el empleo de las tecnologías ha evidenciado que puede ser una vía para la solución de esta problemática.

El objetivo de la investigación es realizar el diseño de un guante con sensores de flexibilidad que traducen letras del abecedario del lenguaje sordomudo utilizando Micropython. Para ello, el estudio sigue un enfoque cualitativo, con diseño no experimental y alcance descriptivo, de tipo retrospectivo.

La investigación es guiada por un procedimiento compuesto por ocho pasos, con tiempo de duración de un año. Fue desarrollada en la Universidad Politécnica Salesiana de Guayaquil, Ecuador. Además, forma parte de un proyecto de investigación del Laboratorio de Telecomunicaciones de la universidad. Se emplean los métodos de investigación de análisis documental y modelación para su desarrollo satisfactorio.

El diseño del guante se basa en un cuero sintético y poliéster talla pequeña que equivale a 20 centímetros de largo la cual se acondicionará a la mano derecha para tener excelente comodidad, flexibilidad y adherencia en el momento de su uso. Como resultado se obtiene el diseño de un guante eléctrico, utilizando software libre, el cual puede facilitar la comunicación de personas sordomudas, favoreciendo su inclusión sin barreras comunicacionales en la sociedad.

ABSTRACT

AÑO	TITULO	ALUMNA	DIRECTOR	TEMA DE TITULACIÓN
2020	Ingeniero Electrónico	Arianna Lisbeth Morales Iturralde	Ing. Mónica Miranda	DESIGN OF A GLOVE WITH FLEXIBILITY SENSORS THAT TRANSLATE ALPHABET LETTERS FROM THE DEAF-MUTE LANGUAGE USING MICROPYTHON

In Communication is an essential aspect of daily life, which it defines people as social beings. In this context, the disability of people to speak and listen constitutes a barrier in the communicational process, which it increasingly affects their inclusion in society.

This is because few people understand sign and sign language. However, the use of technology has shown that it can be a way to solve this problem.

The objective of the research is to design a glove with flexibility sensors that translate letters of the alphabet of the deaf-mute language using Micropython. To do this, the study follows a qualitative approach, with a non-experimental design and a descriptive, retrospective scope.

The research is guided by a procedure composed of eight steps, with a duration of one year. It was developed at the Salesian Polytechnic University of Guayaquil, in Ecuador. In addition, it is part of a research project at the University Telecommunications Laboratory.

The research methods of documentary analysis and modeling are used for its satisfactory development. The glove design is based on a small size synthetic leather and polyester with 20 centimeters of length which conditions the right hand to have excellent comfort, flexibility and adherence at the moment to use.

As a result, the design of an electric glove is obtained, using free software, which it can facilitates the communication of deaf-mute people, favoring their inclusion without communication barriers in society.

ÍNDICE GENERAL

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA.....	1
CESIÓN DE DERECHOS DE AUTOR	2
CERTIFICADO DE DIRECCIÓN DE TRABAJO DE TITULACIÓN	3
DEDICATORIA.....	4
AGRADECIMIENTO.....	5
RESUMEN.....	6
ABSTRACT.....	7
ÍNDICE GENERAL.....	8
ÍNDICE GENERAL DE TABLAS	¡Error! Marcador no definido.
ÍNDICE GENERAL DE FIGURAS	¡Error! Marcador no definido.
1 Introducción	100
1.1 Descripción del problema.....	101
1.2 Importancia y alcance	11
1.3 Delimitación	112
1.4 Objetivos.....	12
1.4.1 Objetivo general.....	12
1.4.2 Objetivos específicos	12
2 Marco Teórico Referencial	12
2.1 Dactilología o alfabeto gestual.	12
2.1.1 Lenguaje de Señas	13
2.1.2 Discapacidad Auditiva.....	13
2.2 Giroscopio.....	13
2.3 Micropython	13
2.4 Sensores de Contacto.....	14
2.5 VUE JS	14
2.6 Sensor de Flexibilidad.....	14
2.7 ESP32.....	14
2.8 Pyrhon	¡Error! Marcador no definido.
3 Marco Metodológico.....	16
3.1 Fase inicial: Esquema del proyecto.....	16
3.2 Fase de aplicación de métodos	17
3.2.1 Diagrama de conexión de los sensores de contacto.	17
3.2.2 Diagrama de conexión de los sensores de flexibilidad	18
3.2.3 Diagrama de conexión del sensor de giroscopio MPU6050 ...	¡Error!
Marcador no definido.	
3.2.3.1 Diseño del Prototipo	19
3.2.3.2 Aplicación Movil Android	19

3.2.3.3	Diagrama de Flujo	19
4	Resultados	23
4.1	Prueba de funcionamiento de los sensores de flexibilidad.	23
4.2	Pruebas de funcionamiento con el MPU-6050.	25
4.3	Prueba con el lenguaje de señas.	26
4.4	Similitudes Físicas entre señas y uso del sensor de contacto.....	27
	PROYECTOS VINCULADOS.....	28
	CONCLUSIONES	29
	RECOMENDACIONES	30
	REFERENCIAS BIBLIOGRÁFICAS	31
	GLOSARIO	33
	Anexos.....	34
	Anexo A: Script para el proceso de calibración de los sensores	34
	Anexo B: Script para el proceso de reconocimiento de letras	36
	Anexo C: Presupuesto	¡Error! Marcador no definido.

1. Introducción

En la actualidad, gracias a los avances de la ciencia y la tecnología, en la medicina se han llevado a cabo disímiles descubrimientos, los cuales han propiciado una mayor calidad de vida de pacientes y familiares (Izaguirre et al., 2020; Ramírez et al., 2016). Los pacientes discapacitados, son uno de esos sectores beneficiados con tales invenciones, llegando a impactar en personas con discapacidad para hablar y escuchar, sordo-mudos como habitualmente se conocen (OPS, 2011; Pérez-Gómez, 2020).

La comunicación se puede definir como la actividad de transmitir información, por lo cual debe existir un emisor, un receptor y un canal para la transmisión de la información. Diversos son los canales que pueden ser empleados, como el intercambio de mensajes o información a través del habla, señales, escritura, imágenes o el propio comportamiento (Márquez-López et al., 2016; Meza, 2018). Sin embargo, a pesar de que existen estos canales de comunicación, las personas que tienen la capacidad de hablar usualmente prefieren utilizar el lenguaje oral, reduciendo la posibilidad de utilizar los demás tipos de canales de comunicación, o su combinación indistinta.

De acuerdo con datos de Latinoamérica, provistos por la Organización Mundial de la Salud (OMS), se prevé que el número actual de discapacitados ronda entre el 10 y 15% de la población total de la región. Dichos estudios realizados evidencian el problema que tienen estas personas para ser incluidas en diversas esferas como la económica, política, social y educativa (CEPAL, 2011). Además, el organismo refiere que aun cuando existe un aumento de la visibilidad de tal temática, la situación se sigue caracterizando por una profunda desigualdad. Asimismo, según estadísticas computadas por el Consejo Nacional para la Igualdad de Discapacidades del Ecuador, en dicho país existen alrededor de 455.829 personas con algún tipo de discapacidad. De este grupo, aproximadamente un 3.97% tiene problemas con el lenguaje, mientras que un 14.01% problemas relacionados con la audición (CONADIS, 2020).

La pérdida de la audición y del habla supone serios efectos para el aprendizaje durante la niñez y la adolescencia. Es por ello que genera grandes problemas de comunicación y adaptación en la adultez. La comunicación con personas sordo-mudas se ha facilitado en gran medida gracias al empleo del lenguaje de signos o señas con las manos. No obstante, su complejidad y la utilización de movimientos precisos han afectado su generalización. Es por ello que quienes actualmente lo conocen representa un por ciento bien reducido de la población total (Estrada, 2016).

Las personas privadas de hablar y escuchar hacen uso del lenguaje de signos y señas para poder comunicarse con la sociedad. Sin embargo, debe tenerse en cuenta que la gran mayoría de las personas que tienen la capacidad de hablar no entienden el lenguaje que los sordo-mudos emplean para comunicarse con los demás. En este contexto, el empleo de las manos es primordial, es por ello que su estudio es esencial en la elaboración de propuestas desde el empleo de la ciencia y la tecnología, que posibiliten disminuir el problema existente hoy en el proceso de comunicación con las personas sordo-mudas. El estudio de la mano humana comprende el entendimiento de su estructura anatómica, partes y movimientos, de manera que permita el desarrollo de propuestas novedosas y aplicables hoy en día. No obstante, es un área que no deja de ser compleja, ello se debe a la presencia de una elevada cantidad de nervios, arterias, músculos, tendones y huesos. Teniendo en cuenta la situación anterior, el objetivo de la investigación es realizar el diseño de

un guante con sensores de flexibilidad que traducen letras del abecedario del lenguaje sordo-mudo utilizando Micro Python.

Para darle cumplimiento al objetivo general, como parte del proyecto de investigación, los autores definen los siguientes objetivos específicos:

- Construir el guante traductor de letras del Abecedario del lenguaje de señas mediante componentes pre-fabricados.
- Obtener los valores de los sensores de Flexibilidad y acelerómetro utilizando el Microcontrolador con Micro Python.
- Convertir los valores obtenidos por el microcontrolador en letras del lenguaje sordo-mudo.
- Crear comunicación entre el Microcontrolador y el dispositivo móvil para el envío de los datos.
- Diseñar una aplicación móvil por la cual se podrá visualizar y escuchar las letras del abecedario traducidas desde el guante.

1.1 Descripción del problema

La comunicación es la actividad de transmitir información a través del intercambio de pensamientos, mensajes o información por medio del habla, imágenes, señales, escritura o comportamiento. Sin embargo a pesar de que existe estos canales de comunicación aquellas personas que tienen la capacidad de hablar prefieren utilizar siempre el lenguaje oral reduciendo la posibilidad de utilizar los otros tipos de canales de comunicación.

En Ecuador existen 455.829 personas con algún tipo de discapacidad registrados en la actualidad según la base de datos del Consejo Nacional de Discapacidades . De este grupo, aproximadamente un 3.97% tiene problemas de lenguaje y un 14.01% problemas relacionados con la audición.[5]

Las personas privadas de hablar y escuchar hacen uso del lenguaje de signos y señas para poder comunicarse con la sociedad, sin embargo hay que tener en cuenta que la gran mayoría de las personas que tienen la capacidad de hablar no entienden el lenguaje que tales personas utilizan para comunicarse con los demás. Es por este motivo que la actual propuesta de Proyecto de Titulación plantea implementar un prototipo electrónico de comunicación que permita la traducción de patrones dactilológicos (lenguaje de señas) hacia letras, de tal manera que puedan ser adquiridas e interpretadas por un computador y mostradas en una interfaz gráfica..

1.2 Importancia y alcance

La importancia de este proyecto se debe con el fin de mejorar la comunicación de las personas con discapacidad auditiva y de lenguaje ofreciendo una herramienta capaz de traducir el lenguaje de señas y así garantizar mayor independencia estimulando su interés por aprender. El alcance del proyecto será condicionado por las características de los componentes utilizados para la fabricación del mismo.

El prototipo desarrollado permite establecer una comunicación bidireccional, enviando la información a una aplicación para dispositivos móviles en donde se

visualiza y se escucha la traducción, de la misma forma, el receptor puede comunicarse con la persona discapacitada a través de mensajes que son enviados mediante una conexión por wifi y visualizados en un celular móvil

1.3 Delimitación

1. TEMPORAL

El proceso de desarrollo del proyecto tendrá un plazo normal de 12 meses a partir de la aprobación.

2. GEOGRÁFICA

El desarrollo de la propuesta se llevará a cabo dentro de la Carrera de Ingeniería Electrónica mención Telecomunicaciones en el Laboratorio de Telecomunicaciones en las instalaciones de la Universidad Politécnica Salesiana.

3. ACADÉMICA

La presente propuesta busca entregar al Laboratorio de Telecomunicaciones un guante traductor con el fin de llegar a motivar y afianzar los conocimientos utilizando el microcontrolador ESP32 y MicroPython a los estudiantes de la carrera de Electrónica sede guayaquil.

1.4 Objetivos

1.4.1 Objetivo general

Diseñar un guante con sensores de flexibilidad que traducen letras del abecedario del lenguaje sordo mudo utilizando software libre.

1.4.2 Objetivos específicos

1. Construir el guante traductor de letras del Abecedario del lenguaje de señas mediante componentes pre-fabricados.

2. Obtener los valores de los sensores de Flexibilidad y acelerómetro utilizando el Microcontrolador con MicroPython.

3. Convertir los valores obtenidos por el microcontrolador en letras del lenguaje sordo mudo.

4. Crear comunicación entre el Microcontrolador y el dispositivo móvil para el envío de los datos.

5. Diseñar una aplicación móvil por la cual se podrá visualizar y escuchar las letras del abecedario traducidas desde el guante.

2 Marco Teórico Referencial

2.1 Dactilología o alfabeto gestual

La Grupo de señas para representar las letras del alfabeto con el que se escribe la lengua oral del país. Las personas instruidas que tienen discapacidad auditiva y de lenguaje lo emplean para lograr una comunicación efectiva (ONSIPD, 2011).

Tiene 30 configuraciones manuales para simbolizar las letras, las cuales se muestran en la Figura 1.

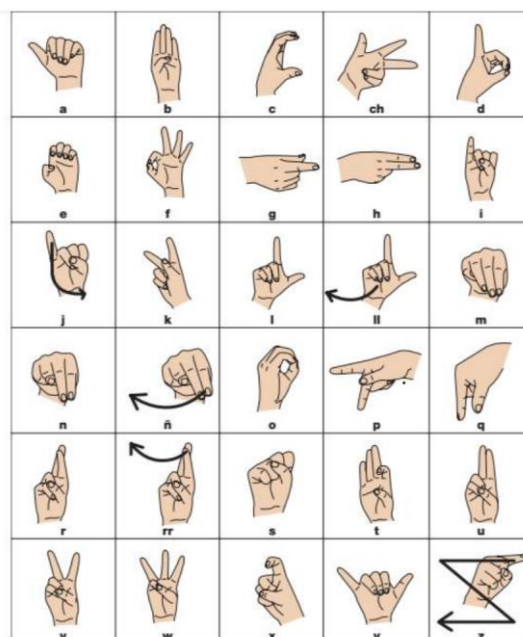


Figura 1. Alfabeto Dactilológico Universal. Fuente: (Estrada, 2016).

2.1.1 Lenguaje de señas

Es una lengua natural que se basa en expresiones gesto-espaciales, movimientos corporales y de percepción visual. Tiene un código y estructura propia, conformada por seis parámetros: configuración, localización, movimiento, orientación, dirección y expresión facial. Mediante ellos, una persona se puede comunicar fácilmente con otra (Geovanni & Diego, 2017).

2.1.2 Discapacidad Auditiva

La discapacidad auditiva se refiere a la falta o disminución en la capacidad para oír debido a un problema en algún lugar del aparato auditivo (INP, 2006).

Tecnologías y herramientas de hardware y software empleadas

2.2 Giroscopio

Es un sensor de medida inercial que permite calcular la velocidad de rotación sobre su propio eje. Para la elaboración del sistema electrónico se requirió de un giroscopio que permitiera obtener la inclinación del brazo en los ejes XY (Omega, 2020).

2.3 Micropython

Es una implementación del lenguaje de programación Python, que está orientada a dispositivos con muy pocos recursos, como pueden ser microcontroladores. Micropython se compone de un compilador y un intérprete en tiempo de ejecución que funciona en el hardware de un microcontrolador (La Jaquería, 2019).

Se utilizará Thony que es el entorno de Desarrollo Integrado de Python el cual nos permite programar el ESP32.

2.4 Sensores de Contacto

Con este sensor se podrá medir el contacto entre los dedos. Este sensor funcionará como un pulsador conectado al ESP32. Al momento que se juntan los sensores se genera un circuito cerrado generando un pulso al ESP32 .

2.5 VUE JS

Es un Framework progresivo para construir interfaces de usuario. A diferencia de otros Frameworks monolíticos, Vue está diseñado desde el inicio para ser adoptado incrementalmente. La biblioteca principal se enfoca solo en la capa de la vista, y es muy simple de utilizar e integrar con otros proyectos o bibliotecas existentes. Por otro lado, Vue también es perfectamente capaz de soportar aplicaciones sofisticadas de una sola página (Conatel, 2020).

2.6 Sensor de Flexibilidad

El sensor flexible es un elemento piezoeléctrico cuya resistencia varía a medida que el sensor se dobla. Este tipo de sensor es utilizado para guantes y para movimiento de articulaciones en robots.

Con estos sensores se podrá medir la flexión generada por los dedos. El sensor Flex funciona como un potenciómetro el cual mediante la flexión de los dedos varía el voltaje generado. Con este voltaje generado se podrá obtener que tan flexionado están los dedos. Cada dedo cuenta con un sensor de flexibilidad, para tener una lectura independiente de cada dedo.

2.7 ESP32

Es una placa de desarrollo que permite hacer proyectos rápidamente, para su desarrollo cuenta con gran variedad de software, lenguajes de programación, marcos de trabajo, librerías y otros recursos. (Naylamp, 2020).



Figura 2. ESP32-NODEMCU-32. Fuente: (Naylamp, 2020).

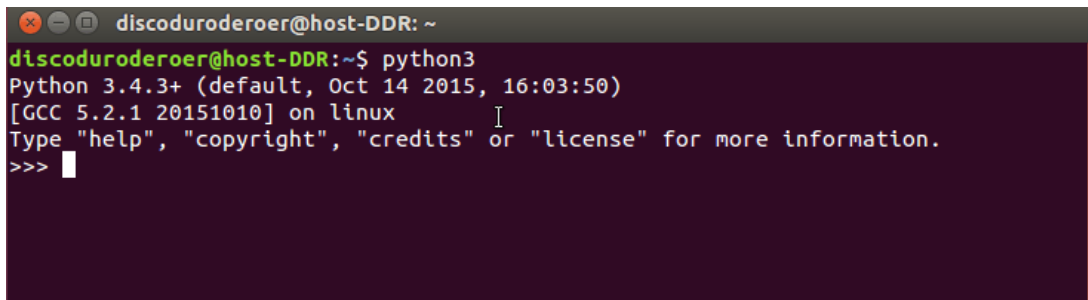
2.8 Python

Python es un lenguaje de programación con una curva de aprendizaje más corta comparada con otros lenguajes, esto hace que sea fácil, pero el ser sencillo no es un impedimento para realizar programas potentes. Tiene estructuras de datos de alto nivel eficientes y un enfoque simple pero efectivo para la programación orientada a objetos. La sintaxis y el tipado dinámico de Python, junto con su naturaleza

interpretada, lo convierten en un lenguaje ideal para el desarrollo de scripts y aplicaciones rápidas en muchas áreas en la mayoría de las plataformas.

El intérprete de Python y la extensa biblioteca estándar están disponibles de forma gratuita en forma de fuente o binario para todas las plataformas principales del sitio web de Python, y se pueden distribuir libremente. El mismo sitio también contiene distribuciones y punteros a muchos módulos, programas y herramientas de Python de terceros gratuitos, y documentación adicional.

El intérprete de Python se extiende fácilmente con nuevas funciones y tipos de datos implementados en C o C ++ (u otros lenguajes a los que se puede llamar desde C). Python también es adecuado como lenguaje de extensión para aplicaciones personalizables. La Figura 2 muestra la consola Python 3.4.

A screenshot of a terminal window with a dark background. The window title is "discoduroderoer@host-DDR: ~". The terminal shows the command "python3" being executed, followed by the Python 3.4.3+ shell prompt. The output text includes the version, date, time, and GCC version, along with instructions on how to get help. The prompt ">>>" is followed by a cursor.

```
discoduroderoer@host-DDR: ~  
discoduroderoer@host-DDR:~$ python3  
Python 3.4.3+ (default, Oct 14 2015, 16:03:50)  
[GCC 5.2.1 20151010] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>> █
```

Figura 2 Consola de Python 3.4

3 Marco Metodológico

3.1 Fase inicial: Esquema del proyecto



Figura 3 Diagrama de bloques del guante electrónico.

Fuente: elaboración propia.

El prototipo desarrollado permite establecer una comunicación bidireccional, enviando la información a una aplicación para dispositivos móviles, en donde se visualiza y se escucha la traducción, de la misma forma, el receptor puede comunicarse con la persona discapacitada a través de mensajes que son enviados mediante una conexión por Wifi y visualizados en un celular móvil.

Además, es de gran utilidad debido a que se trata de un sistema eficaz y fiable de asistencia para personas con discapacidad auditiva y de lenguaje, que permite mejorar su comunicación, desenvolvimiento e inclusión en diferentes sectores como el educativo, social y laboral.

Es importante considerar que el ESP32 es el componente principal del guante traductor en la cual cuenta con las siguientes funciones:

1. Medir sensores de Flexibilidad
2. Medir datos del giroscopio
3. Medir sensores de contacto
4. Crear servidor Web para recibir la petición GET HTTP desde la aplicación móvil

El ESP32 se lo programa con el lenguaje de programación Micropython.

El diagrama de bloques del guante electrónico se muestra en la Figura 3, el cual traduce letras del Abecedario del lenguaje sordo-mudo utilizando software libre y en la cual se utilizan sensores de flexibilidad para así lograr captar las señales de posicionamiento ocasionada en los dedos. Además, se utiliza el acelerómetro que ayuda a obtener la posición de la mano en el plano XYZ.

Entre las principales funciones y características con las que consta el guante traductor se encuentran:

- El poder traducir el lenguaje de señas a letras, a partir del desarrollo del guante eléctrico, con sensores de flexibilidad que traducen letras del abecedario del lenguaje sordo-mudo utilizando Micro Python.
- Con el desarrollo de la aplicación informática se puede escuchar y visualizar las letras, la cual consta de soporte Wifi y Bluetooth, para un correcto establecimiento de comunicación.

- La posibilidad del establecimiento de conexión inalámbrica entre el guante eléctrico y la aplicación informática.

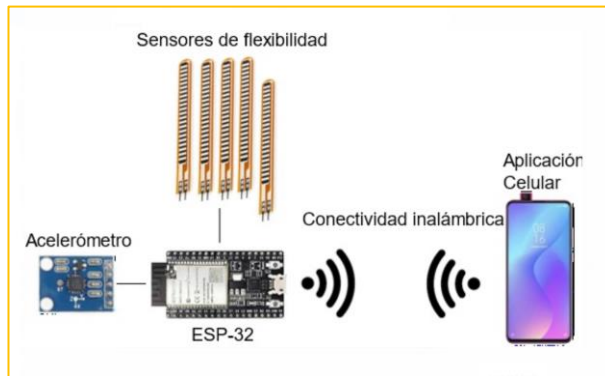


Figura 4. Conexión física del guante traductor. Fuente: elaboración propia.

En la Figura 4 se muestra el modelo de despliegue o conexión física que se establece como parte de la propuesta de solución presentada, entre la aplicación y el guante electrónico, el cual está integrado por los sensores de flexibilidad, el acelerómetro y la placa de desarrollo para ESP-32.

3.2 Fase de aplicación de métodos

3.2.1 Diagrama de conexión de los sensores de contacto.

En la figura 6 se mostrará el diagrama de conexión de los sensores de contacto, donde se puede observar que los círculos negros están conectados directamente a la tierra del ESP32 y los círculos blancos están conectados cada uno a un pin digital del ESP32, para realizar la lectura digital cuando se realice el contacto entre los círculos negros y los círculos blancos.

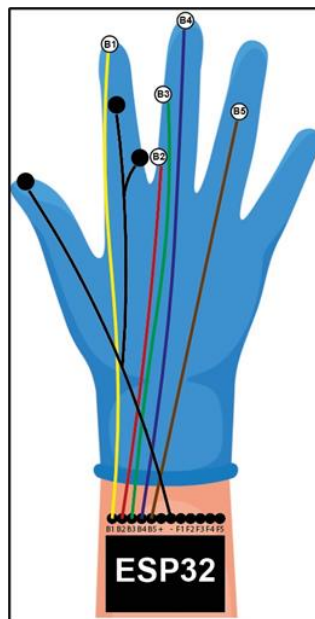


Figura 5. Diagrama de conexión de los sensores de contacto.

3.2.2 Diagrama de conexión de los sensores de flexibilidad

En la figura 7 se muestra que cada sensor de flexibilidad está conectado en el pin izquierdo con +3V del ESP32. Todos los sensores de flexibilidad están conectados cada uno a un pin analógico del ESP32, para realizar la lectura analógica del voltaje generado mediante la flexión del sensor.

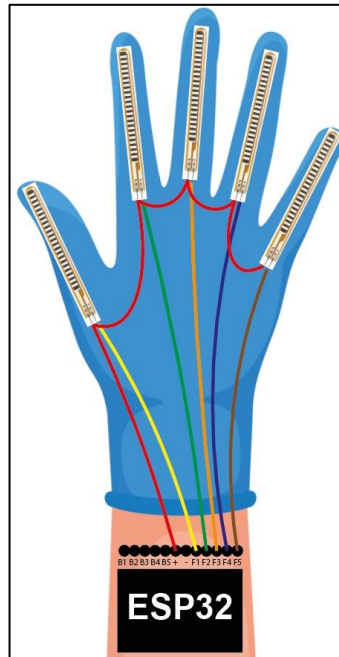


Figura 6. Diagrama de conexión de los sensores de Flexibilidad.

3.2.3 Diagrama de conexión del sensor de giroscopio MPU6050

La comunicación con este sensor es I2C. El ESP32 funcionaría como master y el módulo MPU6050 funcionaría como esclavo.

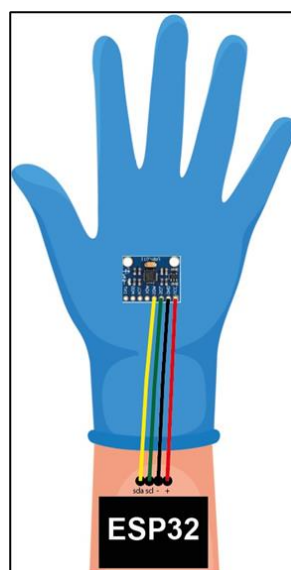


Figura 7 Diagrama de conexión del sensor de giroscopio MPU6050

3.2.3.1 Diseño del Prototipo

El material utilizado para el desarrollo de este prototipo electrónico es un guante de cuero sintético y poliéster talla pequeña que equivale de a 20 centímetros de largo para una mano derecha; la excelente comodidad, flexibilidad y adherencia que tiene este material permiten adaptar de mejor manera los sensores flexibles que son ubicados en cada dedo de la mano derecha, es decir que se tiene una excelente manipulación y coordinación de los movimientos de los dedos sin que exista un mal trato de los sensores por parte del material del guante. Además, la adquisición del guante se lo realizó en una tienda deportiva a nivel nacional, a un costo muy accesible.

A Continuación, en la figura 9 se podrá visualizar el diseño de prototipo junto con los componentes pre-fabricados.



Figura 9: Diseño del guante traductor

Se utilizó cable tipo termo contraíble para tapar las uniones que se necesitaban para las conexiones de los cables para los sensores de flexibilidad y los sensores de contacto.

En la construcción del guante se tuvo unos problemas mediante la conexión de los cables, el problema era debido a que los cables se cortan por la fricción o por que se retorció cuando se realizaban los movimientos con la mano para realizar las letras correspondientes. Para tratar de disminuir el corte de los cables se utilizó un pegamento y para que los cables no se muevan.

Se realizó el diseño de una placa electrónica impresa en baquelita de cobre, gracias el programa Eagle en el cual permitió realizar el ruteo de las pistas.

También en la pista de cobre se puede observar el microcontrolador ESP32, teniendo en consideración que la misma integra lo que es el Wi-Fi y Bluetooth por lo que no hubo necesidad de añadir un módulo inalámbrico.

3.2.3.2 Aplicación Móvil Android

El desarrollo de la aplicación se lo efectuó con el Framework Nativescript-Vue para el dispositivo Android, el funcionamiento de la aplicación se lo hace mediante las librerías Nativescript-Texttospeech el cual nos permite reproducir el texto que escribió el guante y la librería de Axios nos permite realizar la comunicación con el ESP32 mediante el protocolo GET HTTP.

En la figura 10 se observa la interfaz de la aplicación que tiene una única vista, la cual cuenta con el título del documento científico, la palabra que se formará mediante el guante traductor, botones de control (Borrar-Reproducir-Espacio) y la letra generada por el guante traductor.



Figura 10. Diseño de la aplicación móvil.

Las librerías utilizadas para la aplicación son:

Nativescript-texttospeech: Reproducir texto en voz.

- Axios -> La conexión HTTP con el módulo ESP32.

La comunicación la realiza mediante una petición GET HTTP al módulo ESP32, la cual le devuelve la letra que el guante traductor está generando.

3.2.3.3 Diagrama de Flujo

Como parte del conocimiento del procedimiento seguido para la elaboración del prototipo, se realizaron los diagramas de flujo correspondiente para explicar detalladamente la lógica detrás de un programa por lo que se mostrará una sucesión de procesos para llegar a un resultado en donde cada operación es almacenada en una estructura.

Para la calibración de los sensores se iniciará un bucle para recorrer las letras del abecedario, para que de esa manera se logre obtener los valores máximos y mínimos de cada sensor, los ángulos del giroscopio y los valores de los sensores de contacto

por lo cual en la figura 10 se detallará en diagrama de flujo del proceso de calibración de sensores

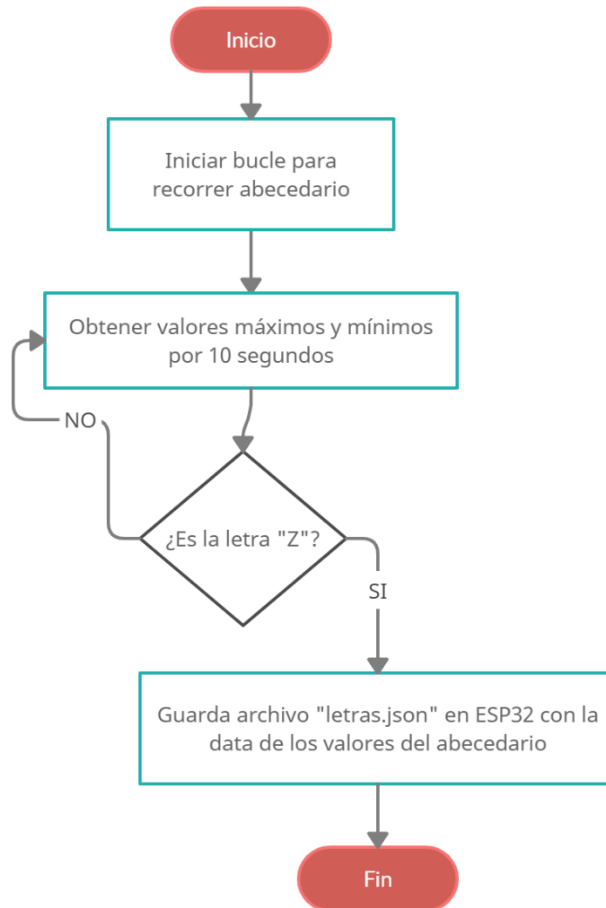


Figura 10. Diagrama de flujo del proceso de calibración de sensores.

Luego de cumplir los procesos del diagrama anterior, finalmente se podrá reconocer las letras realizadas con el guante.

En el siguiente proceso por medio de todas las mediciones realizadas, el ESP32 ejecutará validaciones por cada letra del abecedario y si llega a encontrar una coincidencia entre los datos guardados de la letra, se seteará la letra obtenida como lo indica en la figura 11.

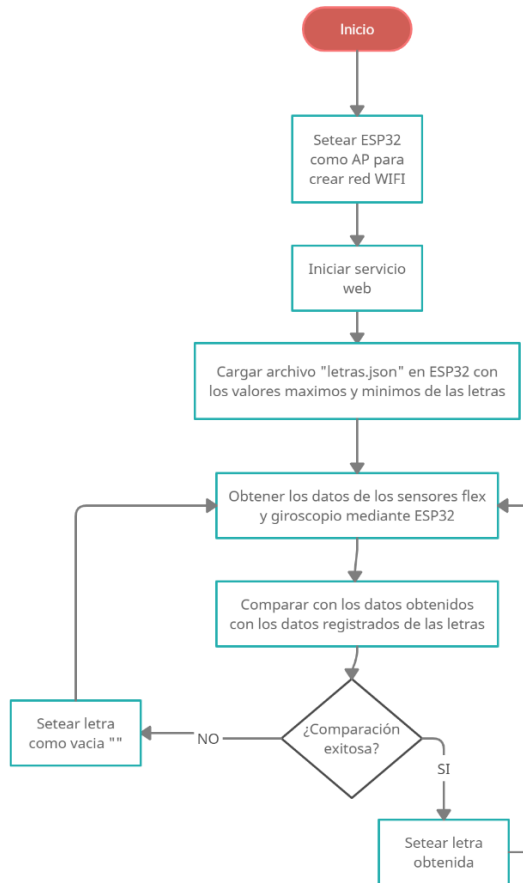


Figura 11. Diagrama de flujo del proceso de reconocimiento de letras.

En la figura 12 corresponde al Diagrama de flujo del app movil donde la comunicación la realiza mediante una petición GET HTTP al modulo ESP32.

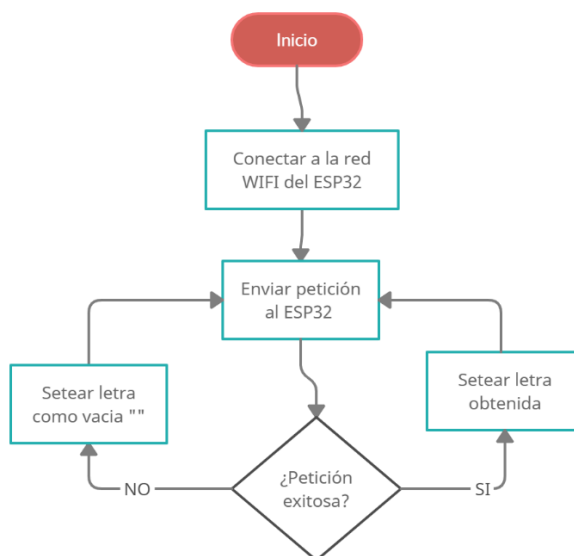


Figura 12. Diagrama de flujo del app movil.

4 Resultados

4.1 Prueba de funcionamiento de los sensores de flexibilidad

Para las respectivas pruebas de funcionamiento con los sensores Flex se va a tomar un aproximado de 60 muestras de cada uno de los sensores para detectar y determinar los rangos de valores en los que se puede manipular los datos. Como se puede observar en las siguientes figuras, se analiza el comportamiento de cada uno de los sensores Flex al momento de abrir y cerrar el puño, en la cual cada uno de los sensores permite una variación de bits amplia en todos los casos dando como resultado poder identificar la flexión que tiene los dedos de la mano.

Como se observa en la figura 13 el rango de manipulación para el dedo Meñique se encuentra desde 1333 a 2476 bits.

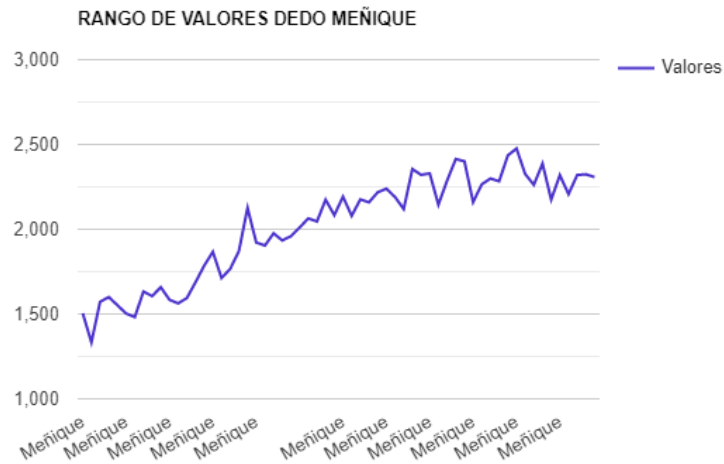


Figura 13. Rangos para el dedo Meñique

En la figura 14 el rango de manipulación para el dedo Anular se encuentra desde 1414 a 2327 bits.



Figura 14. Rangos para el dedo Anular

Como se observa en la figura 15 el rango de manipulación para el dedo Medio se encuentra desde 751 a 2268 bits.

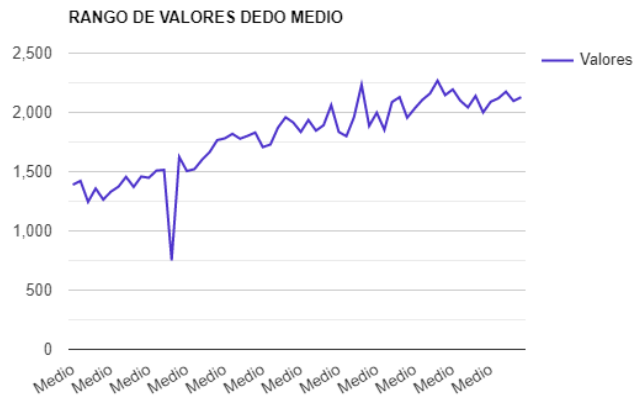


Figura 15. Rangos para el dedo Medio

Como se observa en la figura 16 el rango de manipulación para el dedo índice se encuentra desde 1267 a 2096 bits.

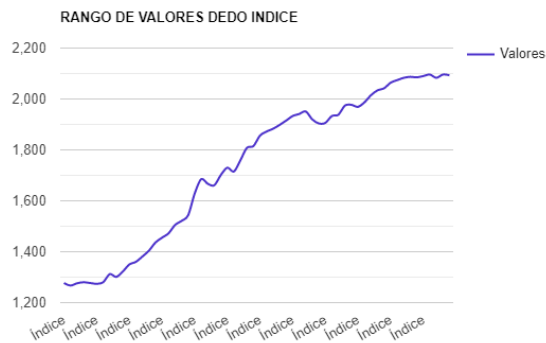


Figura 16. Rangos para el dedo Índice

En la figura 17 el rango de manipulación para el dedo índice se encuentra desde 1947 a 2194 bits.



Figura 17. Rangos para el dedo Pulgar

En base a los resultados obtenidos se realizó el siguiente grafico de barras ilustrado, el cual muestra los rangos de cada uno de los sensores Flex y así lograr a obtener una referencia al momento de realizar el lenguaje de señas. A continuación en la figura 18 se mostrará los Rangos Mínimo y Máximo de cada Dedo

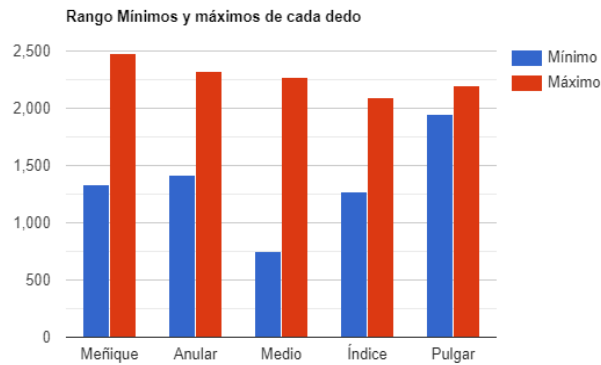


Figura 18. Rangos Mínimo y Máximo de cada Dedo.

4.2 Pruebas de funcionamiento con el MPU-6050

Es importante la adquisición de valor de los ángulos de la mano para poder obtener la identificación de la mayoría de los patrones del lenguaje de seña. Posterior a lo mencionado se realizaron pruebas de funcionamiento utilizando el MPU-6050 para obtener los ángulos esperados con su respectiva variación.

Como se observa en la figura 19 el rango de inclinación de la mano con respecto al eje X se encuentra desde -87.56 a 63.05 grados.

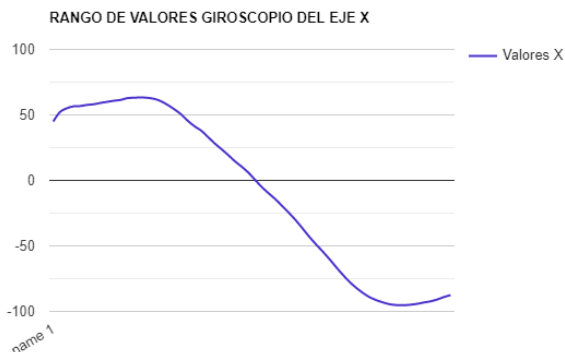


Figura 19. Rangos de valores del MPU-6050 del eje X

Con respecto a la figura 20 se observa el rango de inclinación de la mano con respecto al eje X se encuentra desde -58.70 a 43.77 grados.



Figura 20. Rangos de valores del MPU-6050 del eje Y

4.3 Prueba con el lenguaje de señas

Para la adquisición de datos primero se deberá calibrar todos los sensores para obtener los valores máximos y mínimos de cada sensor Flex, los ángulos mediante el sensor Flex y cuáles son los sensores de contactos activados, con esta calibración tendremos los valores que deben ser para cada letra.

Después de calibrar los sensores ya se podrá reconocer las letras realizadas con el guante. Aparte al momento de reconocer la letra en los sensores Flex colocamos un 10% menos en los valores mínimos y un 10% más en los valores máximos para tener un rango más amplio en los sensores de flexibilidad.

En el sensor de giroscopio no se colocó un porcentaje, pero se seteó 20°C menos al mínimo y 20°C más al máximo para dar un rango más amplio a los valores calibrados.

En los sensores de contacto no se necesita ningún porcentaje, ni valor fijo porque solo son dos únicos valores “1” ó “0”.

Mediante todas las mediciones realizadas el ESP32 tiene que ejecutar validaciones por cada letra del abecedario, si llega a encontrar una coincidencia entre los datos guardados de la letra, selecciona la letra que obtuvo la mayor coincidencia y se la devuelve a la aplicación móvil para que el usuario la pueda visualizar. A continuación, en la figura 21, se observa la posición de cada seña básica planteada en el lenguaje de señas básicas con sus respectivas posiciones de la mano. Dando como resultado formar la palabra deseada y visualizar la misma en el aplicativo móvil que se muestra en la figura 10.

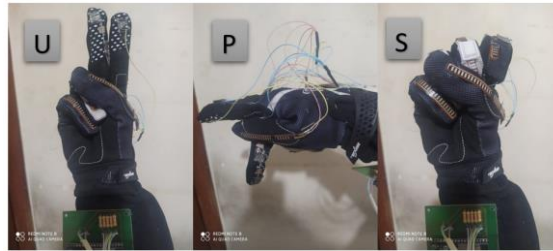


Figura 21. Lenguaje de señas

4.4 Similitudes Físicas entre señas y uso del sensor de contacto

Luego de adquirir los valores de cada letra del lenguaje de señas se procedió a realizar pruebas entre varias letras que físicamente son semejantes, revelando que sus valores de flexión son aproximados. Un ejemplo claro es la similitud entre las letras U y V. Tomando en cuenta lo sucedido se optó por utilizar los sensores de contacto con material de hilo conductor entre los dedos medio e índice, lo cuales unidos mediante cables realizaran la función de un switch al momento que se juntan los 2 nodos dando resultado un circuito cerrado.

Como resultado de las pruebas efectuadas utilizando el guante podemos concluir que se obtuvo una respuesta positiva y que el prototipo se ajusta a los requerimientos como son diseño, peso, precisión y facilidad de uso tomando en consideración que el prototipo es un dispositivo innovador y será de un gran uso para la comunidad con discapacidad auditiva.

PROYECTOS VINCULADOS

- “Diseño y construcción de un guante prototipo electrónico capaz de traducir el lenguaje de señas de una persona sordomuda al lenguaje de letras”.
- “Guante con sensores para reproducir el sonido de las letras por medio del lenguaje sordo mudo con soporte de tecnología JAVA”.
- “Diseño e implementación de una guante electrónico que permite transformar el lenguaje de señas en caracteres y reproducción sonora de voz artificial”
- “Prototipo g.t.s.b-1 (guante traductor de señas básicas), para personas con discapacidad auditiva y de lenguaje.”
- “Diseño e implementación de un prototipo para la traducción de lenguaje de señas mediante la utilización de un guante sensorizado.”.
- “Diseño e implementación de un par de guantes intérpretes del lenguaje de señas elementales a lenguaje escrito mediante software libre para facilitar el aprendizaje en la unidad educativa especializada Cotopaxi”..

CONCLUSIONES

En la actualidad, las personas discapacitadas sordo-mudas se ven afectadas cada vez más, en una sociedad que los excluye de las distintas esferas debido a su afección y la falta de comunicación, donde el lenguaje sordo-mudo es hablado por muy pocas personas. Esta situación les crea efectos negativos serios desde la niñez y en su tránsito hacia la adultez. El empleo de la ciencia y la tecnología como los sensores de flexibilidad, los acelerómetros, las placas de desarrollo para ESP-32 y los dispositivos móviles han demostrado que constituyen una vía novedosa, aplicable de manera integrada y de impacto para solucionar la barrera de comunicación que se crea en la sociedad, debido al problema que resulta en el intercambio social con personas discapacitadas sordomudas.

Se realizó la investigación de las características de cada uno de los componentes electrónicos para la construcción del guante traductor, en la cual los sensores de flexibilidad se adaptaron de la mejor manera hacia al mismo por lo que no hubo ninguna dificultad con respecto a la obtención de los rangos de valores que entrega el sensor y así mismo para el reconocimiento del lenguaje de señas.

También se empleó el uso del giroscopio MPU-6050 que a pesar de ser un componente de bajo costo, esta herramienta permite una adecuada adquisición de los parámetros de ubicación de la mano al momento de la inclinación dando resultado la obtención de los ángulos deseados. Asimismo, es importante mencionar las ventajas del hilo conductor que cumplió con los requerimientos planteados que es el hacer la función de un pulsador para así lograr obtener una mejor precisión y no haya coincidencia entre las letras que se desea ejecutar.

Unos de los componentes principales y fundamental en el proyecto es el dispositivo ESP32 debido a que cumple importantes funciones que son medir los valores de los sensores de flexibilidad, medir datos del giroscopio, medir sensores de contacto y permitir la comunicación inalámbrica desde el ESP32 hacia la aplicación móvil. El diseño del prototipo se basa en una excelente comodidad, flexibilidad y durabilidad que ayuda al buen funcionamiento y fácil adaptación para el usuario.

El guante eléctrico diseñado, con sensores de flexibilidad que traducen letras del abecedario del lenguaje sordomudo utilizando Micropython, como parte de herramientas del software libre, constituye una innovación tecnológica novedosa para enfrentar el problema existente que produce la discapacidad para hablar y escuchar, posibilitando una inclusión cada vez mayor a la sociedad, de este subgrupo de personas discapacitadas.

RECOMENDACIONES

Tomar en cuenta las siguientes recomendaciones:

- Los usuarios deben evitar realizar movimientos bruscos puesto que pueden producirse daños en las conexiones de los sensores o en los elementos electrónicos del sistema.
- Para futuros proyectos se recomienda realiza un análisis físico de robustez y flexibilidad de cables para asegurar que el dispositivo no sufra fallas en el funcionamiento
- Antes de utilizar el guante por primera vez es recomendable calibrar el guante, debido a que cada persona tiene diferentes movimientos (mano y dedos) y tamaño de mano.
- En mejoras para el proyecto se recomienda realizar reconocimiento de palabras para que las personas sordomudas se puedan comunicar más rápido en vez de estar realizando una seña por cada letra de la palabra.
- Para que la aplicación se muestre en su totalidad sin hacer ningún recorte se recomienda utilizar un celular con una resolución mayor o igual a 1080x2340 píxeles
- Se recomienda no mojar los guantes debido a que no son a prueba de agua y se pueden averiar.

REFERENCIAS BIBLIOGRÁFICAS

1. CEPAL. (2011). Las personas con discapacidad en América Latina: del reconocimiento jurídico a la desigualdad real. Comisión Económica para América Latina: del reconocimiento jurídico a la desigualdad real. [En línea]. Disponible en: <https://www.cepal.org/es/publicaciones/7135-personas-discapacidad-america-latina-reconocimiento-juridico-la-desigualdad-real>
2. ONSIPD. (2011). Discapacidad. ¿Qué es el lenguaje de señas? Oficina Nacional de Salud Integral para la Población con Discapacidad. [En línea]. Disponible en: http://www.minsa.gob.pa/sites/default/files/publicaciones/boletin_9_lenguaje_de_senas.pdf
3. OPS. (2011). Traumatismos causados por el tránsito y discapacidad. Organización Panamericana de la Salud. [En línea]. Disponible en: <https://iris.paho.org/bitstream/handle/10665.2/31080/9789275316597-spa.pdf?sequence=1&isAllowed=y>
4. CONADIS. (2020). Estadísticas de Discapacidad. Consejo Nacional para la Igualdad de Discapacidades. [En línea]. Disponible en: <https://www.consejodiscapacidades.gob.ec/estadisticas-de-discapacidad/>
5. Geovanni, B., & Diego, G. (2017). Guante electrónico para traducir de lenguaje de señas a caracteres con voz artificial y conexión inalámbrica a dispositivos móviles para personas con discapacidad auditiva y de lenguaje en la Universidad Técnica de Ambato. Ecuador. [En Línea]. Disponible en: https://repositorio.uta.edu.ec/jspui/bitstream/123456789/25193/1/Tesis_t1222ec.pdf
6. Izaguirre, L. V., Cossio, F. L., Pérez, J. F. R., & García, A. O. (2020). Impact of applications and Computer Services Developed by the University of Informatics Sciences for the health sector. *Revista Cubana de Informática Médica*, 12(1), 58-75.
7. La Jaquería. (2019). Preparando un taller de Micropython en ESP32. [En línea]. Disponible en: <https://lajaqueria.org/proyectos/2019/03/30/preparando-un-taller-de-microPython-en-esp32.html>
8. Márquez-López, J. O., Cedeño-Rengifo, R. L., & Mejía-Ruperti, L. M. (2016). Particularidades sobre la comunicación. *Dominio de las Ciencias*, 2(2), 216-228

9. Naylamp Mechatronics. (2020). Placa de desarrollo para ESP32 NodeMcu-32. [En línea]. Disponible en: <https://naylampmechatronics.com/espressif-esp/384-placa-de-desarrollo-para-esp32-nodemcu-32.html>
10. Omega. (2020). Acelerómetro - Introducción y tipos. [En línea]. Disponible en: <https://mx.omega.com/prodinfo/acelerometro.html>
11. Pérez-Gómez, S. C. (2020). Prejuicios, lenguaje y discapacidad: notas en torno a la terminología antigua y moderna relativa a las personas con discapacidad. *Cultura, Lenguaje y Representación/Culture, Language and Representation*, 23, 47-63

GLOSARIO

DISCAPACIDAD AUDITIVA: La discapacidad auditiva se refiere a la falta o disminución en la capacidad para oír debido a un problema en algún lugar del aparato auditivo.

MICROCONTROLADOR: Circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica.

SENSOR FLEXIBLE: Elemento piezoeléctrico cuya resistencia varía a medida que el sensor se dobla, es un elemento pasivo por lo que requiere de polarización para convertir una energía en otra..

ESP32: Es una serie de microcontroladores de bajo costo y consumo de energía, tiene capacidades de Wifi y bluetooth gracias al poderoso procesador de doble núcleo Tensilica Xtensa LX6..

CÓDIGO FUENTE: Conjunto de instrucciones escritas en un lenguaje de programación específico..

DACTILOLOGÍA: : Representación manual de cada una de las letras que componen el alfabeto. A través de ella la persona sorda puede transmitir cualquier palabra que desee comunicar.

Thonny: Thonny es un IDE de Python simple con características útiles para aprender programación.

HIPOACUSIA: Incapacidad total o parcial para escuchar sonidos en uno o ambos oído..

NativeScript: NativeScript es una herramienta de desarrollo de aplicaciones móviles para desarrollar su aplicación en múltiples plataformas.

Driver: programa informático que interactúa y permite al sistema operativo interactuar con un periférico, haciendo una abstracción del hardware y proporcionando una interfaz para utilizar el dispositivo.

Script: Programa ejecutable que contiene instrucciones a interpretar por otro programa. Generalmente consiste en un archivo de texto y su función habitual es realizar diversas tareas como combinar componentes, interactuar con el sistema operativo o con el usuario..

Software Libre o Abierto: Son todos aquellos programas que tienen la libertad de ser vistos, copiados, modificados, ejecutados, distribuidos y estudiados libremente. Señal

Anexos

Anexo A: Script para el proceso de calibración de los sensores

```
1 from machine import Pin, ADC, I2C
2 from time import sleep, time, ticks_ms
3 import json
4 import imu
5
6
7 data = []
8
9 letras = [
10     "A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M",
11     "N", "Ñ", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z"
12 ]
13 letras = [
14     "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12",
15     "13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26"
16 ]
17 # letras = [
18 #     "21"
19 # ]
20
21 letra_diccionario = {
22     "0": "A",
23     "1": "B",
24     "2": "C",
25     "3": "D",
26     "4": "E",
27     "5": "F",
28     "6": "G",
29     "7": "H",
30     "8": "I",
31     "9": "J",
32     "10": "K",
33     "11": "L",
34     "12": "M",
35     "13": "N",
36     "14": "Ñ",
37     "15": "O",
38     "16": "P",
39     "17": "Q",
40     "18": "R",
41     "19": "S",
42     "20": "T",
43     "21": "U",
44     "22": "V",
45     "23": "W",
46     "24": "X",
47     "25": "Y",
48     "26": "Z"
49 }
50
51 letras_guardadas = []
52 with open('letras.json') as json_file: # Lectura del archivo letras.json que se guardo cuando se realizo la calibracion
53     letras_guardadas = json.load(json_file) # colocar la data de las letras en la variable "letras"
54     print(letras_guardadas)
55
56 dedos = [
57     ADC(Pin(32)),
58     ADC(Pin(35)),
59     ADC(Pin(34)),
60     ADC(Pin(39)),
61     ADC(Pin(36))
62 ]
63
64 botones = [
65     Pin(23, Pin.IN, Pin.PULL_UP),
66     Pin(22, Pin.IN, Pin.PULL_UP),
67     Pin(21, Pin.IN, Pin.PULL_UP),
68     Pin(19, Pin.IN, Pin.PULL_UP),
69     Pin(18, Pin.IN, Pin.PULL_UP)
70 ]
71
72 maximo = None
73 minimo = None
74
75
76 for n in range(5):
77     dedos[n].atten(ADC.ATTN_11DB)
78     sleep(0.1)
79     dedos[n].atten(ADC.WIDTH_12BIT)
80     sleep(0.1)
81
82
83 angulo_y = 0
84 angulo_x = 0
85
86 imu.init_MPU()
87 imu.calibrate_sensors()
88
89
90 def conseguirValoresFlex():
91     global minimo
92     global maximo
93     valoresFlex = []
94     for n in range(5):
95         valor = dedos[n].read()
96         if valor > maximo[n]:
97             maximo[n] = valor
```

```

97     maximo[n] = valor
98     if valor < minimo[n]:
99         minimo[n] = valor
100     valoresFlex.append(valor)
101     sleep(0.1)
102     return valoresFlex
103
104 def conseguirValoresBotones():
105     data=[
106         botones[0].value(),
107         botones[1].value(),
108         botones[2].value(),
109         botones[3].value(),
110         botones[4].value()
111     ]
112     return data
113
114
115 def conseguirValoresGiroscopio():
116     global minimo
117     global maximo
118
119     (angle_x, angle_y) = imu.read_mpu6050()
120
121     if angle_y > maximo[5]:
122         maximo[5] = angle_y
123     if angle_y < minimo[5]:
124         minimo[5] = angle_y
125
126     if angle_x > maximo[6]:
127         maximo[6] = angle_x
128     if angle_x < minimo[6]:
129         minimo[6] = angle_x
130
131     print(angle_y,angle_x)
132
133
134
135
136
137 for letra in letras:
138     print("Realiza la "+letra_diccionario[letra])
139
140     sleep(5)
141
142     start = time()
143     maximo = [0, 0, 0, 0, 0, -90, -90]
144     minimo = [4095, 4095, 4095, 4095, 4095, 90, 90]
145     while time()-start <= 10:
146         valoresFlex = conseguirValoresFlex()
147         conseguirValoresGiroscopio()
148         botones_valores = conseguirValoresBotones()
149         print("Valores actuales: ",str(valoresFlex))
150         print("Maximo: ",maximo)
151         print("Minimo: ",minimo)
152         print("Contacto: ",botones_valores)
153         print()
154         sleep(0.5)
155
156     data.append([maximo,minimo,botones_valores])
157
158 ## Recorrer solo las letras seteadas en la parte de arriba ddel codigo
159 ## Guardar solo las letras seteadas
160 for element in zip(letras, data):
161     print(element)
162     letras_guardadas[int(element[0])]= element[1]
163
164 with open('letras.json', 'w') as outfile:
165     json.dump(letras_guardadas, outfile)
166
167 print("Se termino reconocimiento de letras, por favor ejecutar el otro código de sensor de letras.")
168 sleep(100)
169
170

```

Anexo B: Script para el proceso de reconocimiento de letras

```
1 # Importaciones de librerias
2 from machine import Pin, ADC
3 from time import sleep, ticks_ms
4 import time
5 import json
6 import imu
7 try:
8     import usocket as socket
9 except:
10 import socket
11 import network
12 import esp
13 esp.osdebug(None)
14 import gc
15 gc.collect()
16
17
18 # inicializar libreria IMU
19 imu.init_MPU()
20 imu.calibrate_sensors()
21
22
23 # Arreglo para los sensores de contacto que se les configura un pin digital del ESP32
24 botones = [
25     Pin(23, Pin.IN, Pin.PULL_UP),
26     Pin(22, Pin.IN, Pin.PULL_UP),
27     Pin(21, Pin.IN, Pin.PULL_UP),
28     Pin(19, Pin.IN, Pin.PULL_UP),
29     Pin(18, Pin.IN, Pin.PULL_UP)
30 ]
31
32 # Crear array para los dedos de la mano con sus respectivos Pin del ESP32
33 dedos = [
34     ADC(Pin(32)),
35     ADC(Pin(35)),
36     ADC(Pin(34)),
37     ADC(Pin(39)),
38     ADC(Pin(36))
39 ]
40
41 porcentaje_flex = 0.10 # Porcentaje de aceptacion para los sensores flex
42 valor_fijo_grados = 20 # Valor fijo de grados de aceptacion para el sensor de giroscopio
43
44 # Array vacio para las letras
45 letras = []
46 with open('letras.json') as json_file: # Lectura del archivo letras.json que se guardo cuando se realizo la calibracion
47     letras = json.load(json_file) # colocar la data de las letras en la variable "letras"
48     print(letras)
49 # Diccionario para establecer la posicion de cada letra en el array anterior de "letras"
50 letra_diccionario = {
51     "0": "A",
52     "1": "B",
53     "2": "C",
54     "3": "D",
55     "4": "E",
56     "5": "F",
57     "6": "G",
58     "7": "H",
59     "8": "I",
60     "9": "J",
61     "10": "K",
62     "11": "L",
63     "12": "M",
64     "13": "N",
65     "14": "Ñ",
66     "15": "O",
67     "16": "P",
68     "17": "Q",
69     "18": "R",
70     "19": "S",
71     "20": "T",
72     "21": "U",
73     "22": "V",
74     "23": "W",
75     "24": "X",
76     "25": "Y",
77     "26": "Z"
78 }
79
80 # Configurar las lecturas analogicas para los pines de los dedos de los sensores flex
81 for n in range(5):
82     dedos[n].atten(ADC.ATTN_11DB)
83     dedos[n].atten(ADC.WIDTH_12BIT)
84
85 # Esta función es para obtener los valores de los sensores de contacto
86 def conseguirValoresBotones():
87     data=[
88         botones[0].value(),
89         botones[1].value(),
90         botones[2].value(),
91         botones[3].value(),
92         botones[4].value()
```

```

93 ]
94 return data
95
96
97 # Esta función nos permite obtener los valores de los sensores flex mediante la lectura analogica de los pines del ESP32
98 def conseguirValoresFlex():
99     try:
100         valoresFlex=[] # Crear arreglo para los valores de los sensores flex
101         for n in range(5):
102             valor = dedos[n].read() # Se obtiene un valor entre 0 - 4095
103             valoresFlex.append(valor) # Se lo agrega al arreglo
104             sleep(0.2) # Tiempo de espera entre cada lectura analogica
105         return valoresFlex # Se retorna el arreglo con la lectura de cada sensor flex
106     except:
107         return [0,0,0,0,0] # en caso de dar error se retorna un 0 para cada sensor
108
109
110 # Obtener los grados que nos proporciona el giroscopio
111 # Utiliza una libreria para poder filtrar los valores debido a que el giroscopio a veces da valores erroneos
112 def conseguirValoresGiroscopio(data):
113     (angle_x, angle_y) = imu.read_mpu6050()
114     data.append(angle_y)
115     data.append(angle_x)
116
117     return data
118
119
120 def obtenerLetra():
121
122 def obtenerLetra():
123     data = conseguirValoresFlex() # Agregar los valores de los sensores flex
124     data = conseguirValoresGiroscopio(data) # Agregar los valores del giroscopio
125     botones_valores = conseguirValoresBotones() # Obtener los valores de los sensores de contacto y almacenarlos en una nueva variable
126     print(data)
127     print(botones_valores)
128     print()
129     for i, letra in enumerate(letras): # Recorrer todas las letras existente
130         encontrado = [False,False,False,False,False,False,False] # crear un arreglo con valores iniciales False
131         for n in range(7):
132             # Obtener valores maximos y minimos con el porcentaje de los sensores flex
133             if n >=0 and n<=4:
134                 maximo = letra[0][n]+letra[0][n]*porcentaje_flex
135                 minimo = letra[1][n]-letra[1][n]*porcentaje_flex
136             # Obtener valores maximos y minimos con el valor fijo del giroscopio
137             elif n>=5 and n<=6:
138                 maximo = letra[0][n]+valor_fijo_grados
139                 minimo = letra[1][n]-valor_fijo_grados
140
141             # Colocar en una variable el valor de la lectura al sensor correspondiente
142             valor = data[n]
143             if valor < maximo and valor > minimo: # Comprobar que la lectura actual se encuentre del del maximo y minimo de los sensores
144                 encontrado[n] = True # Si el valor se encuentra dentro los valores permitidos el valor de False se lo cambia True
145
146
147             # Validacion para ver si los valores de los sensores de contacto son iguales a los valores de las lectura actual con la lectura de la calibracion
148             botones_encontrados = False
149             if letra[2] == botones_valores:
150                 botones_encontrados = True
151                 botones_encontrados = True
152
153             # Verificar que todas las validaciones sean TRUE
154             # Si todos los valores son TRUE entonces se busca la letra en el diccionario y se la retorna
155             if not (False in encontrado) and botones_encontrados == True:
156                 print("Letra conocida: "+ letra.diccionario[str(i)])
157                 return letra.diccionario[str(i)]
158                 break
159
160     return " "
161
162
163 # Nombre y clave de la red a crear con el ESP32
164 ssid = 'MicroPython-AP'
165 password = '123456789'
166
167 # Configuraciones para el ESP32
168 ap = network.WLAN(network.AP_IF) # Configurar el ESP32 como Access Point
169 ap.active(True) # Activar AP
170 ap.config(ssid=ssid, password=password) # Configurar nombre y clave para red
171
172 # Bucle para esperar al ESP32 cuando este modo activo con la red AP
173 while ap.active() == False:
174     pass
175
176 # Impresiones por pantalla para observar la conexion exitosa
177 print('Conexión exitosa')
178 print(ap.ifconfig())
179
180
181 addr = socket.getaddrinfo('0.0.0.0', 80)[0][0][0][0] # Configurar la direccion con la IP 0.0.0.0 y el puerto 80
182 s = socket.socket() # Crear un socket para el WebServer
183 s.bind(addr) # Setear la direccion
184 s.listen(1)
185
186 header = """HTTP/1.0 200 OK
187 Content-Type: text/html; charset=utf-8
188 Access-Control-Allow-Origin: *
189
190 """
191
192 # Bucle infinito
193 while True:
194     conn, addr = s.accept() # Verifica si hay una petición por parte del cliente y guardarla en la variable "conn"
195     print('Se obtuvo un conexión desde %s' % str(addr))
196     request = conn.recv(1024)
197
198     letra = obtenerLetra() #Se llama a la función para poder obtener la letra generada por el guante
199
200     print(letra)
201     print("+++++")
202     conn.send(header)
203     conn.sendall(letra) # Enviar respuesta
204     conn.close() # Cerrar conexion
205

```

Anexo C: Presupuesto

En la siguiente tabla se muestra detallado los gastos que se han hecho para la construcción del analizador de espectro.

Tabla 1
Presupuesto del analizador de espectro

Articulo	Cantidad	Precio	Precio Total
Pc portatil core i7	1	\$ 90.00	\$ 900.00
Flex Sensor	10	\$ 20.00	\$ 200.00
Modulo ESP32	1	\$ 20.00	\$ 20.00
Conectores y resistencias	2	\$ 5.00	\$ 10.00
Estructura del guante	1	\$ 25.00	\$ 25.00
	1	\$ 90.00	\$ 90.00
TOTAL			\$ 1,245.00

Los valores antes expuestos serán cubiertos por el autor del proyecto.