

**Diseño y desarrollo de un dispositivo de rehabilitación de mano basado en estimulación
eléctrica funcional**

Jhonatan David Forero Ríos y John Wilmar Alvarez Caballero

Trabajo de Grado para optar al título de Ingeniero Electrónico

Director

Jorge Eduardo Quintero Muñoz

Ingeniero Electricista, Esp.

Codirector

Jaime Guillermo Barrero Perez

Ingeniero Electrónico, M.Sc

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones

Bucaramanga

2024

Dedicatoria

A Dios, primero agradecerle por acompañarme y guiarme por el camino correcto, por levantarme cuando ya mis fuerzas físicas se agotaban, por permitirme disfrutar al máximo cada momento de esparcimiento y conocimiento, además de conocer personas con tanto talento humano.

A mis padres por siempre permanecer al pie de mi carrera, como sostenimiento cuando habían obstáculos difíciles de superar y con la voz de aliento para continuar disfrutando el proceso de desarrollo como una persona integra.

A mi hermana Andrea, por ser el principal ejemplo a seguir, por brindarme esa mano de equilibrio para poder seguir adelante y ser la mujer ingeniera como propósito de guía.

A Nicollas, por ser esa persona paciente y aventurera, que ha sido mi bastón en los momentos donde estuve por caer, por hacer todo lo posible por sacarme sonrisas en los momentos más difíciles.

Adicionalmente, agradezco a mi alma mater la Universidad Industrial de Santander con todo su talento humano que me proporcionó para mi aprendizaje como ser humano e intelectualmente para ser un ingeniero capaz de poder dar soluciones a los problemas que se me pueden presentar, En especial a nuestro Director y codirector de este proyecto, por darnos la oportunidad de demostrar con él toda nuestra enseñanza de nuestra carrera.

John Alvarez

Dedicatoria

A Dios, primeramente por la vida y por darme la fuerza y los ánimos para afrontar todo lo que venga por delante.

A mi papa, mi mama y mi hermano, quienes en todo momento estuvieron y están a mi lado brindándome todo su amor, dandome animos, apoyándome a seguir y no darme por vencido a pesar de las dificultades, quienes aguantaron muchas noches y momentos de estrés al sentir que no lograba avanzar, pero aun asi no se dieron por vencidos conmigo y si era necesario me empujaban para seguir.

A Maria Fernanda, mi novia, quien me acompañó durante el desarrollo de mi carrera, creció junto a mi y llenó mis días con sus sonrisas y encontrando siempre la manera de hacer cada día uno diferente y mejor al anterior. Quien siempre me brinda la mano cuando lo necesito, siempre ha sido mi compañía y apoyo incondicional, quien me insistió en seguir con el trabajo en esos momentos en los que los ánimos no eran los mejores e incluso salían lágrimas por culpa de la frustración.

A mi director de proyecto, Esp. Jorge Quintero, gracias por sus enseñanzas y su guía durante el desarrollo de este proyecto, para lograr los mejores y óptimos resultados.

A Kaiser, mi perrito, quien se acostaba a dormir en mis pies acompañándome durante todas las noches de traspasado durante el desarrollo de mi carrera y muchas otras noches desarrollando este trabajo, hasta que lentamente nos dejó hace un año.

Dedicado a toda mi familia y amigos quienes no dejaban de preguntarme cuando iba a terminar. Y que ahora a pesar de estar terminando este proyecto en un país diferente a Colombia, siempre

me ayudaron y me tuvieron paciencia.

Muchas gracias a todos, este trabajo de grado va dedicado a todos ustedes con todo mi amor y esfuerzo para que por fin me puedan llamar ingeniero.

Jhonatan Forero

Agradecimientos

Los autores de este trabajo expresan sus agradecimientos:

Al profesor Esp. Jorge Eduardo Quintero, director del presente Trabajo de Grado, por su guía, paciencia y conocimiento transmitido.

Al profesor M.Sc Jaime Guillermo Barrero Perez por su disposición.

A los laboratorios pertenecientes a la E3T, por permitir el uso de sus equipos y materiales para realizar las pruebas necesarias, y a cada docente presente en él, que de una y otra manera influyó en el avance del proyecto.

Finalmente, a esta alma máter; la Universidad Industrial de Santander, por tener a disposición un gran talento humano e instalaciones, que proporcionaron tanto aprendizaje como ser humano e intelectualmente para lograr ser ingenieros.

Tabla de Contenido

Introducción	18
1. Justificación	20
2. Objetivos	21
2.1. Objetivo general	21
2.2. Objetivos específicos	21
3. Fundamentación teórica	22
3.1. La mano humana	22
3.1.1. Músculos funcionales	22
3.1.2. Nervios funcionales	24
3.1.2.1. Explicación de los movimientos de la mano	25
3.1.2.2. Explicación de los movimientos de los dedos	27
3.2. Parálisis	29
3.3. Lesiones Medulares	30
3.4. Biosensores	30
3.5. Sensores	31
3.6. Electroterapia	31

Dispositivo de rehabilitación de mano con FES	7
3.6.1. Electroestimulación	33
3.6.1.1. Tipos de Electroestimulación	33
3.6.1.2. Seguridad para el uso de Electroestimuladores	35
3.7. Estimulación Eléctrica Funcional(FES)	35
3.7.1. Definición	35
3.7.2. Aplicaciones	36
3.7.3. Componentes de un FES	36
3.7.4. Parámetros de un FES	38
3.8. Modelos comerciales de neuroprotesis FES para rehabilitacion de miembro superior	40
3.8.1. Bioness handmaster H200 wireless	40
3.8.2. XFT-2003EA (Shenzhen XFT Medical Limited, China)	41
3.8.3. Saebo Myotrac Infiniti	43
4. Diseño del dispositivo FES	45
4.1. Condiciones de Diseño	46
4.2. Unidad de control	47
4.2.1. ESP-32	48
4.2.2. Código ESP-32 Sensorial	49
4.2.3. Código ESP-32 Maestra	52
4.3. Interfaz de comunicación entre el dispositivo y el usuario	54
4.4. Diseño de la PCB	84

Dispositivo de rehabilitación de mano con FES	8
4.4.1. PCB Sensorial	84
4.4.2. PCB Maestra	89
4.5. Electroodos	94
4.6. Diseño 3D	94
5. Desarrollo del dispositivo FES	104
6. Validación	110
7. Conclusiones	117
8. Recomendaciones	118
Referencias bibliográficas	119
Apéndices	120

Lista de Figuras

Figura 1.	Músculo de la mano.	24
Figura 2.	Nervios del brazo.	25
Figura 3.	Flexión y extensión de la mano.	26
Figura 4.	Abducción radial y Abducción cubital de la mano.	26
Figura 5.	La pronación y la supinación del antebrazo.	27
Figura 6.	Acercamiento y oposición del pulgar.	28
Figura 7.	Flexión y extensión de los dedos.	28
Figura 8.	Clases de Parálisis. Nota. Diseño propio	29
Figura 9.	Bioness handmaster H200 wireless.	40
Figura 10.	XFT-2003EA.	42
Figura 11.	Saebo Myotrac Infiniti.	43
Figura 12.	Diagrama de bloques del dispositivo.	45
Figura 13.	Diagrama de bloques funcional del dispositivo.	46
Figura 14.	Estructura del código de la portada.	54
Figura 15.	Diseño de la portada.	55
Figura 16.	Estructura del código para el Botón TENS(modo terapéutico).	56
Figura 17.	Estructura del código para el Botón FES.	57
Figura 18.	Diseño de la ventana de Selección de Terapia.	58

Dispositivo de rehabilitación de mano con FES	10
Figura 19. Programación de las variables Iniciales.	59
Figura 20. Programación de inicialización del Screen FES(modos terapéuticos).	59
Figura 21. Programación del botón Conexión Bluetooth.	60
Figura 22. Programación del botón Desconexión Bluetooth.	61
Figura 23. Deslizador de Frecuencia.	62
Figura 24. Deslizador de Ancho de pulso.	62
Figura 25. Deslizador de tiempo.	62
Figura 26. Programación del botón Siguiente en el Screen FES(modos terapéuticos).	64
Figura 27. Programación del botón atrás en el Screen FES(modos terapéuticos).	65
Figura 28. Programación del botón Detener en el Screen FES(modos terapéuticos).	66
Figura 29. Programación del botón Volver Selección de Terapia en el Screen FES(modos terapéuticos).	66
Figura 30. Programación del reloj para el modo terapéutico.	67
Figura 31. Programación del botón de inicio para el modo terapéutico.	68
Figura 32. Programación del botón de Abrir Mano para el modo terapéutico.	69
Figura 33. Programación del botón de Cerrar Mano para el modo terapéutico.	70
Figura 34. Programación del botón de Confirmar para el modo terapéutico.	71
Figura 35. Diseño del Screen FES(modos terapéuticos).	71
Figura 36. Variables globales iniciales del Screen FES(actividades diarias).	72
Figura 37. Programación de inicio del Screen FES(actividades diarias).	73
Figura 38. Programación de botón de conectar por medio Bluetooth.	74

Figura 39.	Programación de botón de desconectar por medio Bluetooth.	74
Figura 40.	Programación de deslizador de frecuencia.	75
Figura 41.	Programación de deslizador de Ancho de pulso.	75
Figura 42.	Programación de deslizador de Ancho de pulso.	75
Figura 43.	Programación de botón Siguiente para visualizar FES(actividades diarias).	77
Figura 44.	Programación de reloj para el conteo regresivo de FES(actividades diarias).	78
Figura 45.	Programación de botón de inicio FES(actividades diarias).	79
Figura 46.	Programación de botón de detener FES(actividades diarias).	80
Figura 47.	Programación del botón de Abrir Mano para FES(actividades diarias).	81
Figura 48.	Programación del botón de Cerrar Mano para FES(actividades diarias).	82
Figura 49.	Programación del botón de Confirmar para FES(actividades diarias).	83
Figura 50.	Programación de botón de volver a la selección de la terapia.	83
Figura 51.	Diseño del Screen Fes(actividades diarias).	84
Figura 52.	Diagrama de flujo del circuito Sensorial.	85
Figura 53.	Esquema del sensor Acelerómetro.	85
Figura 54.	Esquema de circuito de protección.	86
Figura 55.	Esquema de regulador de voltaje.	86
Figura 56.	Esquema de conexiones ESP-32.	87
Figura 57.	Circuito impreso sensorial parte superior.	88
Figura 58.	Circuito impreso sensorial parte inferior.	89
Figura 59.	Diagrama de flujo del circuito maestro.	90

Dispositivo de rehabilitación de mano con FES	12
Figura 60. Esquema de regulador de voltaje.	90
Figura 61. Esquema de conexiones de ESP-32.	91
Figura 62. Circuito de potencia.	91
Figura 63. Circuito de salida para electrodos.	92
Figura 64. Circuito impreso maestro parte superior.	93
Figura 65. Circuito impreso maestro parte inferior.	94
Figura 66. Posición ideal del dispositivo sensorial.	95
Figura 67. Diseño de la Carcasa vista frontal.	96
Figura 68. Diseño de Tapa vista frontal.	96
Figura 69. Diseño en 3D de Carcasa vista frontal.	97
Figura 70. Diseño en 3D de tapa vista frontal.	97
Figura 71. Explicación del método de presión utilizado para la seguridad del dispositivo.	98
Figura 72. Visualización del dispositivo en la posición ideal.	99
Figura 73. Posición ideal del dispositivo maestro.	99
Figura 74. Diseño de la Carcasa vista frontal.	100
Figura 75. Diseño de la Tapa vista frontal.	101
Figura 76. Diseño en 3D de Carcasa vista frontal.	102
Figura 77. Diseño en 3D de la tapa vista frontal.	102
Figura 78. Visualización del dispositivo en la posición ideal.	103
Figura 79. Circuito prototipo Prueba sensorial.	104

Figura 80.	Circuito prototipo Prueba maestra.	105
Figura 81.	PCB sensorial Vista Bottom.	106
Figura 82.	PCB sensorial vista Top.	107
Figura 83.	PCB maestra vista Top.	108
Figura 84.	PCB maestra vista bottom.	109
Figura 85.	Cambio del punto muerto a extensión.	110
Figura 86.	Cambio de extensión a punto muerto.	111
Figura 87.	Cambio del punto muerto a flexión.	111
Figura 88.	Cambio de flexión a punto muerto.	112
Figura 89.	Respuesta de estímulo con frecuencia mínima 20Hz.	113
Figura 90.	Respuesta de estímulo con frecuencia máxima 40Hz.	114
Figura 91.	Respuesta de estímulo con amplitud mínima 100 μ s.	115
Figura 92.	Respuesta de estímulo con amplitud máxima 500 μ s.	116

Lista de Tablas

Tabla 1.	Efectos de la electroterapia y métodos de aplicación	32
Tabla 2.	Tipos de forma de corriente en FES.	39
Tabla 3.	Especificación técnica Handmaster H200 wireless.	41
Tabla 4.	Especificación técnica XFT-2003EA.	42
Tabla 5.	Especificación técnica Saebo Myotrac Infiniti.	44
Tabla 6.	Condiciones de diseño.	46
Tabla 7.	Comparación de especificación de microcontroladores.	47
Tabla 8.	Especificaciones de la ESP-32.	49

Lista de Apéndices

Apéndice A. Manual de usuario.	107.
Apéndice B. Códigos de programación.	115.

Resumen

Título: Diseño y desarrollo de un dispositivo de rehabilitación de mano basado en estimulación eléctrica funcional.

*

Autores: Jhonatan David Forero Ríos y John Wilmar Alvarez Caballero. **

Palabras clave: Rehabilitación, neuroprótesis, atrofia, estimulación eléctrica funcional, disfunción motriz.

Descripción: En el siguiente trabajo de grado se desarrolló un dispositivo para la rehabilitación de la disfunción motriz de la mano, que pudo ser afectada por una lesión o fallo de los nervios que inervan el órgano o por accidentes cerebrovasculares. Este dispositivo es de fácil manejo para el paciente, con una aplicación móvil para controlar las funciones como frecuencia y ancho de los pulsos, para ello se utilizó comunicación por medio de Bluetooth entre el móvil y el dispositivo generador de la corriente de estimulación eléctrica funcional (neuroprótesis FES). Este método de FES no es invasivo y es un instrumento de acompañamiento para la recuperación de las funciones de la mano, puesto que, con él se pueden hacer terapias desde casa lográndose avances con menor tiempo de tratamiento.

* Trabajo de grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y telecomunicaciones. Director: Jorge Eduardo Quintero, Especialista en telecomunicaciones. Codirector: Jaime Guillermo Barrero, Magister en ingeniería electrónica.

Abstract

Title: Design and development of a handheld rehabilitation device based on functional electrical stimulation. *

Authors: Jhonatan David Forero Ríos y John Wilmar Alvarez Caballero. **

Keywords: Rehabilitation, neuroprosthetics, atrophy, functional electrical stimulation, motor dysfunction.

Description: In the following degree work, a device was developed for the rehabilitation of motor dysfunction of the hand, which may have been affected by an injury or failure of the nerves that innervate the organ or by cerebrovascular accidents. This device is easy to use for the patient, with a mobile application to control functions such as frequency and width of the pulses, for which communication via Bluetooth was used between the cell phone and the device generating the functional electrical stimulation current (neuroprosthesis FES). This FES method is non-invasive and is an accompanying instrument for the recovery of hand functions, since it can be used to perform therapies from home, achieving progress with less treatment time.

* Trabajo de grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y telecomunicaciones. Director: Jorge Eduardo Quintero, Especialista en telecomunicaciones. Codirector: Jaime Guillermo Barrero, Magister en ingeniería electrónica.

Introducción

La función motora que implementan los órganos del cuerpo humano es fundamental para el desarrollo normal de las actividades diarias, por lo tanto, cualquier desgaste o pérdida de estas funciones afecta directamente la calidad de vida, y consecuentemente, se considera como una discapacidad. La mayoría de estas afecciones se relacionan a causa de lesiones o fallos de los nervios que inervan el órgano en cuestión, también se asocia a accidentes cerebrovasculares o resultado de la atrofia por algún accidente.

En un constante avance de la ciencia y tecnología, se ha demostrado que cualquier discapacidad se puede sobrellevar con normalidad mediante métodos para la recuperación de las funciones motoras de cualquier órgano afectado. Empezando con la rehabilitación, el caso más común, siguiendo un conjunto de terapias y ejercicios en pro de estimular el órgano afectado. Continuando con un método más invasivo; el quirúrgico, que busca por medio de una cirugía poder restablecer lo mejor posible la función del nervio afectado y así recuperar la función motora. Ahora bien, la estimulación eléctrica funcional (FES, por sus siglas en inglés) es una tecnología prometedora en el campo de la rehabilitación.

La FES implica la aplicación controlada de impulsos eléctricos a los músculos o nervios para inducir la contracción muscular y, en última instancia, mejorar la función motora. En particular, en el ámbito de la rehabilitación de la mano, la FES ofrece un enfoque innovador y efectivo para restaurar la destreza y la funcionalidad.

Con base en lo anterior, a través de un enfoque multidisciplinario que combina la ingeniería biomédica y la fisioterapia, este proyecto tiene como objetivo desarrollar un dispositivo de FES personalizado y adaptado a las necesidades específicas de cada paciente para la rehabilitación de la mano afectada por problemas neurológicos o por traumas craneoencefálicos.

La rehabilitación de la mano con dispositivos de FES tiene el potencial de transformar la vida de las personas, permi-

tiéndoles recuperar la independencia y la capacidad de realizar tareas cotidianas con mayor facilidad, comodidad, y al alcance de todos.

1. Justificación

Actualmente en Colombia, según estudios del Ministerio de Salud, aproximadamente el 7% de la población se ve afectada por discapacidades en uno o más órganos de su cuerpo, de este porcentaje, alrededor del 38% padece de pérdidas de la función muscular, parálisis o espasticidad en sus manos, brazos o piernas (de Salud y Protección Social (2019)); esto debido a tumores, enfermedades cerebrovasculares, afecciones en el sistema nervioso central o como resultado de un accidente. Estas condiciones alejan a las personas que las padecen del desarrollo normal de su diario vivir, ya que dificulta la realización de actividades y tareas diarias al no tener la capacidad de realizar movimientos precisos (Zapata Vélez de Villa (2010)), lo que conlleva en muchas ocasiones a cambiar su estilo de vida, hasta el punto de quedar excluidas del mercado laboral, y en algunos casos, aisladas de la comunidad.

Para la rehabilitación de dichas discapacidades existen tratamientos como la terapia ocupacional, intervenciones quirúrgicas o la electroterapia, dentro de esta última se destaca la estimulación eléctrica funcional debido a su practicidad y buenos resultados en pacientes (Quintero Gonzalez and Ferrer Betancur (2015)). Ahora bien, enfocados en la discapacidad de la mano se realizará un prototipo de un dispositivo de rehabilitación de la misma, el cual usará la estimulación eléctrica funcional para realizar el tratamiento y así iniciar o continuar con la restauración de la función motora de la mano de los pacientes.

2. Objetivos

2.1. Objetivo general

- Diseñar y desarrollar un dispositivo para la rehabilitación de mano basado en estimulación eléctrica funcional.

2.2. Objetivos específicos

- Implementar un prototipo ajustable de estimulador eléctrico funcional para la rehabilitación de mano.
- Desarrollar una aplicación móvil para sistema operativo Android, con una interfaz de usuario de fácil interacción, que permita el manejo remoto del dispositivo de rehabilitación basado en FES.
- Elaborar un prototipo de fácil manejo para los usuarios y apto para el uso diario de pacientes en la rehabilitación de mano.

3. Fundamentación teórica

3.1. La mano humana

Es un órgano principal para los movimientos esenciales en las rutinas diarias de cada persona, como lo es la presión y la manipulación de alimentos. Se compone de 44 unidades músculo-tendón, 27 huesos y 18 articulaciones resultando en torno de 27 grados de libertad. La función de esta estructura está regida por el trabajo sincronizado de varios músculos, que se aplican sobre el puño o los dedos. La mayoría de ellos se origina, principalmente, en el antebrazo, pasa por sobre el puño y los huesos cárpicos para insertarse en los dedos, lo que los convierte en músculos pluriarticulares, los músculos del puño, como parte integral de la función de la mano, terminan por mantener los músculos de los dedos en un tamaño favorable para la producción de la tensión (Zapata Vélez de Villa (2010)).

3.1.1. Músculos funcionales.

Estos músculos son importantes para restaurar la función de la mano en personas que han experimentado lesiones neuromusculares o daño en los nervios. La FES implica el uso de impulsos eléctricos para estimular estos músculos y ayudar en la rehabilitación y recuperación de la función de la mano. Cada músculo puede ser estimulado selectivamente para promover movimientos específicos de la mano y mejorar la destreza y la funcionalidad, a continuación, una lista respectiva de ellos y su esquema en la Figura 1.

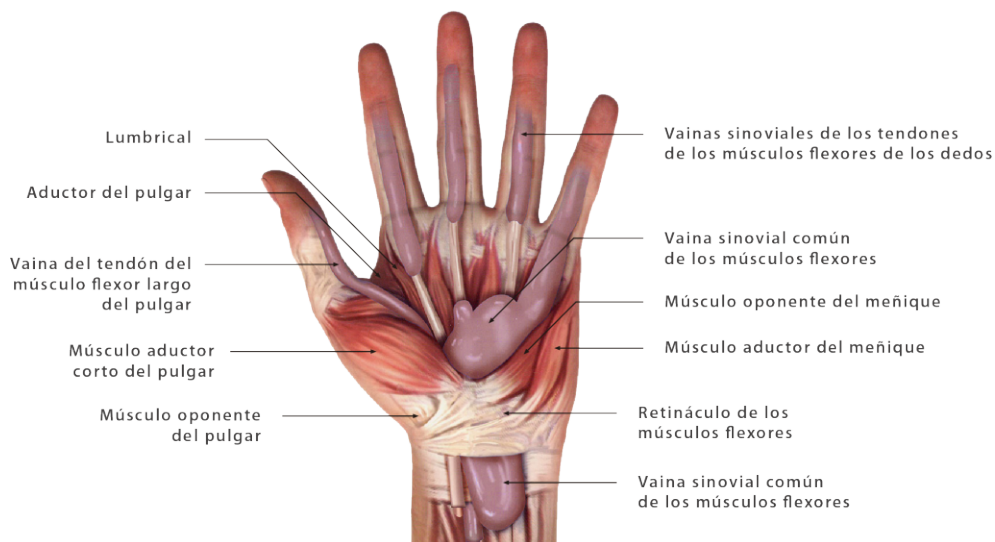
- Músculos flexores:

- El pronador redondo: este músculo se encuentra en el antebrazo y es responsable de pronar o girar la palma hacia abajo. Se utiliza en actividades como girar una llave.
- El flexor común superficial: se encuentra en el antebrazo y es esencial para flexionar las articulaciones de la muñeca y los dedos. Ayuda en acciones como cerrar la mano.
- El flexor común profundo: este músculo trabaja junto con el flexor común superficial para flexionar las articulaciones de la muñeca y los dedos.

- El flexor largo del pulgar: como su nombre lo indica, este músculo flexiona la articulación del pulgar, lo que permite doblar el pulgar hacia la palma de la mano.
- Músculos extensores:
 - El radial largo del carpo: se encarga de extender la articulación de la muñeca, lo que permite mover la mano hacia arriba.
 - El radial corto del carpo: trabaja junto con el radial largo del carpo en la extensión de la muñeca.
 - Cubital del carpo: extiende y abduce la muñeca, permitiendo que la mano se desplace hacia el lado nervio cubital de la misma.
 - Extensor de los dedos: estos músculos extienden las articulaciones de los dedos, lo que permite abrir la mano y los dedos.
 - Extensor del índice: específicamente extiende la articulación del dedo índice.
 - Extensor del meñique: extiende la articulación del dedo meñique.
 - Separador largo del pulgar: ayuda a alejar el pulgar de los otros dedos, lo que se conoce como abducción del pulgar.
 - Extensor corto del pulgar: extiende la articulación del pulgar.
 - Extensor largo del pulgar: trabaja en conjunto con el extensor corto del pulgar para extender la articulación del pulgar.
- Músculos de la muñeca:
 - Músculos laterales; el del pulgar, el flexor corto, el oponente y el aductor. Estos músculos producen una prominencia en la mano llamada eminencia tenar.
 - Músculos internos, los cuales son el del dedo meñique, el flexor corto y el oponente

- Músculos de la región palmar media, que están situados en dos planos y son: los músculos superficiales o lumbricales, y los músculos profundos o interoseos palmares y dorsales

Figura 1.
Músculo de la mano.



Nota. Adaptado (Zapata Vélez de Villa (2010))

3.1.2. Nervios funcionales.

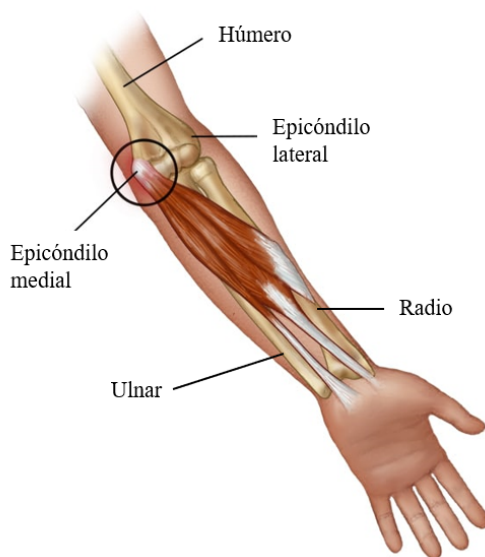
Además de los músculos que desempeñan un papel fundamental en cualquier movimiento de la mano, es importante destacar la presencia de nervios encargados de transmitir las señales eléctricas necesarias para activar estos músculos y coordinar su funcionamiento. Estos nervios cumplen una función vital al conectar el sistema nervioso central con los músculos periféricos, permitiendo así la ejecución de movimientos precisos y coordinados.

El cerebelo, una estructura cerebral clave, desempeña un papel crucial en la mejora de la precisión del movimiento. Lo logra al comparar las señales de las órdenes motoras descendentes, es decir, las instrucciones del cerebro para realizar un movimiento específico, con la información sobre la acción motora resultante. El cerebelo realiza ajustes y correcciones necesarios actuando tanto sobre el tronco cerebral como sobre las áreas motoras corticales que se

comunican directamente con la médula espinal. Este proceso de supervisión y corrección es fundamental para garantizar un movimiento suave y preciso.

Cabe resaltar que la proyección directa desde el córtex motor primario hasta las neuronas motoras que controlan los músculos más distales, como los de las manos y dedos, constituye la única vía por la cual el cerebro puede controlar de manera individualizada estos músculos. Esta conexión permite una manipulación fina y coordinada de los movimientos de las manos y dedos, lo que resulta esencial en una variedad de actividades cotidianas que requieren destreza y precisión. (Zapata Vélez de Villa (2010))

Figura 2.
Nervios del brazo.



Nota. Adaptado (Zapata Vélez de Villa (2010))

3.1.2.1. Explicación de los movimientos de la mano. Las acciones que realiza la mano

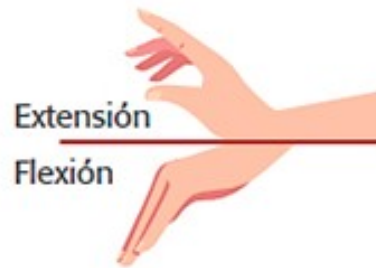
son:

- La flexión y la extensión consiste en acercar y alejar, respectivamente, la palma a la cara anterior del antebrazo.
- La abducción radial es la inclinación hacia el lado radial y la abducción cubital es la inclinación hacia el lado cubital

- La pronación y la supinación es el movimiento del antebrazo que hace girar la mano de fuera-dentro y dentro-fuera, respectivamente, presentando el dorso de ella.

Figura 3.

Flexión y extensión de la mano.



Nota. Adaptado (Zapata Vélez de Villa (2010))

Figura 4.

Abducción radial y Abducción cubital de la mano.



Nota. Adaptado (Zapata Vélez de Villa (2010))

Figura 5.

La pronación y la supinación del antebrazo.



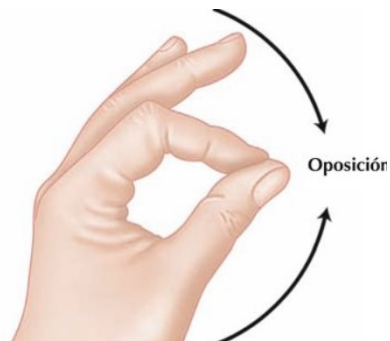
Nota. Adaptado (Zapata Vélez de Villa (2010))

3.1.2.2. Explicación de los movimientos de los dedos. Los dedos son componentes esenciales de nuestras manos y realizan una serie de movimientos cruciales para llevar a cabo tareas cotidianas y actividades más complejas. Estos movimientos se pueden dividir en tres categorías principales, cada una con su propia función distintiva:

- **Movimiento de oposición/acercamiento del pulgar a los dedos:** este movimiento se refiere a la capacidad única del pulgar de moverse hacia los otros dedos de la mano. Es esencial para agarrar y sujetar objetos con precisión, lo que permite tareas delicadas como escribir, sostener objetos o abotonar una camisa. Este movimiento es posible gracias a una articulación especial en la base del pulgar, conocida como la articulación carpometacarpiana del pulgar.

Figura 6.

Acercamiento y oposición del pulgar.

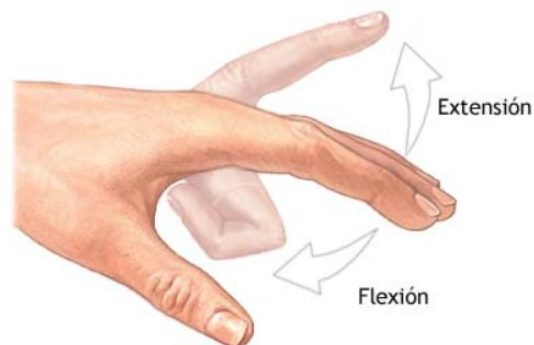


Nota. Adaptado (Zapata Vélez de Villa (2010))

- Movimiento de flexión/extensión de los dedos: este tipo de movimiento implica doblar y enderezar los dedos. La flexión ocurre cuando los dedos se curvan hacia la palma de la mano, mientras que la extensión es el movimiento contrario, donde los dedos se estiran para alinearse con la mano. Estos movimientos son esenciales para realizar una variedad de actividades, desde escribir en un teclado hasta recoger objetos del suelo. Los músculos flexores y extensores de los dedos trabajan en conjunto para controlar estos movimientos y proporcionar destreza y fuerza a las manos.

Figura 7.

Flexión y extensión de los dedos.



Nota. Adaptado (Zapata Vélez de Villa (2010))

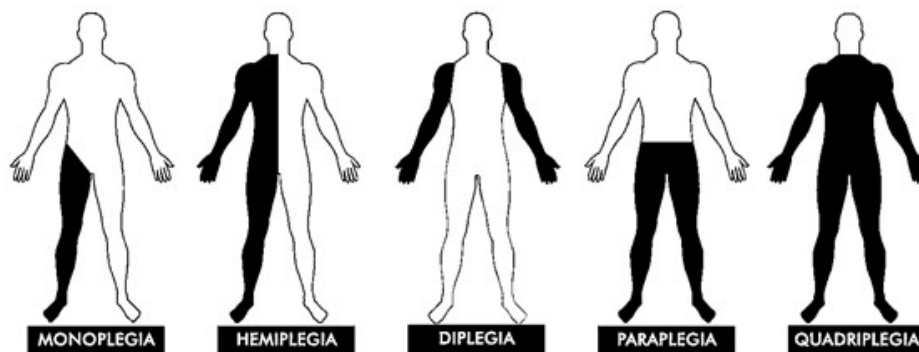
3.2. Parálisis

Dado el contexto del proyecto, es necesario hacer una breve explicación de algunas anomalías de las funciones motoras del cuerpo humano, ya que se hablará de un dispositivo tratante a una disfunción motora de la parte superior, como es la mano.

La parálisis es una pérdida o disminución de la movilidad o de la contractilidad de los músculos en relación con alteraciones, congénitas o adquiridas, de las vías nerviosas o de los propios músculos (Aguar (2016)). Tal es el caso del paciente que genera una independencia de las personas que lo rodean, ya que se verá afectada las capacidades motoras para poder realizar una función cotidiana. Existen algunas clases de parálisis, esta clasificación depende de algunos factores como la duración del tiempo y los grupos musculares afectados. Así mismo, se denomina parálisis localizada la que afecta una área pequeña del cuerpo o generalizada la que afecta gran parte del cuerpo humano.

Figura 8.

Clases de Parálisis. Nota. Diseño propio



Nota. Diseño propio

Como se visualiza en la fig.8 hay 4 clases tales como Monoparesia(un miembro), Diplejia o parálisis bilateral (afecta los miembros simétricos), Hemoplejia (afecta los miembros de un lado del cuerpo), tetraparesia o cuadraplejia (afecta los 4 miembros), Ahora bien, cualquiera de estas, pueden ser temporales o permanentes, ya depende de donde radica la lesión puede ser en la neurona central o en la del asta anterior de la médula.

3.3. Lesiones Medulares

Se refiere a los daños sufridos en la médula espinal debido a un traumatismo o de una enfermedad degenerativa, este es una de las causas más frecuentes por la que se genera una parálisis, puesto que el daño se ocasiona en el nervio motor, por lo consiguiente con este proyecto se intenta proyectar una comunicación alternativa entre el cerebro y los músculos.

Cabe destacar que la mayoría de las personas que sufren de esta anomalía es por accidentes ya sea deportivos o automovilísticos, otra de las causas menos probables es un impacto de bala o un corte del nervio, pues dado que con ello se puede almacenar hemorragias internas que estas afectan a la integridad medular, aunque no podemos solo hablar de lesiones externas, también se pueden afectar músculos internos, como lo es el control del esfínter, que son poco perceptibles al observador.

También se debe mencionar que, las lesiones medulares pueden asociarse con un riesgo de padecer diversas enfermedades que pueden resultar potencialmente mortales como la trombosis venosas profundas, osteoporosis y úlceras de decúbito, hasta puede llegar el paciente a sufrir enfermedades respiratorias. A nivel mundial, se registran entre 250.000 y 500.000 casos de lesiones de este tipo, anualmente (Aguar (2016)); también se menciona que esto esta en incremento y ahí es conveniente traer a colación que existen servicios de atención llamados rehabilitación y mantenimiento de la salud como, por ejemplo, la prevención y tratamientos para estas afecciones.

En relación a la problemática expuesta, se han realizado estudios con el fin de que se pueda evitar una dependencia de por vida de pacientes que padecen de cualquier parálisis y para que en un futuro estas personas puedan llevar una vida a cabalidad, estos proyectos basados en esta afección, va desde prevenciones hasta la rehabilitación de la función motora de los miembros afectados(Quintero Gonzalez and Ferrer Betancur (2015)).

3.4. Biosensores

La adquisición de datos emitidos por alguna fuente generadora de ellos, permite poder controlar algunas funciones de objetos, en el ámbito biomédico estos datos se llaman bioseñales; se le conoce así, ya que son emitidas por sistemas fisiológicos del organismo. Estas mediciones que se pueden realizar a estos sistemas, tiene como objetivo

extraer información necesaria sobre los diferentes funcionamientos del organismo. Cabe recalcar que, dependiendo del estilo probabilístico de la toma de las muestras, se pueden agrupar en diferentes grupos, tales como las señales biomagnéticas, señales bioquímicas, señales biomecánicas, entre otras.

Algunos tejidos celulares, tienen la capacidad de generar potenciales bioeléctricos, debido a la actividad electroquímica que se realiza en las membranas de sus células, a ellas se les conoce como células excitables. Esto se puede observar en los tejidos musculares, nerviosos y glandular. (Guerrero Martinez (2011)).

3.5. Sensores

Los dispositivos electrónicos que permite obtener información del mundo físico, tal como, imitar la capacidad de las percepciones de los seres humanos, a estos dispositivos se le conoce como Sensores. Estos elementos tienen funciones que para la implementación y el desarrollo de dispositivos bioeléctricos, ya que toma datos del entorno y se transmiten a las aplicaciones que procesan estos datos para la toma de decisiones precisas y rápidas para situaciones cambiantes o condiciones específicas. (Serna et al. (2010)) Hay sensores de electromiografía que tienen doble funcionalidad como la adquisición o envío de señales. Estos sensores llamados Electroodos funcionan al tener una superficie conductora que es puesta en contacto con el órgano más grande del cuerpo humano, que es la piel. Se debe tener en cuenta, que se debe colocar en la parte de interés o la que recibirá o emitirá alguna señal, en este caso, el músculo a potenciar.

3.6. Electroterapia

Es denominada como la ciencia del tratamiento de afecciones por medio de la electricidad, también se considera como una disciplina que esta englobada en la fisioterapia. Como se indica es un tratamiento no invasivo que utiliza la técnica de aplicar corriente eléctrica al cuerpo humano efectivas para mejorar patologías de estos padecimientos. Existen muchos tipos de clasificaciones de electroterapia, según su forma (constantes y variables), según su polaridad (alternante o bifásicas, constante o unidireccional), etc. según su frecuencia, es decir, según el número de ondas que hay en cada segundo incluyendo la pausa en corrientes interrumpidas. Su unidad es el Hertz[Hz].(Cruz (2018))

En la siguiente tabla se visualiza un poco de las aplicaciones de los estímulos para la afección o la respuesta

que se trata al administrar la corriente con la frecuencia y forma de onda que se indica en cada ítem.

Tabla 1

Efectos de la electroterapia y métodos de aplicación

Respuestas genéricas de la electroterapia	Aplicación de estímulos eléctricos	Técnica empleada
Respuestas Motoras	Pulsos aislados (exploración y parálisis) Vibración muscular (analgesia y relajación) Contracción en ráfagas (relajar, fortalecer y elongar)	Baja Frecuencia (pulsos aislados de forma triangular o rectangular, ondas farádicas y trenes de alto voltaje). Media frecuencia modulada para vibraciones y trenes de media frecuencia (ondas de Kotz)
Respuestas Sensitivas	Neuroanalgesia sin polaridad	Baja frecuencia (TENS, formación de la corriente deseada triangular/cuadrada). Media frecuencia (MF) modulada y sin modular
Cambios electroquímicos	Galvanismo Iontoforesis Sensitivo con componente galvánico	Baja frecuencia con galvánica e interrumpidas galvánicas
Aporte energético	Cambios metabólicos en los tejidos	Alta frecuencia, Ultrasonido, Magnetoterapia, Luz UVA y LASER, Galvánica de LF, Moduladas con onda portadora de HF

Nota. Adaptado Aguar (2016).

En la actualidad, la Electroterapia se está aplicando en varios campos de la fisioterapia y se ha desarrollado múltiples dispositivos sin correr con riesgos secundarios. Los tratamientos de la aplicación de esta terapia son (Portillo Martínez (2010)): anti-inflamatorio, analgésico, mejora del trofismo, potenciación neuro-muscular y el térmico en el caso de la electroterapia de alta frecuencia.

3.6.1. Electroestimulación.

En el pasado, la rehabilitación se consideraba generalmente como la etapa final en el proceso de atención médica. Las principales áreas de enfoque eran la prevención y el tratamiento de las afecciones, y la rehabilitación se veía como una medida para abordar las complicaciones resultantes o secuelas. Sin embargo, en la actualidad, esta perspectiva ha evolucionado significativamente. Se ha producido un cambio hacia un enfoque más preventivo, que busca reducir las complicaciones asociadas con períodos prolongados de reposo y aprovechar los avances tecnológicos para minimizar la gravedad de las secuelas derivadas de daños orgánicos (Flores Ramos (1998))

En este contexto, la electroestimulación se erige como un principio fundamental que se basa en la administración controlada de corrientes eléctricas con el propósito de estimular músculos, nervios y tejidos específicos en el cuerpo, con el fin de inducir respuestas fisiológicas específicas. El fundamento subyacente es que estas corrientes eléctricas pueden generar contracciones musculares y, en ciertos casos, influir en la función nerviosa. Este proyecto tiene como objetivo principal contribuir a la rehabilitación motora muscular. Se plantea como un complemento a las terapias físicas guiadas por terapeutas, proporcionando un enfoque adicional en la estimulación motora. (Ambrosio et al. (2015))

3.6.1.1. Tipos de Electroestimulación. Como se refleja en la tabla anterior, hay derivaciones de la electroestimulación, dado a esto, se debe explicar los métodos que al paso del tiempo se han utilizado, principalmente han sido más para aliviar o aumentar el músculo, como aplicación deportiva. De esta manera, se debe aclarar que cada tipo de E.E tiene sus características para su tratamiento, desde la forma de onda hasta el efecto que se requiere sobre el cuerpo del paciente. A continuación se dará una breve aclaración de cada variedad de electroestimulación:

- Corriente de Kotz (ondas rusas): Este tipo de EE es muy práctico para el uso deportivo, ya que está destinado a potenciación muscular intensa, el aumento del tamaño del músculo y la resistencia; como se ha indicado, actualmente, por su uso se ha comercializado como altamente tonificante y fines estéticos. Para poder utilizar este método es con una corriente de naturaleza bipolar con ondas rectangulares sinusoidales de alta frecuencia.

- Corriente galvánica continua: Es el primer tipo de corriente que se comenzó a utilizar dentro de la electroterapia como técnica de fisioterapia, se usa cuando queremos generar efectos químicos o también llamados polares (efectos ácidos o alcalinos) en la zona próxima al electrodo. Cuando se elige usar esta corriente en forma terapéutica, se busca sacar provecho de estos efectos químicos. Debe tenerse mucho cuidado en su uso, ya que puede generar quemaduras en la piel si no se administra correctamente, por lo que no se recomienda para uso hogareño. (Aguar (2016))
- Transcutaneous Electrical Nerve Stimulation (TENS): También conocido como electroanalgesia, esta técnica busca aliviar el dolor articular parcial o permanentemente. La TENS se encarga de entregar pulsos eléctricos cuadrados unidireccionales con corriente monofásica o bifásica. Si se aplica correctamente este método no habrá contracción muscular, sino se sentirá solo cosquilleo estimulando las fibras sensitivas; hay dos tipos de TENS:
 - Convencional de baja intensidad y alta frecuencia (60-100 Hz).
 - De alta intensidad y baja frecuencia (menos de 10 Hz).
- Electroestimulación neuromuscular (EENM): Esta práctica de EE tiene un poco el mismo resultado de la corriente galvánica, ya que lo que busca es fortalecer el músculo, mantener y aumentar la masa muscular, puesto que, esta técnica estimula un nervio o músculo hasta lograr una contracción muscular inducida por los impulsos eléctricos. Es otro método que utilizan los entrenadores para la estimulación de los deportistas.
- Electroestimulación funcional (FES en inglés): Es una de la más complejas, ya que el objetivo de esta técnica no sólo es mantener y fortalecer la musculatura afectada, sino que también generar movimientos funcionales, es decir que cumplan un propósito (como bipedestarse, caminar, alcance, presión, deglución, etc.) ya que es usada en patologías neurológicas donde hay daño de la neurona motora superior, pero está indemne la vía motora inferior. Esta electroestimulación es la técnica que se utilizará en este proyecto para la rehabilitación de la movilidad de la mano (Corredor Pastor (2017)).

3.6.1.2. Seguridad para el uso de Electroestimuladores. La norma ANSI/AAMI ES1-1993 de la asociación de avance en instrumentación médica (AAMI) la cual estipula los límites de corriente entre el paciente y el instrumento médico. Indicando que, dependiendo del voltaje y de la corriente que se le aplique al paciente, se puede generar varios fenómenos como lo son: estimulación eléctrica de tejidos excitables (nervio o músculo), calentamiento por efecto resistivo de tejidos y, quemadura electroquímica y daño de tejidos por corriente continua y voltajes muy altos; que para que esto suceda, el cuerpo se comportaría como un elemento más del circuito, sería como una resistencia variable dependiendo del peso del paciente. A continuación se presenta algunos límites que una persona de 70Kg, tomando en cada mano, un cable de cobre N°8, haciendo un lazo cerrado con una fuente de corriente de 60 Hz que se emplea al individuo en un lapso de 1 a 3 segundos.

- Umbral de percepción: El mínimo registrado fue a 0.5 mA a 60 Hz y en corriente continua de 2mA hasta 10mA.
- Corriente de paso: Esta es definida como la máxima corriente a la cual el sujeto puede soportar esta voluntariamente. El umbral mínimo esta corriente es de 6 mA.
- Fibrilación ventricular: El umbral para fibrilación ventricular se da entre 75 y 400 mA. En esta situación, el corazón puede alcanzar los 300 latidos por minuto (el rango normal se encuentra entre 60 y 90 latidos por minuto). (Zapata Vélez de Villa (2010))

3.7. Estimulación Eléctrica Funcional(FES)

3.7.1. Definición.

Es un método terapéutico en el cual se aplica un estímulo con corrientes eléctricas mediante electrodos que se colocan en la superficie de la piel para causar una contracción muscular; el objetivo es producir un movimiento funcional en las extremidades deterioradas que posteriormente es incorporada a una actividad diaria(Castellano (2017)).

3.7.2. Aplicaciones.

Como se menciona anteriormente, se basa en estimular un músculo del cuerpo que este deteriorado o que haya sufrido un accidente. Entre sus tantas aplicaciones las que se reconocen son:

- Terapia de rehabilitación para accidentes cerebrovasculares: Ayuda en la recuperación de la función motora en pacientes que han sufrido un accidente cerebrovascular.
- Tratamiento de incontinencia: La FES se usa para fortalecer los músculos del suelo pélvico y tratar la incontinencia urinaria y fecal.
- Mejora de la circulación sanguínea: Puede utilizarse para estimular la circulación sanguínea en personas con problemas circulatorios.
- Estimulación de nervios periféricos: Se utiliza para activar nervios periféricos y músculos en personas con daño nervioso periférico.
- Rehabilitación cardíaca: Puede aplicarse en programas de rehabilitación cardíaca para mejorar la fuerza y la función del músculo cardíaco.

3.7.3. Componentes de un FES.

Los componentes de un Estimulador Eléctrico Funcional (FES) son fundamentales para su funcionamiento y su capacidad para interactuar con el usuario. A continuación, se describen con más detalle estos componentes:

- Interfaz de Aplicación: Este componente actúa como el punto de interacción entre el usuario y el FES. Permite al usuario manipular y controlar el estimulador eléctrico funcional según sus necesidades y preferencias. La interfaz de aplicación puede variar según el diseño del dispositivo, e incluir botones, perillas, pantallas táctiles u otros dispositivos de entrada que faciliten la configuración y el control del FES (Ortiz and Ruiz (2014)).
- Generador de Pulsos: El generador de pulsos es una parte esencial del FES, ya que es responsable de generar las corrientes eléctricas pulsadas necesarias para estimular los músculos o nervios. Este componente permite

ajustar parámetros clave, como la frecuencia y la duración de los pulsos, para adaptarse a las necesidades terapéuticas específicas de cada usuario. La capacidad de variar estos parámetros es crucial, ya que diferentes aplicaciones de FES pueden requerir patrones de estimulación distintos.

- **Interfaz de Contacto (Electrodo):** La interfaz de contacto, comúnmente compuesta por electrodos, establece la conexión física entre el sistema electrónico del FES y el sistema biológico del usuario (músculo o nervio). Los electrodos permiten registrar señales eléctricas del cuerpo y aplicar estímulos eléctricos de manera precisa. Estos electrodos pueden colocarse en la piel o de manera invasiva, según la aplicación específica del FES. La elección adecuada de los electrodos y su ubicación es crucial para garantizar una estimulación efectiva y segura.
- **Sensores y Detectores:** En algunos sistemas de FES, se pueden incorporar sensores y detectores que monitorean la actividad muscular, la respuesta del usuario o la posición de las extremidades. Estos sensores permiten al FES ajustar la estimulación en tiempo real en función de la retroalimentación obtenida, lo que puede mejorar la precisión y la efectividad del tratamiento.
- **Batería o Fuente de Energía:** Para garantizar la portabilidad y la autonomía del FES, es esencial contar con una fuente de energía confiable. Esto puede ser una batería recargable o una fuente de alimentación externa, según las necesidades del usuario y la aplicación del dispositivo.
- **Unidad de Control:** La unidad de control es el cerebro del FES. Contiene la lógica de procesamiento y control que interpreta las señales del usuario, ajusta la estimulación según los parámetros establecidos y coordina la operación general del dispositivo.
- **Características de Seguridad:** Los FES suelen incorporar características de seguridad importantes, como límites de corriente máxima, detección de fallas en los electrodos y mecanismos de apagado de emergencia. Estas características garantizan que la estimulación eléctrica se realice de manera segura y controlada.
- **Transformador de pulso:** Este transformador es de tipo especial ya que trabaja en los circuitos conmutando la corriente continua. Para este caso, se busca minimizar la caída de voltaje en la bobina secundaria aislada,

puesto en el se genera un voltaje dependiendo de la conmutación de la bobina primaria.

3.7.4. Parámetros de un FES.

En la Tabla 2 se relacionan los tipos de forma de corriente en un FES.

- Frecuencia : Para que la contracción muscular sea tetánica la frecuencia de los impulsos que forman el tren debe ser mayor o igual a 20 Hz, si bien las frecuencias más utilizadas son entre 20 y 30 Hz, ya que frecuencias por encima de 50 Hz despolarizan fundamentalmente las fibras rápidas de tipo II, produciendo fatiga muscular
- Ancho de Pulso : Con respecto al ancho de pulso estegeneralmente se puede seleccionar entre 100 y 400 microsegundos en función de la respuesta neuromuscular, aunque hay autores y sistemas de estimulación que utilizan la misma anchura de pulso para todos los pacientes.
- Intensidad : Con respecto a la intensidad de la corriente, ésta debe provocar una contracción muscular suficiente pero no excesiva para llevar a cabo la función que se pretende (prensión, bipedestación, marcha, etc.). Un punto importante a considerar es que la intensidad puede variar entre sesión y sesión como cualquier método de electrostimulación , pero es recomendable usar la intensidad obtenida al momento de la evaluación electrodiagnostica. El rango empleado en FES va desde 0 a 100 mA de corriente máxima de pulsos.

Tabla 2

Tipos de forma de corriente en FES.

Tipo de corriente	Corriente	Características
Corrientes Galvánicas	Direct Current (DC)	<p>Flujo unidireccional continuo de partículas cargadas con una duración de al menos 1 segundo.</p> <p>Un electrodo es siempre el ánodo (+) y el otro es siempre el cátodo (-) durante todo el evento.</p> <p>Nota: Monofásico también se refiere a corriente continua, pero es interrumpida y no continua (es decir, pulsada), por lo que el efecto químico es mínimo.</p>
Corrientes Bifásicas	Altern Current(AC)	<p>Flujo de iones bidireccional e ininterrumpido; La dirección cambia al menos una vez por segundo.</p> <p>Frecuencia: la velocidad a la que la corriente cambia de dirección.</p> <p>Los electrodos alternan continuamente su polaridad en cada ciclo, por lo tanto, no se acumula carga debajo de los electrodos.</p> <p>Tipos de corriente alterna modulada utilizadas en tejido biológico:</p> <p>Modulada por ráfagas: conocida como corriente rusa"</p> <p>Modulada en amplitud: conocida como corriente interferencial</p>
Corriente Pulsante	DC y AC	<p>El flujo de partículas cargadas se detiene periódicamente durante menos de 1 segundo antes del siguiente evento.</p> <p>Los pulsos pueden ocurrir individualmente o en una serie.</p> <p>Los pulsos monofásicos no se alternan;</p> <p>La corriente monofásica pulsada (viaja en una dirección) permite que se acumule una carga en el tejido biológico.</p>

Nota. Adaptado (Flores Ramos (1998)).

3.8. Modelos comerciales de neuroprotesis FES para rehabilitación de miembro superior

3.8.1. *Bioness handmaster H200 wireless.*

El handmaster H200 wireless proporciona estimulación eléctrica a los nervios de los músculos que controlan la apertura y cierre de la mano y el movimiento del pulgar. La ortesis también estabiliza la muñeca.

Figura 9.

Bioness handmaster H200 wireless.



Nota. Adaptado (Bioness Inc. (2018))

Tabla 3

Especificación técnica Handmaster H200 wireless.

Especificación Técnica	Valor
Duración de pulso	100/200/300 μ s
Corriente pico máxima	80 mA
Frecuencia	20-45 Hz
Tipo de onda	Bifásica simétrica
Carga nominal	500 Ω

Nota. Adaptado (Bioness Inc. (2018)).

3.8.2. XFT-2003EA (Shenzhen XFT Medical Limited, China).

El XFT-2003EA detecta y analiza las señales EMG del paciente en tiempo real a través de los sensores de electrodos, y luego proporciona simultáneamente una estimulación eléctrica cómoda de baja frecuencia de acuerdo con la señal EMG que a su vez evocará la contracción muscular y permitirá a los pacientes participar activamente en las actividades de la vida diaria.

Figura 10.
XFT-2003EA.



Nota. Adaptado (Shenzhen XFT Medical Limited (2023)).

Tabla 4

Especificación técnica XFT-2003EA.

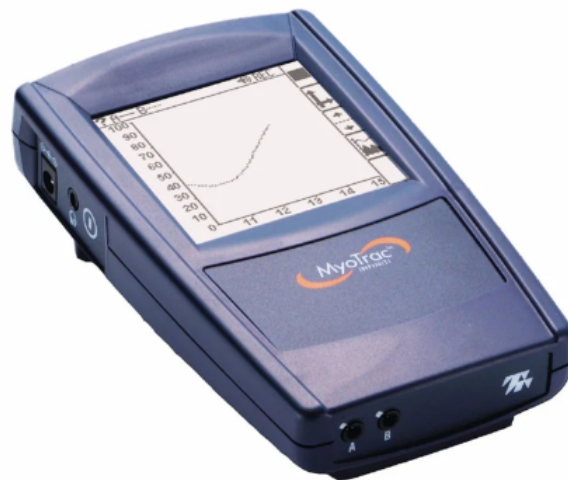
Especificación Técnica	Valor
Duración de pulso	50-450 μ s
Corriente pico máxima	60 mA
Frecuencia	2-100 Hz
Tipo de onda	Bifásica asimétrica
Carga nominal	500 Ω

Nota. Adaptado (Shenzhen XFT Medical Limited (2023)).

3.8.3. Saebo Myotrac Infiniti.

El Saebo MyoTrac Infiniti es una solución para todo el cuerpo que cabe en la palma de su mano. Este dispositivo es un sistema avanzado de estimulación eléctrica con biorretroalimentación que ofrece Más de 60 programas preestablecidos para elegir. Este revolucionario dispositivo todo en uno ofrece 4 modalidades de tratamiento que incluyen características basadas en evidencia que van desde EMG biorretroalimentación, estimulación eléctrica, estimulación activada por EMG y la innovación, EMG recíproco pendiente de patente estimula la estimulación para el entrenamiento de agonistas/antagonistas.

Figura 11.
Saebo Myotrac Infiniti.



Nota. Adaptado (Saebo MyoTrac Infiniti (2021))

Tabla 5

Especificación técnica Saebo Myotrac Infiniti.

Especificación Técnica	Valor
Duración de pulso	2-100 μ s
Corriente pico máxima	100 mA
Frecuencia	2-100 Hz
Tipo de onda	Bifásica simétrica
Carga nominal	500 Ω

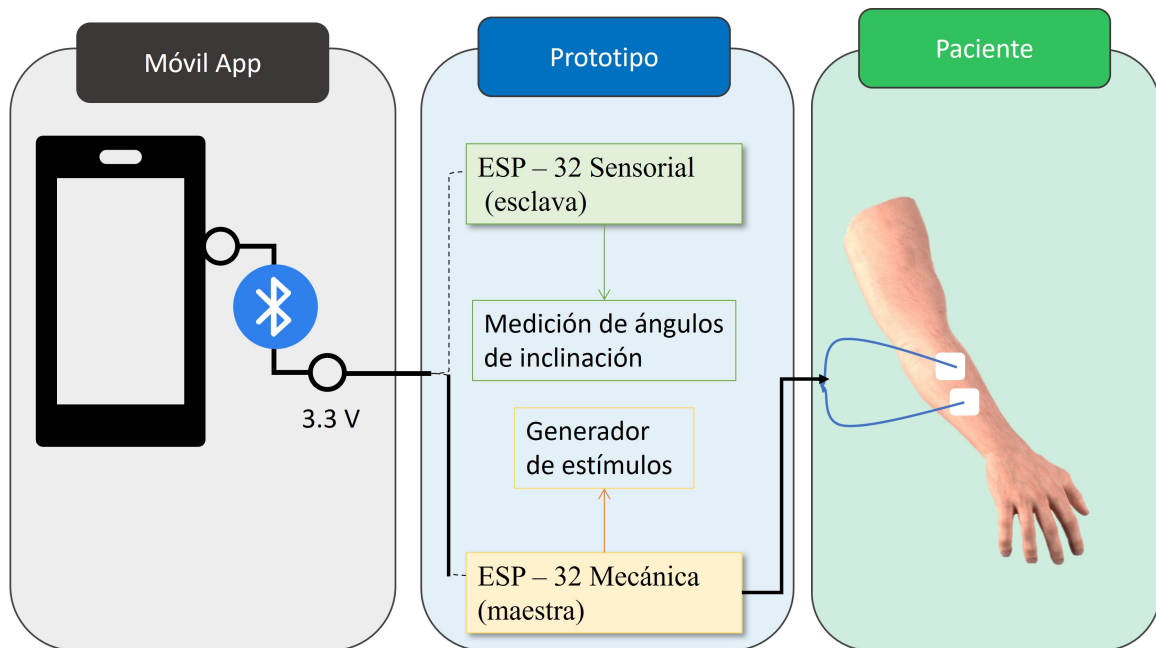
Nota. Adaptado (Saebo MyoTrac Infiniti (2021)).

4. Diseño del dispositivo FES

El diagrama de bloques del prototipo de FES se muestra en la Figura 10, el cual consiste en un microcontrolador (ESP32) y este se comunica con un dispositivo móvil por medio de Bluetooth y una aplicación que controla el modo de corriente de electroterapia deseada y sus variables.

Figura 12.

Diagrama de bloques del dispositivo.

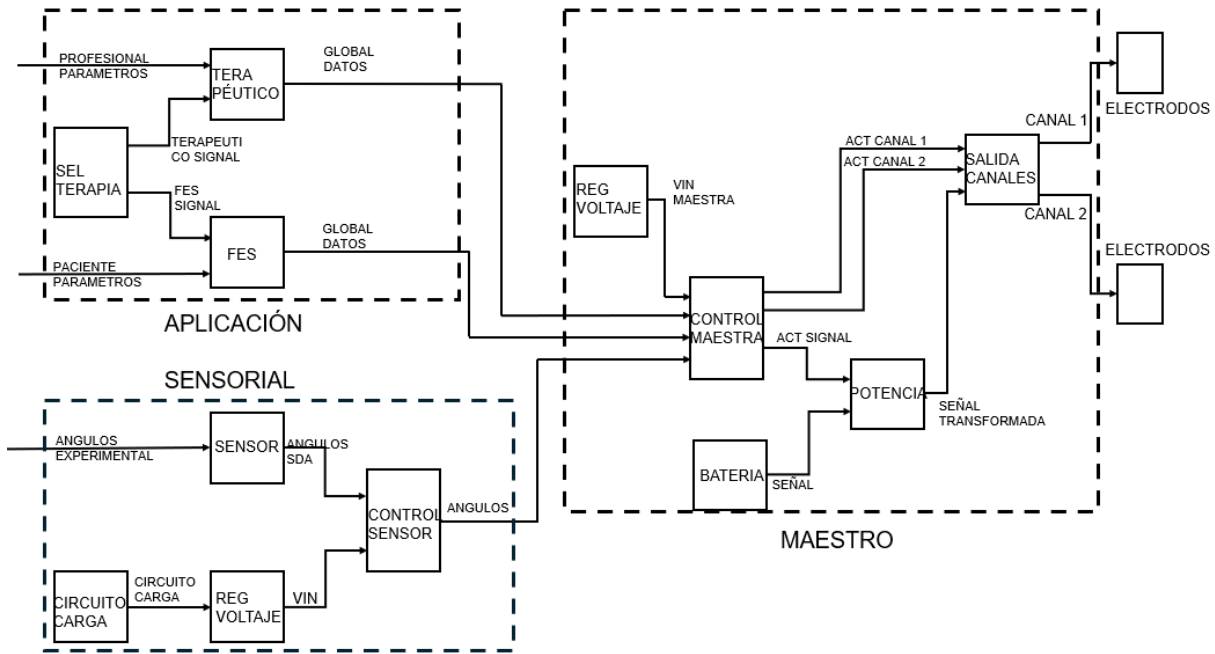


Nota. Diseño propio

A continuación, se podrá visualizar con más detalle el procesamiento de señal que se logrará a través de este dispositivo. Este es un diagrama de bloques con sus respectivas señales de entrada, salida y su función de bloque.

Figura 13.

Diagrama de bloques funcional del dispositivo.



Nota. Diseño propio

4.1. Condiciones de Diseño

Tabla 6

Condiciones de diseño.

Condición	Especificación
Forma de onda	Monofásica Cuadrada
Corriente de pulso máxima	100mA
Rango de frecuencia de los pulsos	20-40 hz
Rango de anchos de pulso	100-500 μ S

Nota. Diseño propio.

4.2. Unidad de control

Al comparar diferentes modelos de microcontroladores, aspectos clave como la velocidad de procesamiento, la cantidad de memoria, las capacidades de E/S y el consumo de energía deben considerarse meticulosamente. Estas características no solo afectan el rendimiento del dispositivo, sino también su eficiencia energética y su capacidad para adaptarse a diversas aplicaciones. Por lo tanto, al evaluar opciones, es crucial ponderar las necesidades específicas del proyecto junto con las características técnicas de cada microcontrolador para tomar la decisión más acertada. A continuación se presenta una tabla comparativa de los microcontroladores que se consideraron para el diseño del dispositivo.

Tabla 7

Comparación de especificación de microcontroladores.

FUNCIÓN	ESP-32	NodeMCU ESP8266	STM32	Arduino Nano
Microcontrolador	Microprocesador Xtensa Dual Core LX6 de 32 bits	Tensilica 32-bit Xtensa LX106	ARM@32 bits Cortex@-M3 CPU Core	ATMega328P
Memoria Flash	4MB	4 MB	64 Kb - 128 Kb	32 KB
SRAM	520MB	128 KB	20KB	2KB
EEPROM	No disponible	No disponible	2KB	1KB
Velocidad del reloj	Hasta 240 MHz	80 o 160 MHz	4 a 16MHz	16 MHz
Tensión de funcionamiento	3.3V DC	3.3V DC	1,7 V DC	5V DC
Tensión de entrada	3.3V DC	7-12 DC	2,0 a 3,6 VDC	6 a 20V
Consumo actual	80 mA - 90mA	15 μ A - 400 mA	45 mA - 80 mA	19 mA
Corriente continua por pin de E/S	40 mA	12 mA	25mA	40 mA
Pines digitales IO	36	17	14	22
Pines de entrada analógica	Hasta 18	1	Hasta 16	8
UARTs	3	2	3	No
SPI	4	2	2	4
I2C	2	1	2	2
CAN	Sí	No	Sí	No
PWM	16	4	6	6
Wi-Fi	Sí, 802.11 b/g/n	Sí	No	No
Bluetooth	Sí, especificación Bluetooth v4.2 BR/EDR y BLE	No	Sí	No

Nota. Diseño propio.

La ESP32 destaca en comparación con otros microcontroladores debido a su combinación única de características avanzadas y versatilidad. Equipada con un potente procesador de doble núcleo y conectividad inalámbrica integrada Wi-Fi y Bluetooth, la ESP32 ofrece un rendimiento excepcional y la capacidad de comunicarse con una va-

riedad de dispositivos y redes. Además, su amplia gama de periféricos e interfaces, como UART, SPI, I2C y GPIO, la convierten en una opción ideal para una variedad de aplicaciones, desde el desarrollo de proyectos IoT hasta sistemas embebidos complejos. La comunidad de desarrollo activa y el amplio soporte de software, incluidos frameworks como Arduino y MicroPython, hacen que la programación y el desarrollo con la ESP32 sean accesibles incluso para aquellos con experiencia limitada en microcontroladores. Con un costo relativamente bajo y una amplia disponibilidad, la ESP32 se destaca como una opción líder para proyectos que requieren un rendimiento sólido, conectividad avanzada y facilidad de desarrollo.

4.2.1. ESP-32.

Componente encargado de controlar la frecuencia de aplicación de los pulsos hacía el electrodo dependiendo la necesidad del paciente basados en la electroestimulación funcional. Por medio del protocolo Bluetooth, la aplicación móvil del dispositivo(celular, tablet) y el estimulador eléctrico que tiene como componente de comunicación la ESP-32, deben hacer conexión entre si y de este modo, desde la app el usuario elige la opción del tratamiento requerido y el móvil envía la información de la función a realizar al estimulador eléctrico. Esta comunicación se hace realidad por medio de una programación de un código que se realiza en la *ESP – 32* y al dispositivo móvil.

Ahora, en este proyecto se ha utilizado dos microcontroladores, para la comunicación entre ellos, se maneja *ESP – NOW*, siendo un protocolo de comunicación que permitirá intercambiar información entre las dos *ESP – 32*, proporcionando conexiones de una dirección o en ambas direcciones.

La comunicación dada la conexión en una dirección la iniciará el(los) dispositivo(s) llamado(s) Maestro(s), y la recibirá el(los) dispositivo(s) llamado(s) Esclavo(s), en esta configuración tenemos variedad de tipos de conexiones, tales como:

- Un maestro, un esclavo: Esta configuración es la más fácil de realizar, debido a que solo se debe tener en cuenta que, el maestro debe enviar la información hacía un solo esclavo y puede ser en un tiempo determinado.
- Un maestro, varios esclavos: Esta configuración es muy parecida a la de un maestro, un esclavo, pero, aquí se debe precisar muy bien la forma de envío de información, ya que el maestro manejará a vario esclavos, de esta

manera la comunicación debe ser clara para que se realice una buena estructuración de la función, debido a que el maestro envía al mismo tiempo la información para todos los esclavos.

- Varios maestros, Un esclavo: Esta configuración es un poco complicada de desarrollar, ya que el esclavo debe estar recibiendo información de varios maestros y este tiene como misión realizar cada trabajo que reciba de cada maestro al tiempo que se le haya indicado.

Además, existe la conexión en ambas direcciones, en esta configuración, no existe un maestro o esclavo, aquí las dos placas son remitentes y destinatarias, cada una recibe información de la otra y así podrán ejecutar acciones en pro del buen funcionamiento del dispositivo.

Cabe aclarar que, para este proyecto se utiliza la conexión un maestro, un esclavo. Agregando a lo anterior, se va a explicar el código que hace el correcto funcionamiento.

Tabla 8

Especificaciones de la ESP-32.

Tipo	Especificación
Voltaje Entrada	3.3V
CPU y Memoria	low-power Xtensa® 32-bit LX6 448 kB of ROM
Dimensión	25.65mm*18.15mm*3.25mm
Consumo	180mA
GPIO	32 Programable GPIOs
Bluetooth	Bluetooth 5, Bluetooth Mesh, speed:125Kbps, 500Kbps,1Mbps, 2Mbps
Wi-fi	802.11n(2.4GHz), Up to 150Mbps

Nota. Diseño propio.

4.2.2. Código ESP-32 Sensorial.

Continuando la explicación, para la parte esclava se agrega a la programación unas librerías; en la siguiente lista se mencionaran las que se utilizaron:

- *I2Cdev.h*: Esta librería contiene el protocolo de comunicación I2C, debido a que, el sensor esta programado por medio de esta comunicación, por ello, se agrega al código utilizado.
- *MPU6050.h*: Se agrega esta librería para poder adquirir los datos e inicializar los parámetros del sensor.
- *Wire.h*: Esta librería apoya en el protocolo de comunicación I2C.

Luego de escoger las librerías necesarias, se debe tener en cuenta, los protocolos que se utilizaran para la comunicación con la parte Maestra, para ello, se usa el protocolo *< Esp_now.h >* el cual sirve para comunicar las *ESP – 32* que se encuentran en ambas partes por medio de un protocolo de WiFi, y para complementar esta comunicación se debe agregar *Wifi.h* para poder hacer uso de toda la comunicación del protocolo WiFi.

Adicionalmente, se inicializa un objeto de la librería MPU6050 con el nombre que se le ha colocado al sensor y así, poder hacer uso del código.

Continuando, se generan dos variables inicializadas conectadas a 2 leds externos, en efecto, se les asignan nombres a estas variables como *led* y *led1* que estan programadas para indicar si la ESP esclava se conecta de forma correcta por medio del protocolo a la ESP Maestra para recibir información.

Se debe aclarar que, el sensor que se maneja, ofrece dos opciones, siendo una el comportamiento de un acelerómetro y la otra como un giroscopio, en este proyecto se enfoca la parte sensorial que ofrece como acelerómetro. Ahora bien, se crean variables enteras de 16bits que se usa para la lectura de los ejes del acelerómetro, para poder generarlas de esta forma, se debe anteponer *init16_t* a las variables en la línea del código.

Para finalizar la generación de variables, se inicializa una variable con la dirección MAC de la ESP-maestra que es la que recibe los datos, con el fin de cerrar la comunicación entre ellas.

Para continuar, se genera un objeto con la estructura de los datos que se van a enviar a la parte maestra, para ello, se genera dos variables de tipo flotante llamadas en el código *angulo – x* y *angulo – y*, para este caso, solo necesitaremos estos dos ángulos referentes. Luego de generar la estructura, se crea el objeto llamado *angulos*.

Para no tener fallos en la comunicación de las ESP se programa una función llamada *Callback*, esta sirve para el envío de un mensaje y así poder detectar si esta correctamente la comunicación entre ellas.

Luego de generar todas las variables que se van a usar, se empieza con la parte del código llamado VOID SETUP.

Lo primero que se debe realizar es declarar el modo de las variables si son entrada o salida del sistema, es por ello que, la primera línea de esta función es la declaración de las variables de salida que son los leds, siendo ellos, un aviso físico del correcto funcionamiento.

En efecto, es recomendable ejecutar el código con una comunicación serial, para poder realizar depuraciones desde el computador; además de programar lo anterior, se debe inicializar la librería *WIRE* para la comunicación *I2C* y el sensor para el uso de él mismo. Se realiza un test para la conexión del sensor y se agrega la línea de impresión de respuesta de esta conexión, para tener todo controlado y saber que esta correctamente funcionando.

Se toma una ESP como una estación de WIFI, de igual manera se inicializa el protocolo ESP-NOW y se agrega líneas de prueba para poder saber si se activa correctamente.

Agregamos la programación de recepción de los *callback* para comprobar si la información es recibida correctamente. Después se registra la *ESP-32* maestra y se agrega en la comunicación con *ESP-NOW*; esta comunicación se verifica adicionando unas líneas de programación para los led, siendo de la forma Led rojo- conexión nula y Led verde- conexión establecida.

Ahora, ingresamos a la parte del código llamado *VOID LOOP*, y se empieza con tomar los valores de la aceleración del sensor con la función *getAcceleration* para los parámetros *X, Y, Z*, luego hallamos los ángulos de inclinación con las operaciones de tangente inversa (*atan*), raíz cuadrada (*sqrt*), teniendo en cuenta los valores de aceleración recibidos del sensor. Adicionalmente, se imprime los valores de inclinación para poder hacer una corrección debido que al reiniciar la *ESP-32* se pierde la calibración del sensor, ya que es demorada al realizarla.

Cuando se hace la modificación de las variables y se establecen con el protocolo de comunicación para hacer el envío por medio de *ESP-NOW*. Finalmente, se realiza una evaluación para poder visualizar que hasta este punto esta todo correctamente funcionando.

Al pensar en poder sintetizar el código para que sea un poco más rápida la respuesta, se generan algunas funciones tales como la de los led.

4.2.3. Código ESP-32 Maestra.

Iniciando con la parte Maestra, se establecen las librerías que se utilizarán para el funcionamiento del código, se explicará cada una de ellas a continuación:

- `Bluetoothserial.h`: Esta librería es para establecer una comunicación en el medio bluetooth.
- `ESP-NOW.h`: Es el protocolo que se manejará para la comunicación entre las ESP.
- `Wifi.h`: Esta librería apoyará la comunicación que se genera entre las ESP.

Se realiza una validación del encendido del Bluetooth, y en caso de no estar activado, se enciende automáticamente. Luego, se configura las variables recibidas desde el sensor y debe ser escritas de la misma forma en la parte de la ESP-32 esclava.

En esta parte del código se debe declarar las estructuras de las variables que se van a recibir, teniendo en cuenta que estas estructuras deben coincidir con las variables de los datos enviado por el sensor.

También se inicializan las variables que se van a utilizar durante el código; la variable `incoming[4]` es una vector lista donde se guardará los datos recibidos por el protocolo ESP-NOW; las siguientes variables son las controladoras de los relés que se comportan como puertas de paso de la señal, para ella, se necesita inicializar una variable para poder ser recibida. Para la sincronización de los pulsos de la señal, se inicializa 4 variables que deben ser congruentes con los valores iniciales de la aplicación móvil, los valores son guardados en la variable `Incoming[4]`.

Se programa la función `Callback`, Se genera una variable tipo objeto para el bluetooth para posteriormente usarlo. Al ingresar a la parte del `setup` de este código primero se inicializa el monitor serial para hacer una depuración, En segunda instancia la ESP se programa como `Wifi Station` para la comunicación; Luego, para la verificación de la comunicación se genera un `if` que esta programado para poder corroborar si el protocolo ESP-NOW se inició correctamente y dar una respuesta en caso de que este en fallo, a su vez, se establece un nombre bluetooth para la ESP esclavo, finalmente, se definen las variables de salidas que manejan los relés y la señal que se usará.

En este momento, se plantea en el `Void Loop` un `if` que se encargará de recibir datos por el puerto serial y en

caso de ser así, le asigna esos valores a las 4 variables que se inicializaron en el principio del código, luego de asignar las variables, se debe hacer la conversión de los valores a una misma unidad; milisegundos, para poder trabajar a la misma escala, también se establece la duración total del ciclo, con su periodo de excursión, definir la cantidad de ciclos de los modos de aplicación de masaje, como los tiempos de encendido y apagado, esto se realiza con el fin de no lidiar con problemas a futuro que podría presentar el PWM; es una solución alternativa, sencilla y funcional.

El siguiente paso de programación, se efectúa teniendo en cuenta, los datos que recibe por medio de la comunicación serial emitida por bluetooth; así que, la explicación a la funcionalidad del proceso es, si esta variable llamada ESTADO recibe en valor un 1, ingresa a un if que es la función masaje, ella es la encargada de hacer una leve activación durante un tiempo establecido; además de ello, también se realiza el ajuste de que solo puede estar activado un solo canal a la vez, esto se logra con el control de activación de los relés, al realizar completo este ciclo for, que esta internamente en el ciclo if, la variable ESTADO toma el valor 3, en este estado, en el control de los relés, pasan a estar en modo abiertos, lo cual hace que ninguno de los dos, este dejando pasar alguna señal, se toma en cuenta este ciclo, para tener con seguridad la desactivación de los relés.

A partir de la explicación anterior, se llega al punto de la funcionalidad mas importante del proyecto, ya que en esta parte del código es donde se define, los modos que se ejecutan para la movilidad de la mano, dependiendo del valor que se recibe del sensor por medio de tipo de comunicación ESP-NOW, recordando, que este es el modo de comunicación entre las dos ESP32. Esta parte del código, es la que va activar el canal 1, el canal 2 o desactivarlos, ya que se tuvo en cuenta una franja muerta, para que el cambio de modo no sea tan brusco; en pocas palabras, lo que va a realizar esta parte, es ejecutar solo la señal por medio de un solo canal, porque, claramente, se esta evitando que los dos canales se activen al tiempo. Luego de realizar todo el proceso, vuelve a tomar el valor 3 la variable ESTADO para más seguridad del funcionamiento correcto del mismo.

Para cerrar este ciclo, se presenta el ESTADO 3, un STOP para poder mantener los relés totalmente abiertos y así, evitar alguna filtración de cualquier tipo de señal emitida, este estado se programa por seguridad y se utiliza en todo momento, para poder tener mas control de los relés.

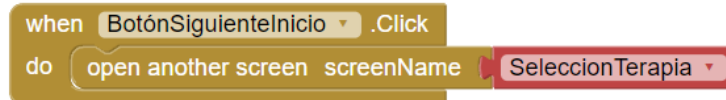
4.3. Interfaz de comunicación entre el dispositivo y el usuario

En esta sección se hablará de la aplicación que se realizó para poder ser manejado remotamente el dispositivo. En ella se podrá escoger el tipo de función que desee el usuario, como lo es la Fes terapéutico o la estimulación eléctrica funcional(FES) actividades de vida diaria, también puede interactuar con el dispositivo, seleccionando el tipo de frecuencia que desee, el tiempo de terapia, y muchas otras variables que desea modificar.

En esta aplicación se logra observar el nombre Estimulador eléctrico Funcional, el logo institucional y un botón con el nombre de Siguiente, que es para la continuación de la siguiente vista; para esta portada, Se programó solo el botón llamada en el código *BotnSiguienteInicio*, puesto que el nombre es una entrada tipo texto, y pues el logo de la universidad es tipo imagen para hacer más llamativa la aplicación y que sea visible un poco de los creadores del dispositivo; para la programación de la variable *VentanaSiguiente* solo se tuvo en cuenta un tipo de evento CLICK, llamada así a la función que se programó, en ella, se logra obtener que, cuando le den click al botón *Siguiente* de inmediato se abra la siguiente ventana, llamada *SeleccionTerapia*. En la siguiente Fig. 14 se muestra la programación de botón.

Figura 14.

Estructura del código de la portada.



Nota. Diseño propio

Figura 15.
Diseño de la portada.

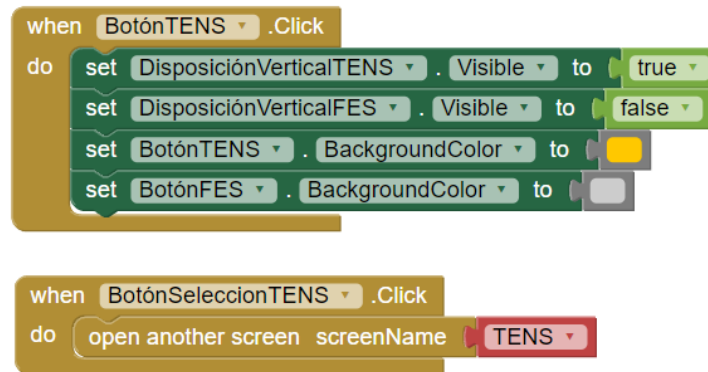


Nota. Diseño propio

Continuando con la explicación de la programación de la APP, incluye el diseño *SeleccionTerapia*, en él se visualizan dos botones llamados *FESmodoteraputico* y *FESactividadesdiarias*. Para el botón *Modoteraputico* cuando se le oprime, da las siguientes funciones:

- Activa todo lo referente al screen *TENS*.
- Desactiva lo referente al screen *FES*.
- Coloca color al botón *TENS* en tono amarillo.
- Deja el mismo color para el botón *FES*.
- Abre la ventana de *TENS*.

Figura 16.
Estructura del código para el Botón TENS(modo terapéutico).



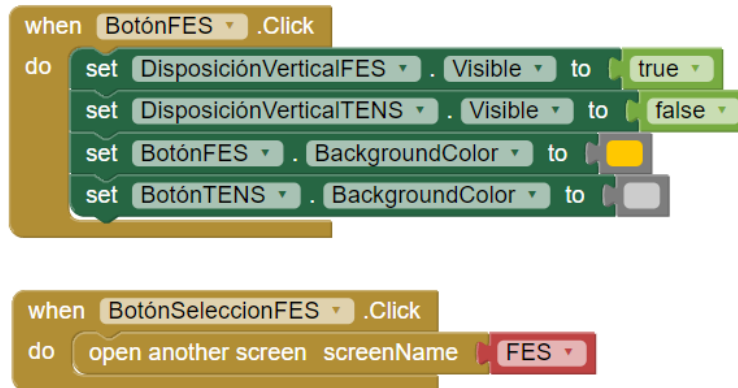
Nota. Diseño propio

Como lo mencionado anteriormente, el botón *FES* tiene la misma estructura del botón *TENS*, solo que con diferentes indicaciones, así que:

- Activa todo lo relacionado con el screen *FES*.
- Desactiva screen *FES(modoteraputico)*.
- Coloca color al botón *FES* en tono amarillo.
- Deja el color gris al botón *TENS*.
- Abre la ventana de *FES*.

Figura 17.

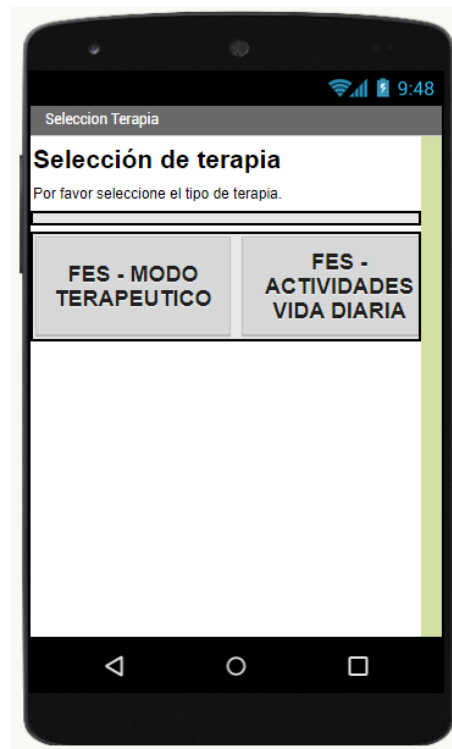
Estructura del código para el Botón FES.



Nota. Diseño propio

Adicionalmente, se diseña la proyección de pantalla hacia la aplicación móvil. En la Fig.18 se mostrará el diseño final del screen.

Figura 18.
Diseño de la ventana de Selección de Terapia.



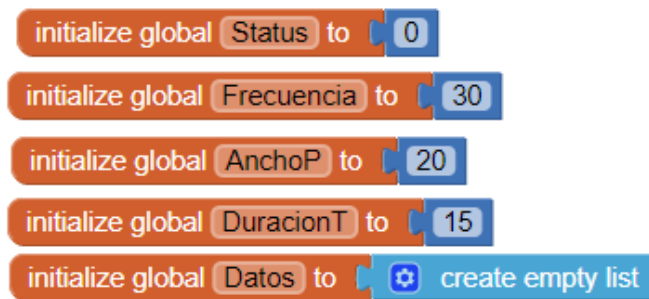
Nota. Diseño propio

Para iniciar con la explicación, se deben modificar variables como lo son:

- Status
- Frecuencia = 30
- Ancho de Pulso $AnchoP = 20$
- Duración de la terapia $DuracionT = 15$
- Datos, generar una lista en blanco.

Figura 19.

Programación de las variables Iniciales.



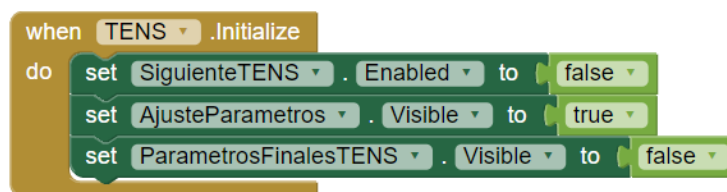
Nota. Diseño propio

Cuando se inicializa el Screen FES(modos terapéutico) se realiza las siguientes funciones:

- el Botón *SiguienteTens* se deshabilita hasta que se seleccione los parámetros para la terapia
- Ajustes Parámetros se hacen visibles
- los parámetros seleccionados finales para la terapia se ocultan, hasta que se oprima el botón *SiguienteTens*.

Figura 20.

Programación de inicialización del Screen FES(modos terapéutico).



Nota. Diseño propio

Adicionalmente, se programa el botón *ConectarBluetooth*, ya que se debe contar con un respaldo de conexión hacia la ESP32. Entonces, al tener claro esto, se procede a explicar la programación luego de dar click, esta cuenta con un ciclo *if* ya que, puede tomar dos caminos, el primero que se conecte y el segundo que no logre hacer conexión. De

esta manera, cada camino desencadena una serie de funciones, el ciclo *IF* tiene como parámetro una comparación, con ella, se define que camino debe tomar la programación, a continuación el primer caso que sería la conexión Bluetooth:

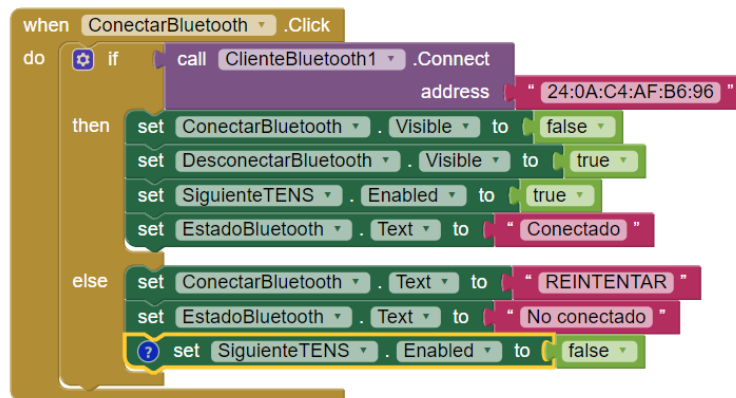
- El botón *ConectarBluetooth* se oculta
- El botón *DesconectarBluetooth* se hace visible
- El botón *SiguienteTENS* se activa
- El tipo texto llamado *EstadoBluetooth* nos muestra el mensaje "Conectado".

El segundo caso, se activaría, si no se hace la conexión Bluetooth, así que se desencadenaría las siguientes funciones:

- El tipo texto *ConectarBluetooth* nos muestra el mensaje "Reintentar"
- El tipo texto llamado *EstadoBluetooth* nos muestra el mensaje "NO Conectado"
- El botón *SiguienteTENS* se desactiva.

Figura 21.

Programación del botón Conexión Bluetooth.



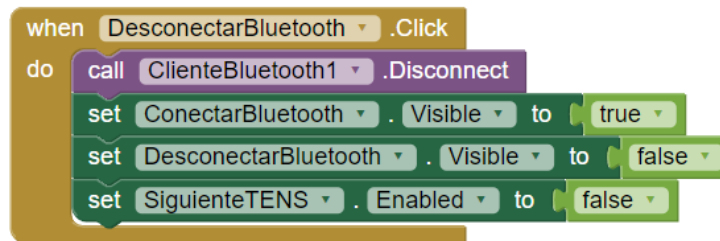
Nota. Diseño propio

Así mismo, como se debe tener un botón de conexión Bluetooth, también se debe contar con un botón de desconexión, por precaución, a continuación las funciones al darle click al botón *DesconectarBluetooth*:

- Se utiliza la función call para desconectar el bluetooth con la ESP-32
- El botón *ConectarBluetooth* se hace visible
- El botón *DesconectarBluetooth* se oculta
- El botón *SiguienteTENS* se desactiva.

Figura 22.

Programación del botón Desconexión Bluetooth.



Nota. Diseño propio

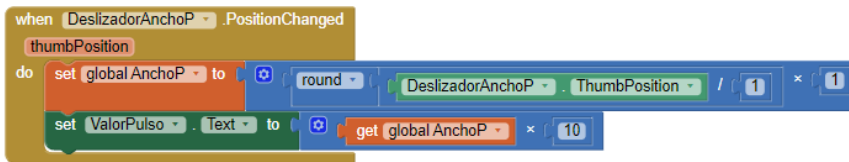
Para continuar con la explicación de la programación, se tienen los ajustes de los parámetros que se utilizarán para la terapia, estos parámetros son los mismos de las variables iniciales: Frecuencia, Ancho del pulso, duración de la terapia. Para poder darle un rango de movimiento a estas variables, ya que para el proyecto deben ser ajustables, se ha optado por programar deslizadores. Así que, será la misma estructura para cada uno, exceptuando un paso para el ancho del pulso; entonces, la estructura es: cuando se hace el deslizamiento para el ajuste de cada variable, debemos definir y redondear el valor, y así poder mostrarlo en el screen en una variable texto y poder ser utilizada en la ESP-32 para la terapia; la única diferencia que tiene el ancho de pulso es que se multiplica la salida por un valor de 10. 23 , 24 y 25

Figura 23.
Deslizador de Frecuencia.



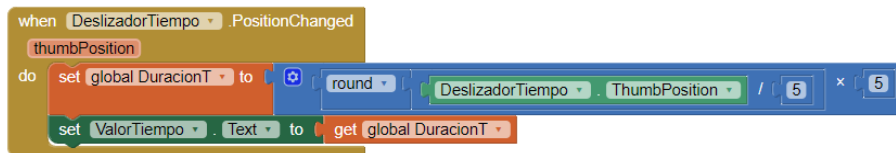
Nota. Diseño propio

Figura 24.
Deslizador de Ancho de pulso.



Nota. Diseño propio

Figura 25.
Deslizador de tiempo.



Nota. Diseño propio

Ahora se le programará el botón *SiguienteTENS* para poder visualizar el avance de la terapia, ya que al seleccionar los parámetros ya solo es enviar los valores seleccionados para poder ejecutar el código de la ESP-32 y el código de la visualización del avance de la terapia. así que las funciones al oprimir este botón son:

- El Deslizador de Frecuencia se deshabilita.
- El Deslizador de Ancho de pulso se deshabilita.

- El Deslizador de tiempo se deshabilita.
- El Ajuste de parámetros se oculta.
- El Reloj de la terapia *RelojTENS* se deshabilita.
- El texto de la frecuencia muestra el mensaje con el valor seleccionado.
- El texto del ancho del pulso muestra el mensaje con el valor seleccionado multiplicado por un valor de 10.
- El texto de duración muestra el mensaje con el valor seleccionado.
- El texto de *minutosRestantes* muestra el mensaje con el valor seleccionado de duración.
- Se muestra los parametros finales TENS.
- Se activa el botón *InicioTENS*.
- Se activa el botón *BotomProbar* se hace visible.
- Se activa el botón *AbrirMano* se habilita para las pruebas.
- Se activa el botón *CerrarMano* se habilita para las pruebas.
- Se activa el botón *BotomConfirmar* se habilita para las pruebas.

Figura 26.

Programación del botón Siguiente en el Screen FES(modos terapéuticos).

```
when SiguienteTENS .Click
do
  set DeslizadorFrecuencia .ThumbEnabled to false
  set DeslizadorAnchoP .ThumbEnabled to false
  set AjusteParametros .Visible to false
  set RelojTENS .TimerEnabled to false
  set FrecuenciaFinal .Text to get global Frecuencia
  set AnchoPFinal .Text to get global AnchoP × 10
  set DuracionFinal .Text to get global DuracionT
  set MinutosRestantes .Text to get global DuracionT
  set ParametrosFinalesTENS .Visible to true
  set BotónInicioTENS .Enabled to true
  set BotonProbar .Visible to true
  set AbrirMano .Enabled to true
  set CerrarMano .Enabled to true
  set BotonConfirmar .Enabled to true
```

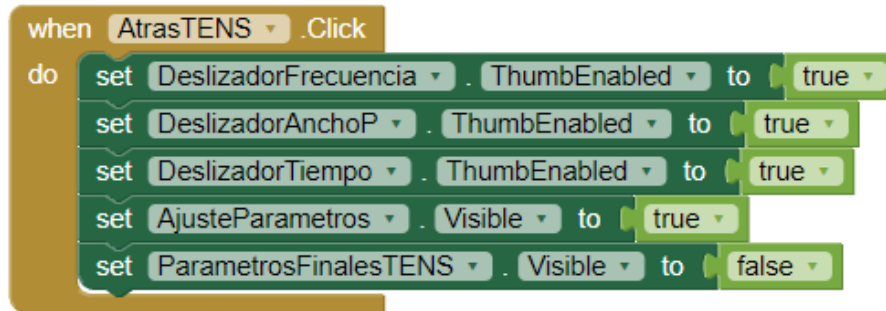
Nota. Diseño propio

Luego del botón *SiguienteTENS* se procede a realizar la programación del botón *atrasTENS* el cual realizará las siguientes funciones:

- El Deslizador de Frecuencia se habilita.
- El Deslizador de Ancho de pulso se habilita.
- El Deslizador de tiempo se habilita.
- El Ajuste de parámetros se hace visible.
- Los parámetros finales de la terapia se oculta.

Figura 27.

Programación del botón atras en el Screen FES(modos terapéuticos).



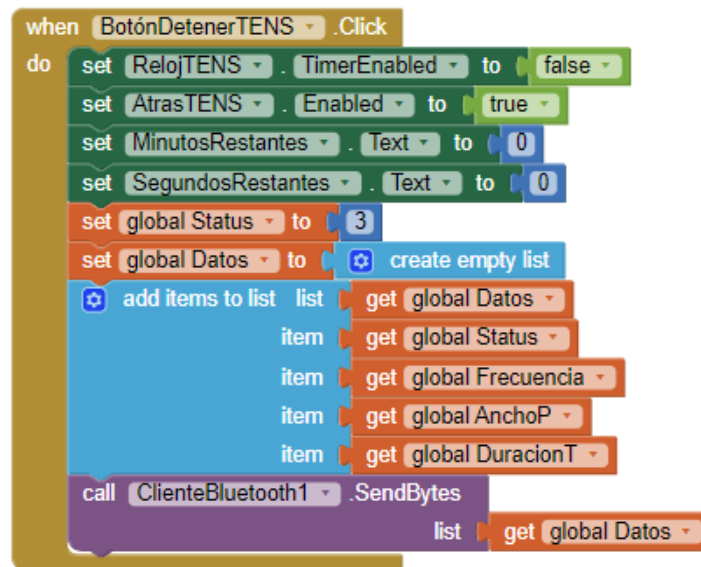
Nota. Diseño propio

Por otro lado, se genera un botón llamado *BotonDetenerTens*, la programación de este hace que se detenga de inmediato todo el proceso de la terapia, para que el paciente no tenga dificultad alguna de poder solucionar la situación, entonces, para que el proceso se finalice, deben ocurrir las siguientes acciones al presionar este botón:

- El Reloj de la terapia *RelojTENS* se detiene
- El botón de *AtrasTENS* se habilita
- El texto *MinutosRestantes* muestra el valor de 0
- El texto *SegundosRestantes* muestra el valor de 0
- La variable global *Status* toma el valor de 3
- Se ingresa a la lista en blanco de global *Datos* las siguientes variables: *Status*, *Frecuencia*, *AnchoP*, *DuracionT*.
- Se envía la lista *Datos* por medio de Bluetooth hacia la ESP-32.

Figura 28.

Programación del botón Detener en el Screen FES(modos terapéuticos).

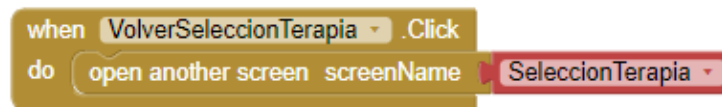


Nota. Diseño propio

También se cuenta con un botón para volver al anterior Screen *SeleccionTerapia*, este botón se le ha llamado *VolverSeleccionTerapia*, al darle click de inmediato se le da la función de mostrar el Screen en el cual se le da la selección de cual terapia quiere comenzar, así como se muestra en la fig. 29

Figura 29.

Programación del botón Volver Selección de Terapia en el Screen FES(modos terapéuticos).



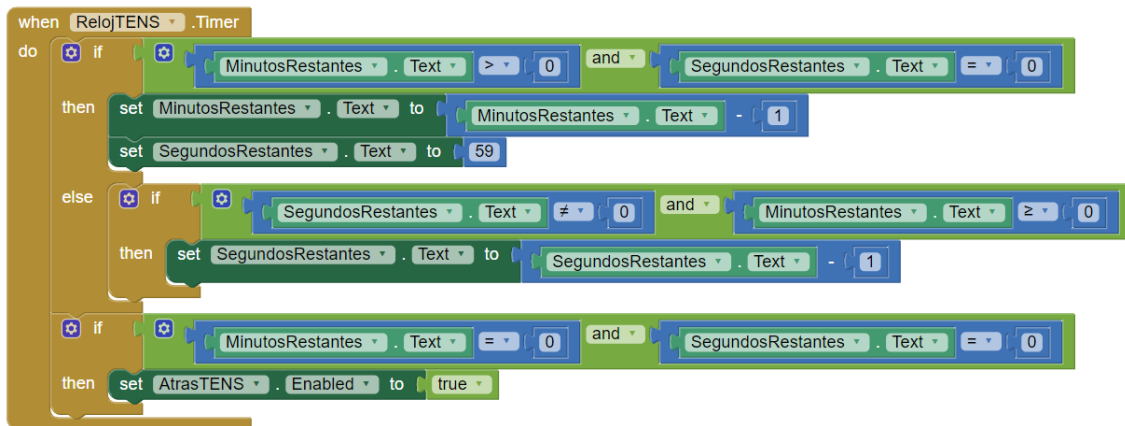
Nota. Diseño propio

Para poder llevar el tiempo de la terapia exacto, se programa un reloj digital, la programación de este reloj contiene dos ciclos *if*, uno es para el conteo regresivo del tiempo, que a su vez, tiene un ciclo *if* anidado y el otro ciclo es para la detención del reloj. Como ya se ha mencionado, el ciclo *if* siempre debe llevar, al menos un condi-

cional, en este caso, son los *MinutosRestantes* y *SegundosRestantes* si se cumple el primer caso, se hace la resta de *MinutosRestantes* -1; si se cumple la segunda condición; se hace la resta de *SegundosRestantes* -1; pero si se cumple la tercera opción, se activa el botón *AtrasTens*; En la fig. 30 se muestra la estructura del reloj.

Figura 30.

Programación del reloj para el modo terapéutico.

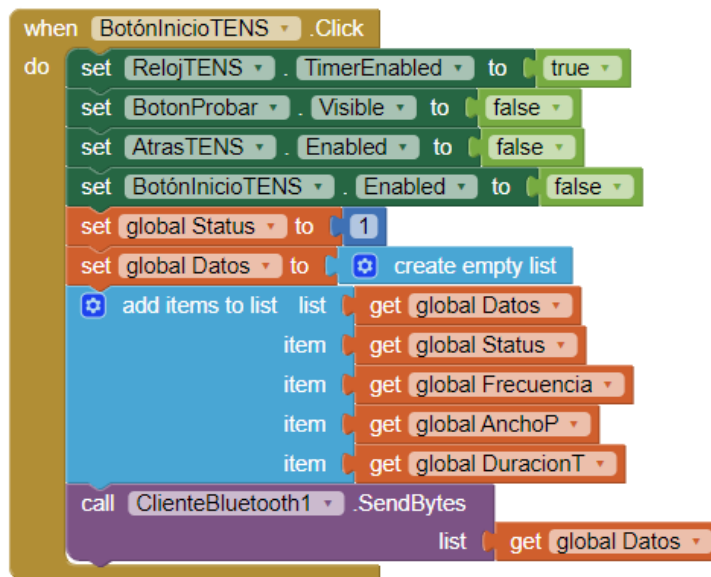


Nota. Diseño propio

Se hablará de la programación del botón inicio de la terapia *BotnInicioTENS* que tiene como funciones tales como:

- El Reloj de la terapia *RelejTENS* empieza a estar activo.
- El botón de *BotonProbar* se deshabilita o se oculta.
- El botón *AtrasTENS* se deshabilita o se oculta.
- El botón *BotonInicioTENS* se deshabilita o se oculta.
- La variable global *Status* toma el valor de 1.
- Se ingresa a la lista en blanco de global *Datos* las siguientes variables: *Status*, *Frecuencia*, *AnchoP*, *DuracionT*.
- Se envía la lista *Datos* por medio de Bluetooth hacia la ESP-32.

Figura 31.
Programación del botón de inicio para el modo terapéutico.



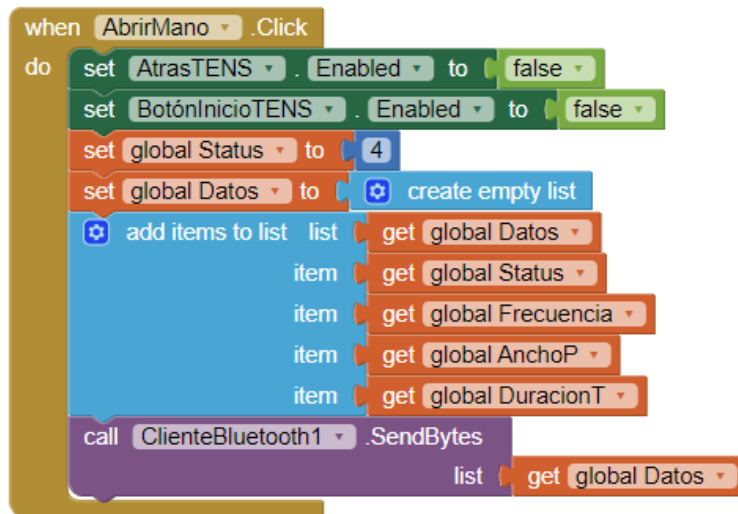
Nota. Diseño propio

Continuamos con la programación del botón *AbrirMano* que hará las siguientes funciones:

- El botón de *AtrasTENS* se deshabilita o se oculta.
- El botón *BotonInicioTENS* se deshabilita o se oculta.
- La variable global *Status* toma el valor de 4.
- Se ingresa a la lista en blanco de global *Datos* las siguientes variables: *Status*, *Frecuencia*, *AnchoP*, *DuracionT*.
- Se envía la lista *Datos* por medio de Bluetooth hacia la ESP-32.

Figura 32.

Programación del botón de Abrir Mano para el modo terapéutico.



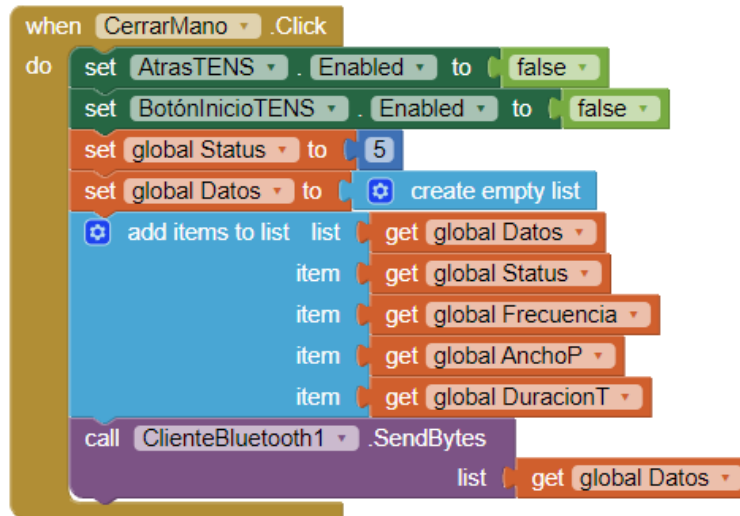
Nota. Diseño propio

La siguiente imagen se visualizará las funciones que están precedidas por el botón *CerrarMano* y se explicarán a groso modo en la siguiente lista:

- El botón de *AtrasTENS* se deshabilita o se oculta.
- El botón *BotonInicioTENS* se deshabilita o se oculta.
- La variable global *Status* toma el valor de 5.
- Se ingresa a la lista en blanco de global *Datos* las siguientes variables: *Status*, *Frecuencia*, *AnchoP*, *DuracionT*.
- Se envía la lista *Datos* por medio de bluetooth hacia la ESP-32.

Figura 33.

Programación del botón de Cerrar Mano para el modo terapéutico.



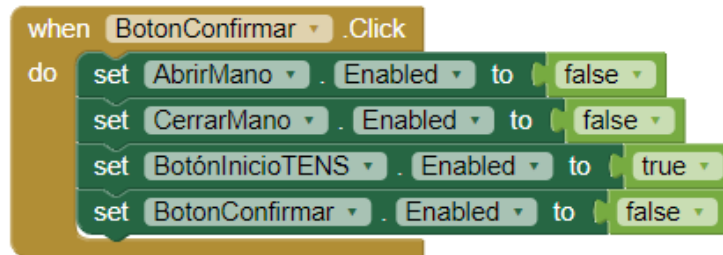
Nota. Diseño propio

El último botón que se explicará es el botón *BotonConfirmar* que de la misma forma que los anteriores tiene funciones a su mando, las siguientes funciones mencionadas son precedidas por la activación de este botón:

- El botón de *AbrirMano* se deshabilita o se oculta.
- El botón *BotonInicioTENS* se habilita o se hace visible.
- El botón *CerrarMano* se deshabilita o se oculta.
- El botón *Botonconfirmar* se deshabilita o se oculta.

Figura 34.

Programación del botón de Confirmar para el modo terapéutico.

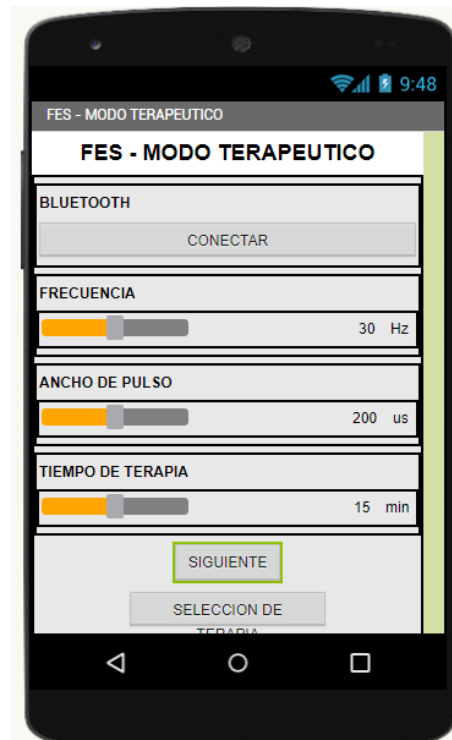


Nota. Diseño propio

Finalmente, se hace el diseño del Screen del modo terapéutico para la visualización en la aplicación móvil. en la siguiente Fig. 35

Figura 35.

Diseño del Screen FES(modos terapéutico).



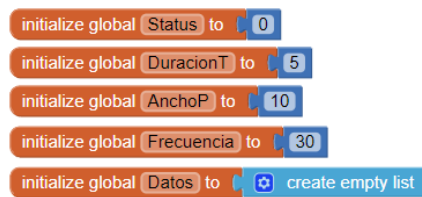
Nota. Diseño propio

A partir de este momento, se empezará a explicar la programación de la segunda terapia que se ofrece en este proyecto, esta terapia es *FES* electroestimulación Funcional. Para empezar, se declaran las variables globales con un valor específico; a continuación los valores asignados para inicializarlas.

- Status = 0
- Frecuencia = 30
- Ancho de Pulso *AnchoP* = 10
- Duración de la terapia *DuracionT* = 5
- Datos, generar una lista en blanco.

Figura 36.

Variables globales iniciales del Screen FES(actividades diarias).

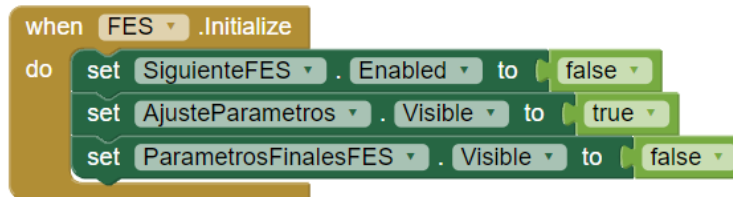


Nota. Diseño propio

Cuando en el Screen de selección de terapias se selecciona la estimulación eléctrica funcional *FES(actividadesdiarias)* se activa el screen de la terapia *FES(actividadesdiaria)* entonces también se despliega una acciones en la programación.

Figura 37.

Programación de inicio del Screen FES(actividades diarias).o



Nota. Diseño propio

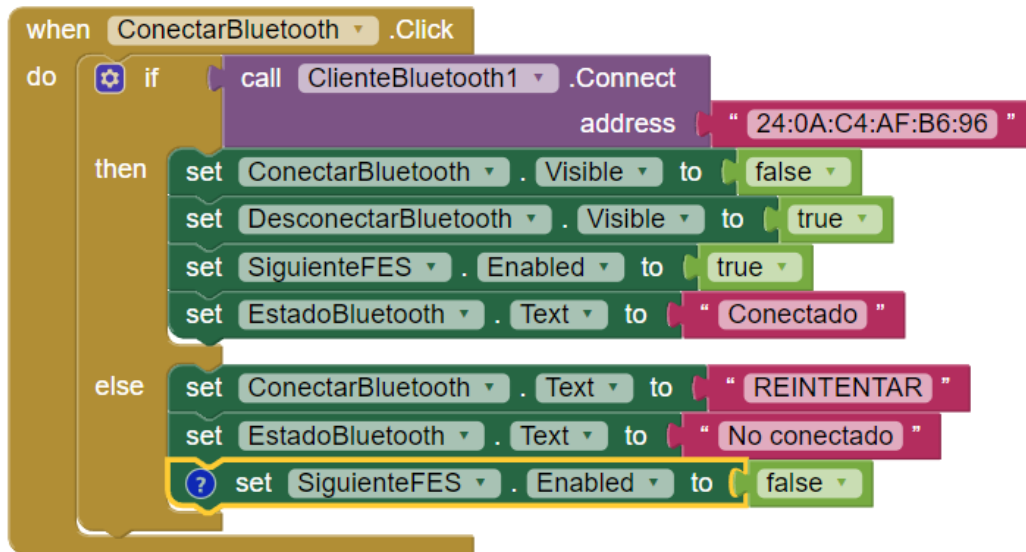
Para la comunicación entre la aplicación y el circuito de control *ESP – 32* se hace por el medio Bluetooth.

Ahora bien, la programación de esta parte, tiene un ciclo *if*, así que lo anterior era el primer caso, ahora se expresa el caso de que no se haga la conexión Bluetooth, se despliega las siguientes acciones:

- El texto *ConectarBluetooth* muestra el mensaje "REINTENTAR"
- El texto *EstadoBluetooth* muestra el mensaje "No Conectado"
- el botón *SiguienteFES* se deshabilita.

Figura 38.

Programación de botón de conectar por medio Bluetooth.

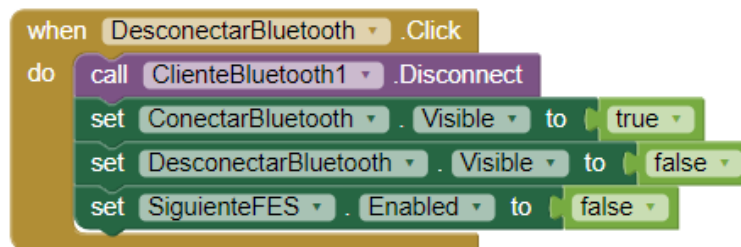


Nota. Diseño propio

También se manejará el botón *DesconectarBluetooth* que hará la función inversa del botón *Conectar* con menos funciones, ya que al desconectar el dispositivo, no es necesario una respuesta de texto *REINTENTAR*. En la siguiente imagen se muestra la programación de él.

Figura 39.

Programación de botón de desconectar por medio Bluetooth.



Nota. Diseño propio

Para los ajustes de los parámetros como lo es frecuencia, ancho de pulso, y duración de tiempo, se programa

unos deslizadores para cada variable. Estos deslizadores tienen la misma estructura, cada variable tiene un valor que se redondea a un valor exacto, y se muestra en un tipo texto el valor que se le asigna, a continuación se muestra la programación en las fig. 40 , 41 y 42

Figura 40.
Programación de deslizador de frecuencia.

```

when DeslizadorFrecuencia .PositionChanged
thumbPosition
do
set global Frecuencia to round(DeslizadorFrecuencia.ThumbPosition / 1 * 1)
set ValorFrecuencia.Text to get global Frecuencia

```

Nota. Diseño propio

Figura 41.
Programación de deslizador de Ancho de pulso.

```

when DeslizadorAnchoP .PositionChanged
thumbPosition
do
set global AnchoP to round(DeslizadorAnchoP.ThumbPosition / 1 * 1)
set ValorPulso.Text to get global AnchoP * 10

```

Nota. Diseño propio

Figura 42.
Programación de deslizador de Ancho de pulso.

```

when DeslizadorTiempo .PositionChanged
thumbPosition
do
set global DuracionT to round(DeslizadorTiempo.ThumbPosition / 1 * 1)
set ValorTiempo.Text to get global DuracionT

```

Nota. Diseño propio

Continuando con la explicación de la programación, lo siguiente es el botón *SiguienteFES*, este botón brinda la opción de poder visualizar el avance de la terapia *FES*, porque al seleccionar los parámetros luego se envía por

medio de conexión bluetooth para poder ejecutar el código de la ESP-32 y el código de la visualización del avance de la terapia. La siguiente lista tiene las funciones descritas:

- El Deslizador de Frecuencia se deshabilita.
- El Deslizador de Ancho de pulso se deshabilita.
- El Deslizador de tiempo se deshabilita.
- El Ajuste de parámetros se oculta.
- El Reloj de la terapia *RelojFES* se deshabilita.
- El texto de la frecuencia muestra el mensaje con el valor seleccionado.
- El texto del ancho del pulso muestra el mensaje con el valor seleccionado multiplicado por un valor de 10.
- El texto de duración muestra el mensaje con el valor seleccionado.
- El texto de *minutosRestantes* muestra el mensaje con el valor seleccionado de duración.
- Se muestra los parametros finales FES.
- Se activa el botón *InicioFES*.
- Se activa el botón *BotomProbar* se hace visible.
- Se activa el botón *AbrirMano* se habilita para las pruebas.
- Se activa el botón *CerrarMano* se habilita para las pruebas.
- Se activa el botón *BotomConfirmar* se habilita para las pruebas.

Figura 43.

Programación de botón Siguiente para visualizar FES(actividades diarias).

```

when SiguienteFES .Click
do
  set DeslizadorFrecuencia .ThumbEnabled to false
  set DeslizadorAnchoP .ThumbEnabled to false
  set DeslizadorTiempo .ThumbEnabled to false
  set AjusteParametros .Visible to false
  set RelojFES .TimerEnabled to false
  set FrecuenciaFinal .Text to get global Frecuencia
  set AnchoPFinal .Text to (get global AnchoP) × 10
  set DuracionFinal .Text to get global DuracionT
  set MinutosRestantes .Text to get global DuracionT
  set ParametrosFinalesFES .Visible to true
  set BotónInicioFES .Enabled to true
  set BotonProbar .Visible to true
  set AbrirMano .Enabled to true
  set CerrarMano .Enabled to true
  set BotonConfirmar .Enabled to true

```

Nota. Diseño propio

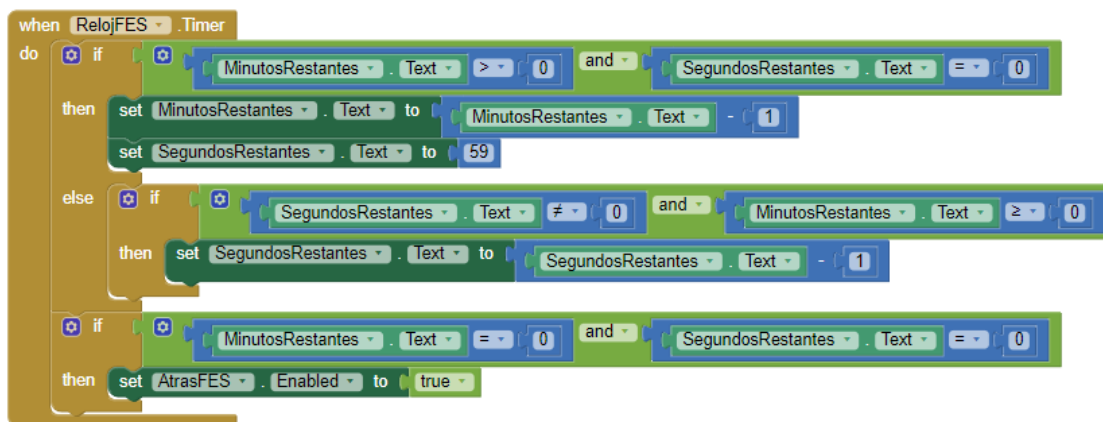
El manejo del conteo regresivo en la terapia debe tener su control en el tiempo, ya que es muy necesario llevar ese proceso avanzando poco a poco con la aplicación de la terapia. Así que, la estructura de este reloj llamado en la programación *RelojFES* es idéntico a la estructura de la terapia *fes(modoteraputico)*, tiene ciclos anidados *IF* y en su totalidad son tres, las variables para las condiciones son *MinutosRestantes* y *SegundosRestantes*, las condicionales a tener en cuenta son:

- *MinutosRestantes* mayor a 0 y *SegundosRestantes* igual a 0
- *MinutosRestantes* mayor o igual a 0 y *SegundosRestantes* diferente a 0
- *MinutosRestantes* mayor a 0 y *SegundosRestantes* igual a 0

Para cada caso, tiene sus propias acciones a realizar en el proceso del conteo regresivo, en la Fig. 44 se va a mostrar las acciones que se realiza si se cumple alguna de las condicionales.

Figura 44.

Programación de reloj para el conteo regresivo de FES(actividades diarias).



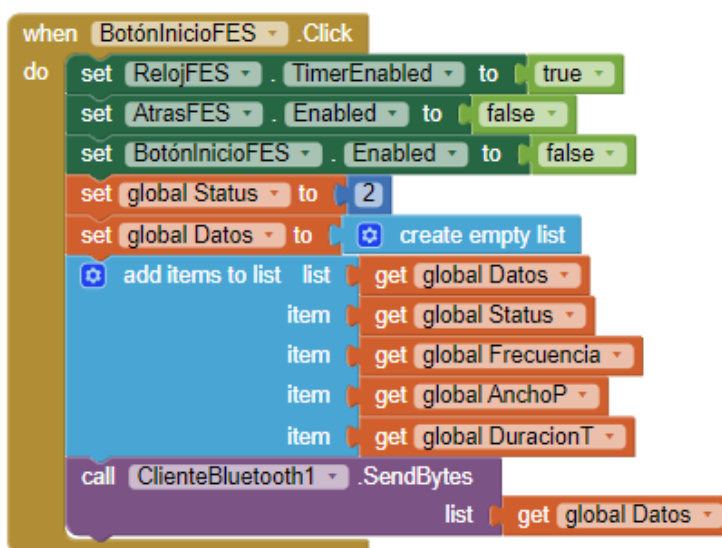
Nota. Diseño propio

Se ha programado un botón Inicio para que el paciente tenga la tranquilidad de poder tener el control del momento exacto para comenzar la terapia, estas son las acciones que se realizan a través de la programación para que se de inicio a la terapia:

- El Reloj de la terapia *RelojFES* se activa
- El botón de *AtrasFES* se deshabilita
- El botón de BotónInicioFES se deshabilita
- El Ajuste de parámetros se hace visible
- La variable global *Status* toma el valor de 2
- Se ingresa a la lista en blanco de global *Datos* las siguientes variables: Status, Frecuencia, AnchoP, DuracionT.
- Se envía la lista *Datos* por medio de bluetooth hacia la ESP-32.

Figura 45.

Programación de botón de inicio FES(actividades diarias).



Nota. Diseño propio

Adicional a ello, se programa un *botonDetenerFES* para poder tener el control de poder finalizar la terapia en cualquier momento y que se finalice de la manera correcta. a continuación una explicación de las funciones a realizar:

- El Reloj de la terapia *RelojFES* se detiene
- El botón de *AtrasFES* se habilita
- El texto *MinutosRestantes* muestra el valor de 0
- El texto *SegundosRestantes* muestra el valor de 0
- La variable global *Status* toma el valor de 3
- Se ingresa a la lista en blanco de global *Datos* las siguientes variables: *Status*, *Frecuencia*, *AnchoP*, *DuracionT*.
- Se envia la lista *Datos* por medio de bluetooth hacia la ESP-32.

Figura 46.

Programación de botón de detener FES(actividades diarias).

```

when BotónDetenerFES .Click
do
  set RelojFES . TimerEnabled to false
  set AtrasFES . Enabled to true
  set MinutosRestantes . Text to 0
  set SegundosRestantes . Text to 0
  set global Status to 3
  set global Datos to create empty list
  add items to list list
  item get global Datos
  item get global Status
  item get global Frecuencia
  item get global AnchoP
  item get global DuracionT
  call ClienteBluetooth1 .SendBytes
  list get global Datos
  
```

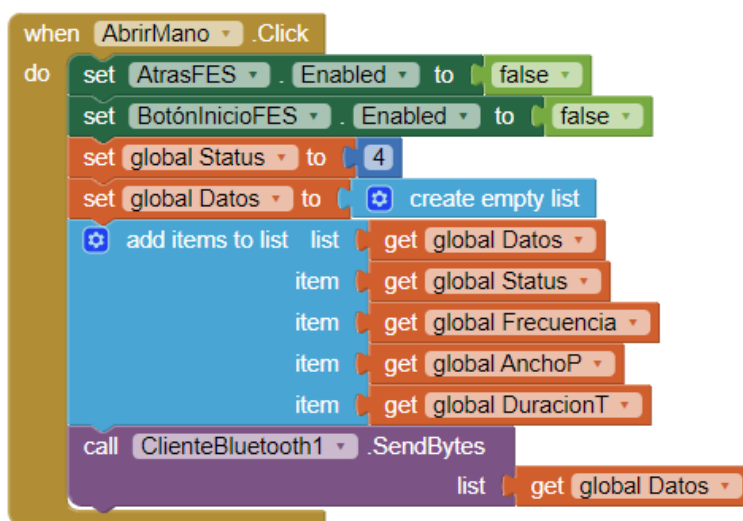
Nota. Diseño propio

Se continua con la programación del botón *AbrirMano* que deberá ejecutar las siguientes funciones:

- El botón de *AtrasFES* se deshabilita o se oculta.
- El botón *BotonInicioFES* se deshabilita o se oculta.
- La variable global *Status* toma el valor de 4.
- Se ingresa a la lista en blanco de global *Datos* las siguientes variables: *Status*, *Frecuencia*, *AnchoP*, *DuracionT*.
- Se envia la lista *Datos* por medio de bluetooth hacia la ESP-32.

Figura 47.

Programación del botón de Abrir Mano para FES(actividades diarias).



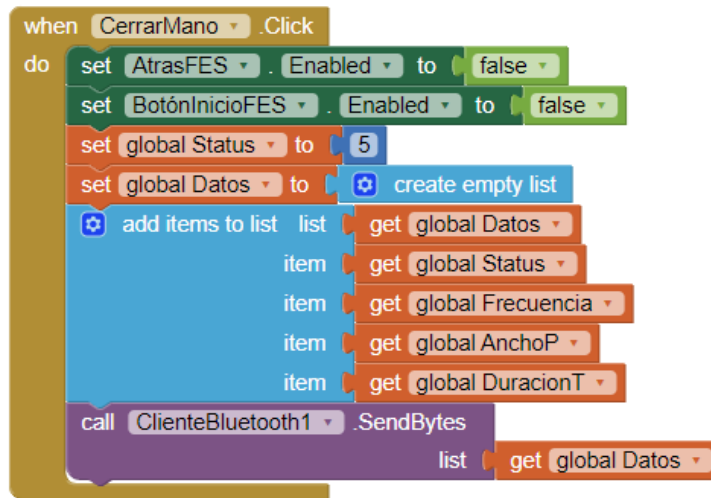
Nota. Diseño propio

La siguiente imagen se visualizará las funciones que están precedidas por el botón *CerrarMano* y se explicarán a groso modo en la siguiente lista:

- El botón de *AtrasFES* se deshabilita o se oculta.
- El botón *BotonInicioFES* se deshabilita o se oculta.
- La variable global *Status* toma el valor de 5.
- Se ingresa a la lista en blanco de global *Datos* las siguientes variables: *Status*, *Frecuencia*, *AnchoP*, *DuracionT*.
- Se envía la lista *Datos* por medio de bluetooth hacia la ESP-32.

Figura 48.

Programación del botón de Cerrar Mano para FES(actividades diarias).



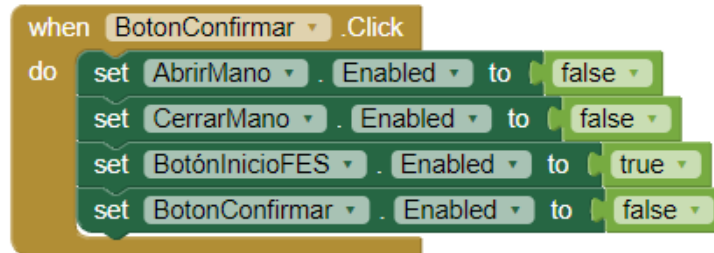
Nota. Diseño propio

Luego de todo lo anterior, se explicará el botón *BotonConfirmar* que de la misma forma tiene sus propias funciones, las cuales son mencionadas a continuación:

- El botón de *AbrirMano* se deshabilita o se oculta.
- El botón *BotonInicioFES* se habilita o se hace visible.
- El botón *CerrarMano* se deshabilita o se oculta.
- El botón *Botonconfirmar* se deshabilita o se oculta.

Figura 49.

Programación del botón de Confirmar para FES(actividades diarias).

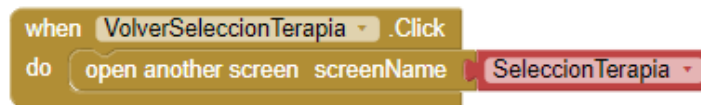


Nota. Diseño propio

Finalizando la explicación de la programación de la aplicación se tiene en cuenta el botón de volver a la selección de terapia en el screen de FES(actividades diarias), así que la función es cerrar el screen y volver al Screen de selección de terapia.

Figura 50.

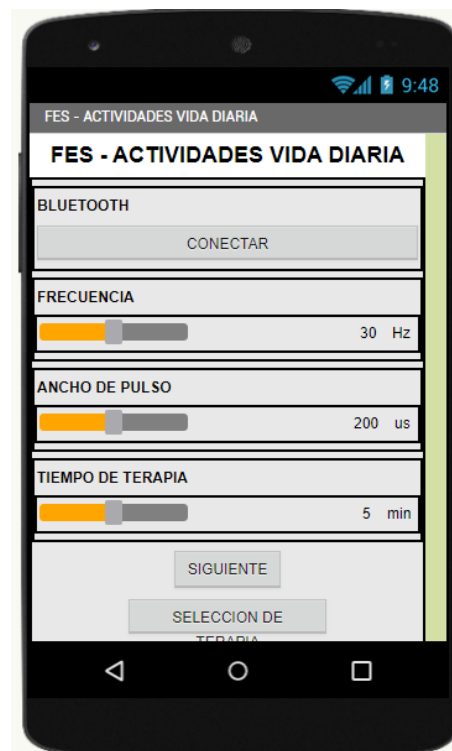
Programación de botón de volver a la selección de la terapia.



Nota. Diseño propio

Finalmente, se muestra el diseño del Screen de FES(actividades diarias) para la visualización en la aplicación móvil. en la siguiente Fig. 51

Figura 51.
Diseño del Screen Fes(activides diarias).



Nota. Diseño propio

4.4. Diseño de la PCB

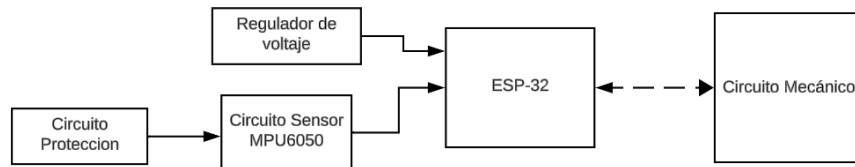
Este diseño se divide en 2 partes, una sensorial y una maestra. A continuación, se mostrará los circuitos correspondientes para la elaboración de las dos PCB.

4.4.1. PCB Sensorial.

A continuación se presenta un diseño de diagrama de flujos como representación sencilla del circuito para la parte sensorial; se visualiza las partes importantes del circuito madre.

Figura 52.

Diagrama de flujo del circuito Sensorial.

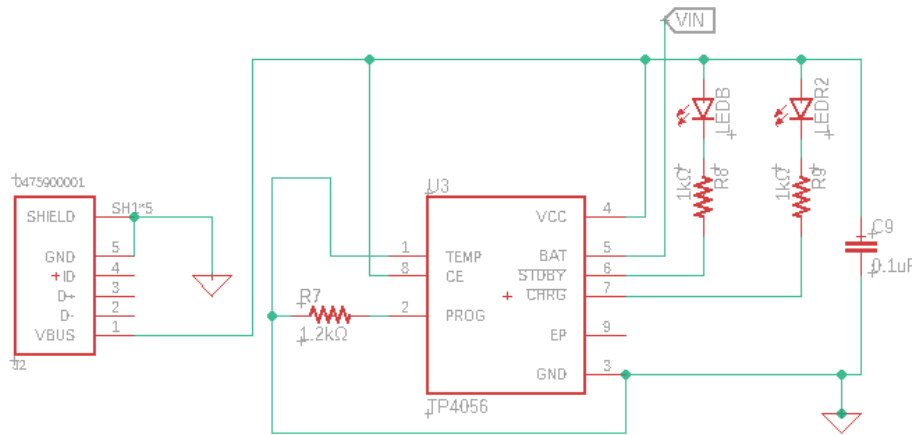


Nota. Diseño propio

Los siguientes diseños se desarrollaron en la aplicación EAGLE 9.6.2, un programa que permite generar esquemas de circuitos impresos. Esta PCB se construye a base de cuatro circuitos: Regulador de voltaje, ESP-32, circuito de protección y regulador de carga.

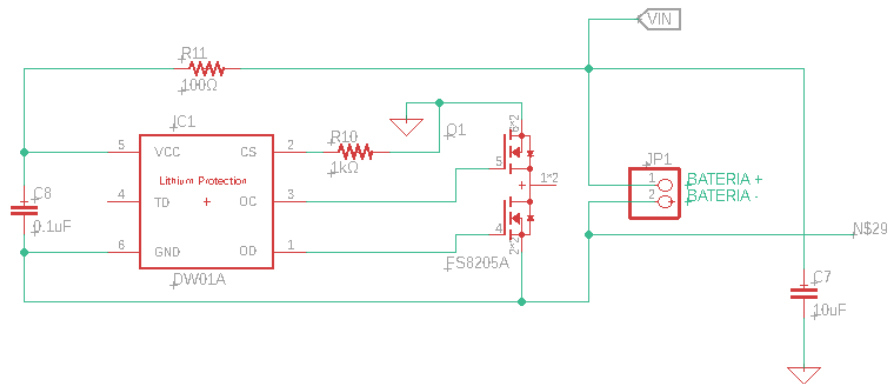
Figura 53.

Esquema del sensor Acelerómetro.



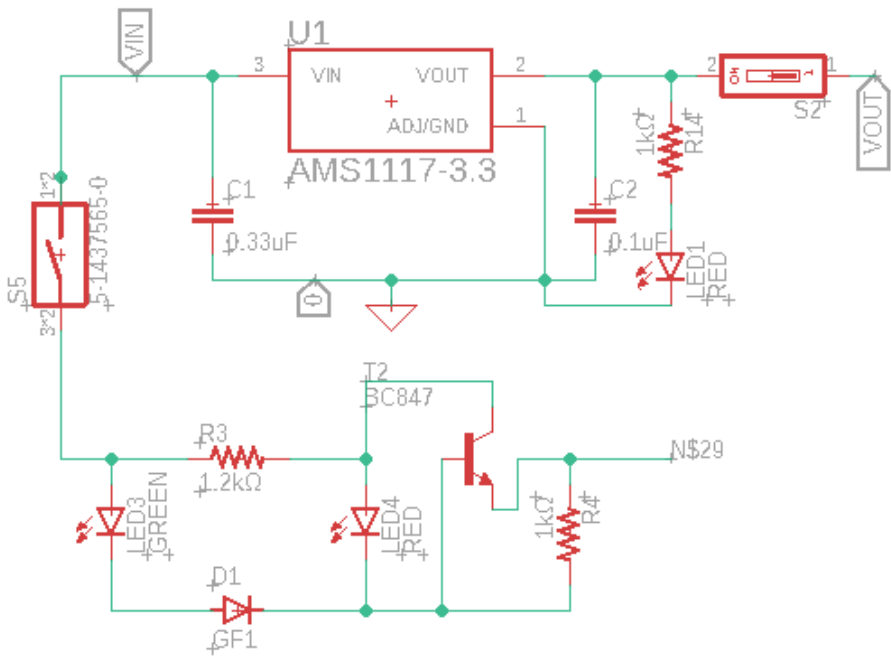
Nota. Diseño propio

Figura 54.
Esquema de circuito de protección.



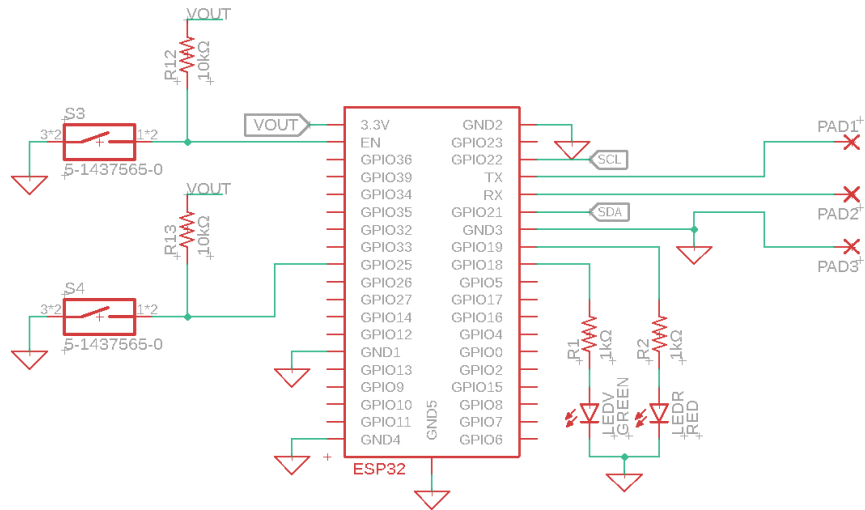
Nota. Diseño propio

Figura 55.
Esquema de regulador de voltaje.



Nota. Diseño propio

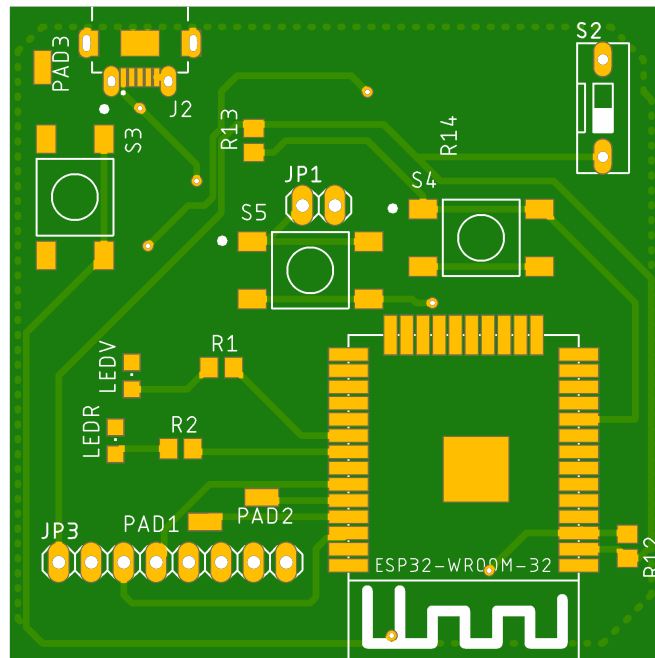
Figura 56.
Esquema de conexiones ESP-32.



Nota. Diseño propio

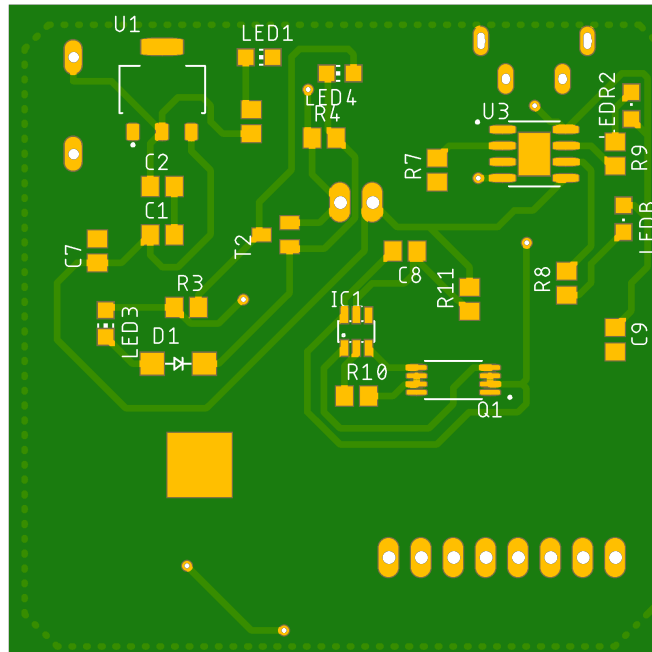
Al agrupar cada circuito mostrado anteriormente, se implementa el siguiente diseño de la PCB que tiene como dimensiones 6x6 cm y con 2 capas de plano GND.

Figura 57.
Circuito impreso sensorial parte superior.



Nota. Diseño propio

Figura 58.
Circuito impreso sensorial parte inferior.



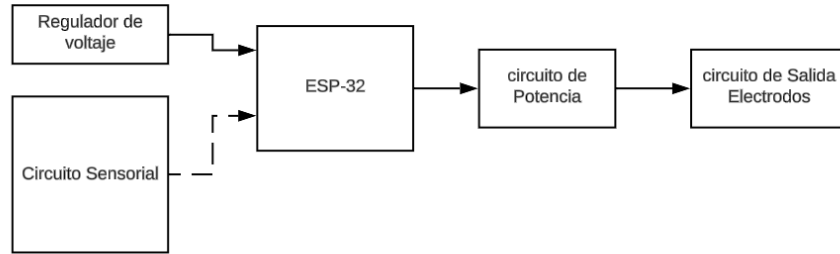
Nota. Diseño propio

4.4.2. PCB Maestra.

Antes de desglosar el circuito por partes, es necesario poder visualizar en un diagrama de flujo un esquema de división de este. A continuación se muestra el diagrama planteado

Figura 59.

Diagrama de flujo del circuito maestro.

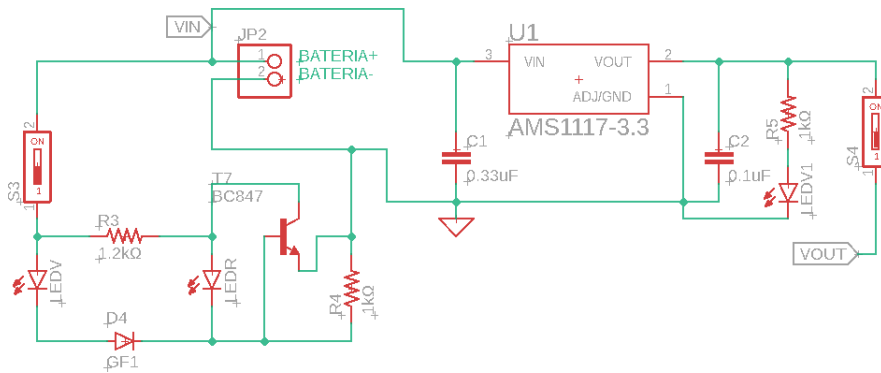


Nota. Diseño propio

Para esta sección de los circuitos impresos, también se utiliza el programa Eagle 9.6.2 y se realiza con un esquema de 4 circuitos: Regulador de voltaje, ESP-32, Circuito de salida de electrodos y circuito de potencia.

Figura 60.

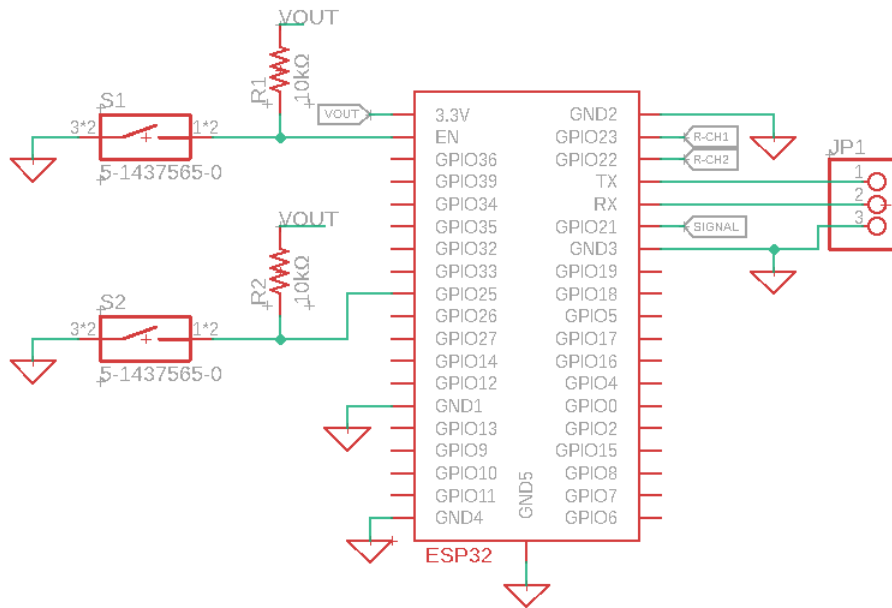
Esquema de regulador de voltaje.



Nota. Diseño propio

Figura 61.

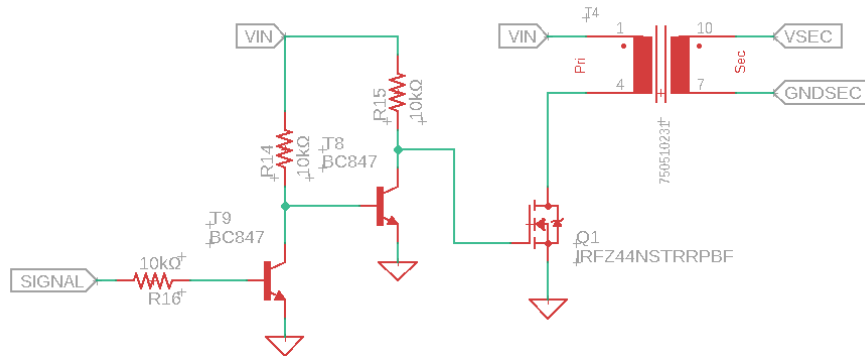
Esquema de conexiones de ESP-32.



Nota. Diseño propio

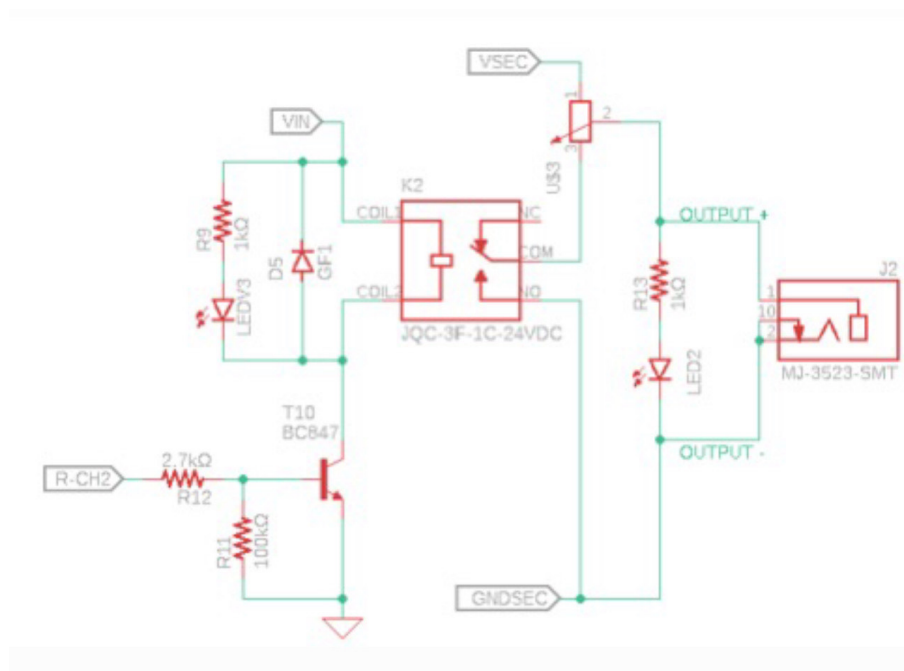
Figura 62.

Circuito de potencia.



Nota. Diseño propio

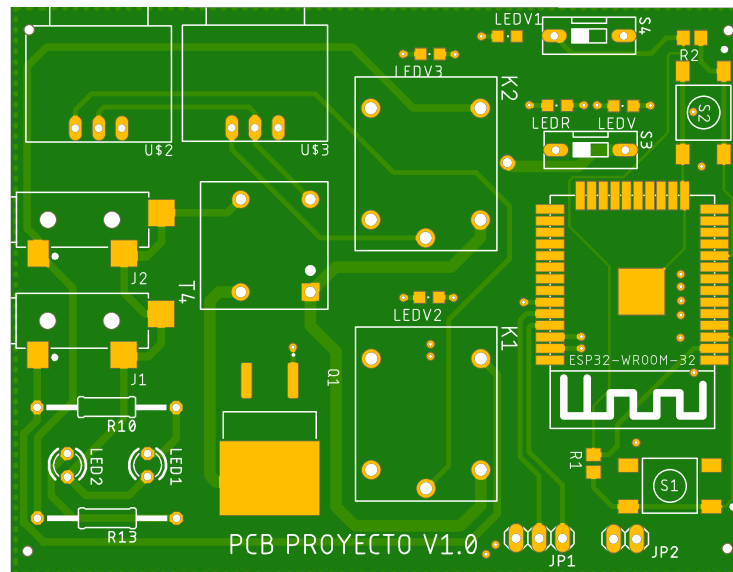
Figura 63.
Circuito de salida para electrodos.



Nota. Diseño propio

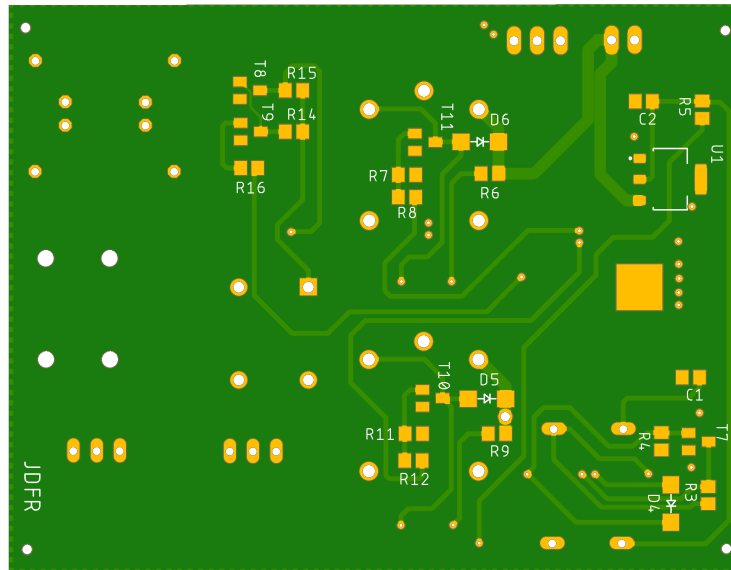
Para finalizar el diseño de las PCB, en esta sección se agrupa los 4 circuitos mencionados anteriormente con las dimensiones 6x8cm y 2 capas de planos GND.

Figura 64.
Circuito impreso maestro parte superior.



Nota. Diseño propio

Figura 65.
Circuito impreso maestro parte inferior.



Nota. Diseño propio

4.5. Electrodo

En electrónica, se denomina electrodo al extremo de un conductor eléctrico, al cual por el contacto físico con otro medio, transfiere o recibe información en forma de corriente eléctrica.

Para este proyecto, se utilizó dos electrodos de contacto físico, por lo cual, es no invasivo, y su función es transferir una corriente eléctrica seleccionada por el usuario por medio de la aplicación para el tratamiento prescrito por el fisioterapeuta, ya que si no se aplica de forma correcta la frecuencia o los impulsos de la corriente, no se tendría avances significativos en el proceso de recuperación. Añadiendo al uso correcto de los electrodos, también se debe tener en la cuenta, la posición correcta en el brazo, asegurando el estímulo en el nervio correspondiente.

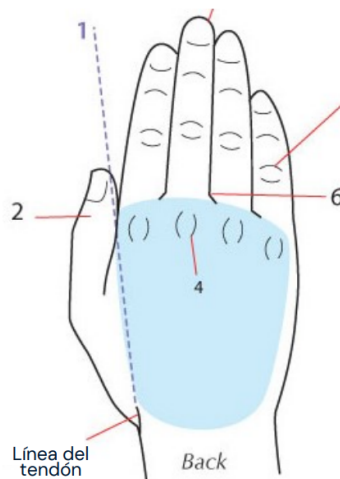
4.6. Diseño 3D

Para la sección de la estructura que protege las PCB, se debe realizar por aparte dos estructuras, ya que cada una debe ir en partes diferentes. La PCB sensorial va en el dorso de la mano funcional y la PCB maestra será ubicada en el

antebrazo, ya que es el lugar más seguro y cerca de los Nervios y/o músculos encargados del movimiento de la mano. Se empezará con la explicación de la carcasa de la PCB sensorial, esta pieza del diseño es diseñada de acuerdo a la forma de la mano, ya que así brindará un mayor ajuste y ergonomía al tiempo de ser usado. En la siguiente Figura se mostrará la posición ideal que se planteó al realizar el diseño de la carcasa de esta.

Figura 66.

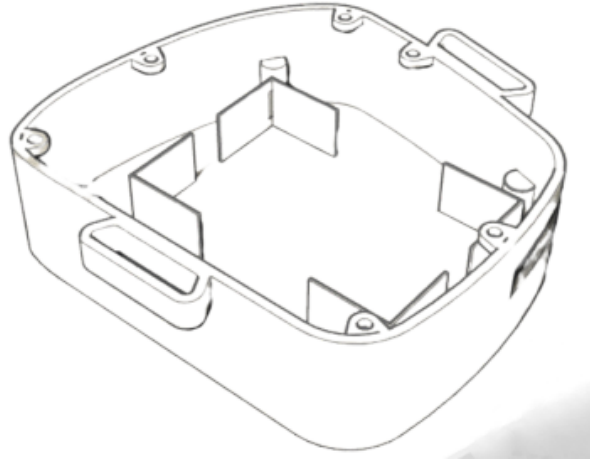
Posición ideal del dispositivo sensorial.



Nota. Diseño propio

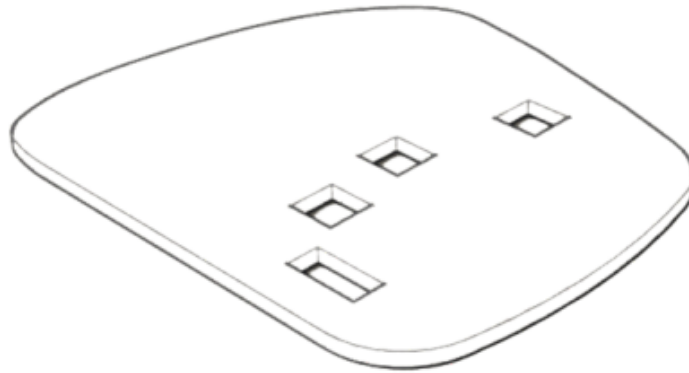
Basándonos en la posición de la mano, tomamos medidas generales y empezamos a diseñar la carcasa. En ella debe haber una pila recargable y la PCB sensorial, entonces, lo que se realizó fue una división vertical para poder encajar perfectamente los dos objetos en ella. Para más comprensión se mostrará la imagen del diseño por la vista frontal para poder ver la estructura completa.

Figura 67.
Diseño de la Carcasa vista frontal.



Nota. Diseño propio

Figura 68.
Diseño de Tapa vista frontal.

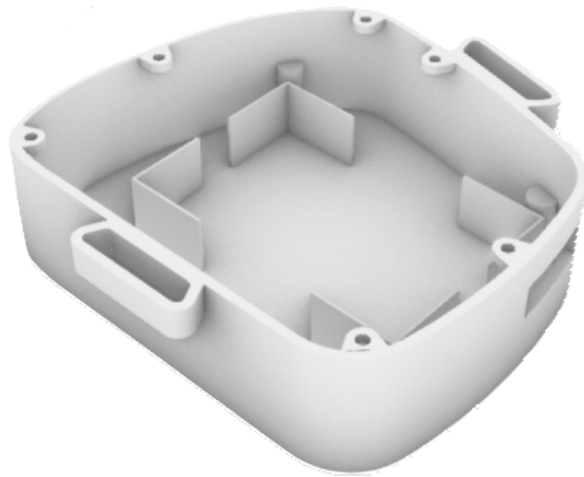


Nota. Diseño propio

Y luego de visualizar el diseño a trazos, y denotar que no hay ningún error en el diseño, podemos mostrar un diseño más adaptado a la realidad como por ejemplo, esta carcasa en 3D. Vamos a Visualizarla en la siguiente imagen.

Figura 69.

Diseño en 3D de Carcasa vista frontal.



Nota. Diseño propio

Figura 70.

Diseño en 3D de tapa vista frontal.

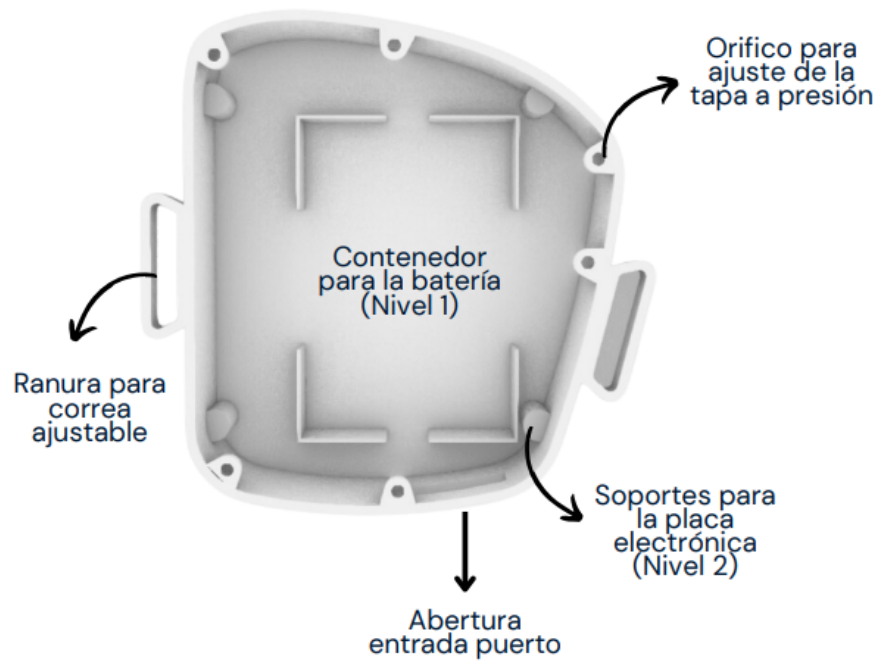


Nota. Diseño propio

Ahora bien, para poder asegurar perfectamente la tapa a la carcasa, se utiliza el método de presión, esto se realiza por medio de unos orificios en la carcasa donde encajan unos pines que se encuentran en la parte inferior de la tapa.

Figura 71.

Explicación del método de presión utilizado para la seguridad del dispositivo.



Nota. Diseño propio

Adicionalmente, esta es una forma de poder imaginar como se utilizaría este dispositivo, como se ha dicho anteriormente se ha realizado de acuerdo a la forma del dorso de la mano, así que vamos a visualizar en la siguiente imagen.

Figura 72.
Visualización del dispositivo en la posición ideal.



Nota. Diseño propio

Como ya se ha mencionado anteriormente, este dispositivo tiene dos componentes, uno sensorial, anteriormente explicado, y uno maestro. En este momento se explicará el diseño de la estructura. Para este diseño, se planteó la posición del brazo, ya que se obtiene una buena disposición del manejo de él y además puede ser ajustable al tamaño del brazo. Se mostrará en la siguiente imagen un modelo que funcionó como inicio para la elaboración del diseño.

Figura 73.
Posición ideal del dispositivo maestro.

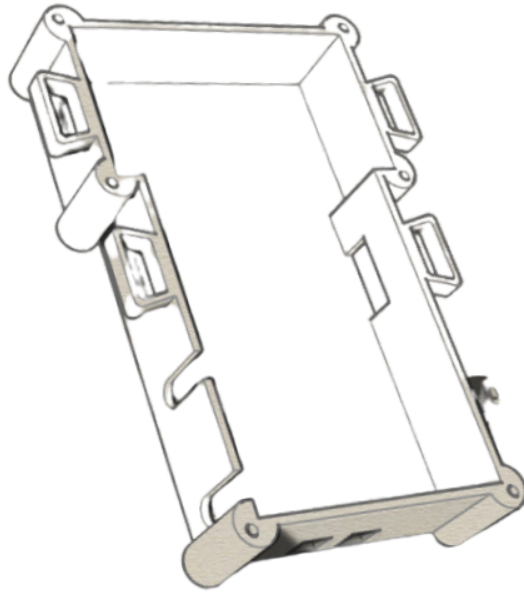


Nota. Diseño propio

Basándonos en este modelo, se empezó a realizar el diseño de la carcasa, objetos que debemos tener en cuenta para esta, es un portabaterías y la PCB. La visualización del diseño es el portabaterías parte superior de la carcasa. En la siguiente figura se dará una representación de la forma del diseño de la carcasa.

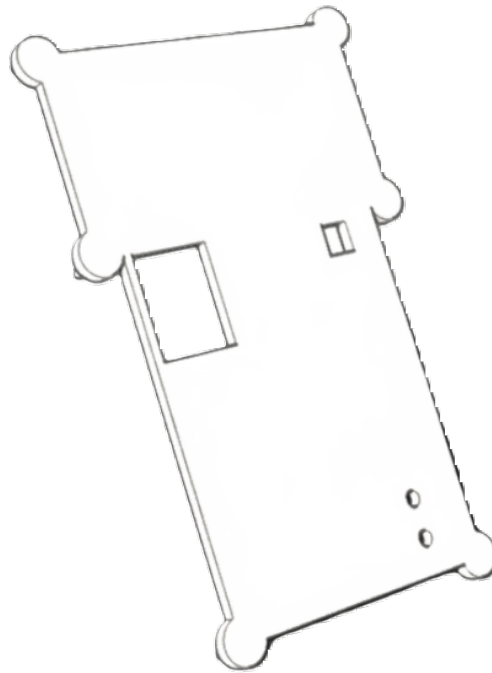
Figura 74.

Diseño de la Carcasa vista frontal.



Nota. Diseño propio

Figura 75.
Diseño de la Tapa vista frontal.

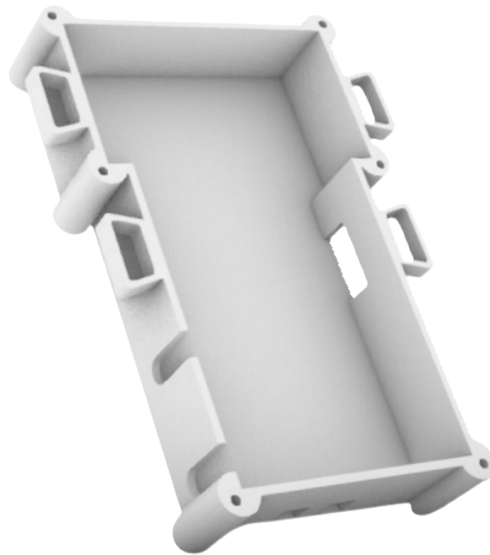


Nota. Diseño propio

Posteriormente a ello, se debe estar seguro que todo el diseño este con las medidas exactas y así poder visualizar en un modo que se parezca a la realidad, como en 3D, ya que así nos dará una idea representativa del diseño.

Figura 76.

Diseño en 3D de Carcasa vista frontal.



Nota. Diseño propio

Figura 77.

Diseño en 3D de la tapa vista frontal.

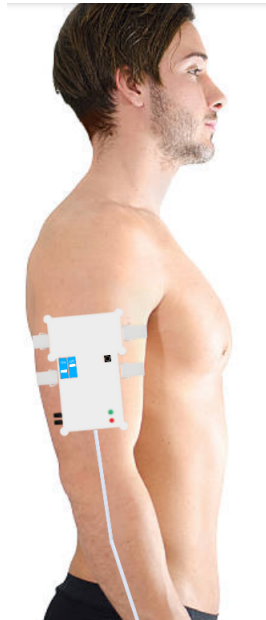


Nota. Diseño propio

Se utilizará el mismo método de seguridad que se usó en la carcasa sensorial. Este método de presión es uno de los mas seguros y más rápidos para poder abrir cuando se necesita reparar algún elemento. Para Finalizar se visualizará la posición ideal del diseño del dispositivo.

Figura 78.

Visualización del dispositivo en la posición ideal.

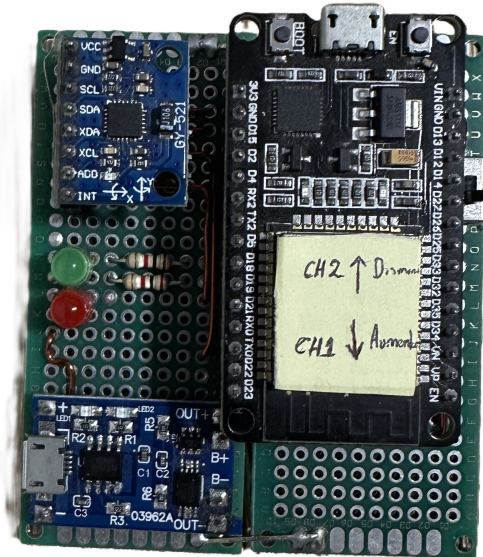


Nota. Diseño propio

5. Desarrollo del dispositivo FES

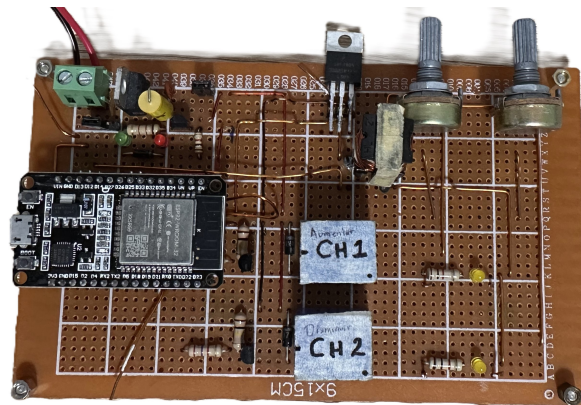
Para poder realizar una exhaustiva validación del circuito impreso, se realiza una PCB de prueba, esto brinda que si algún elemento necesite modificarse en algún momento sea mas fácil. Se ejecuta la comprobación de las conexiones y el buen funcionamiento para luego hacer el respectivo circuito impreso. En las siguientes imágenes se mostrará la PCB de prueba.

Figura 79.
Circuito prototipo Prueba sensorial.



Nota. Diseño propio

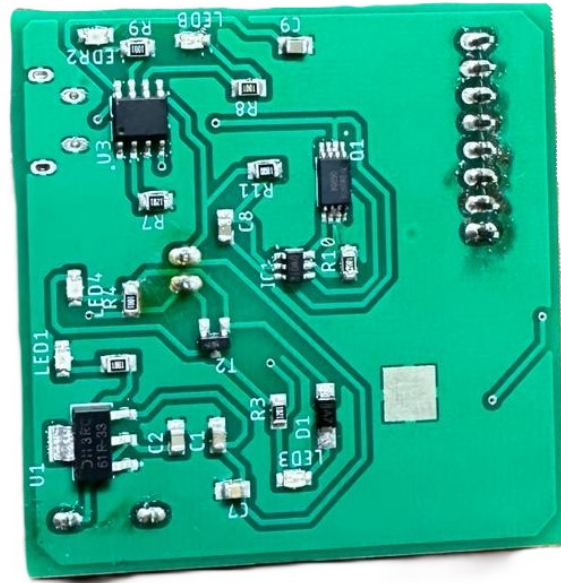
Figura 80.
Circuito prototipo Prueba maestra.



Nota. Diseño propio

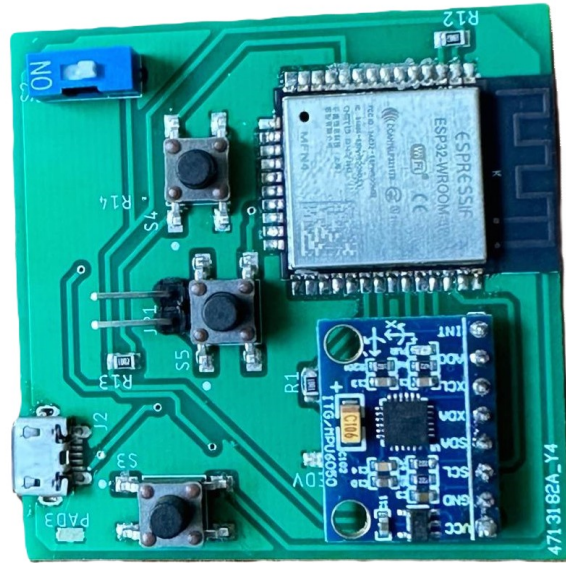
Luego de hacer las respectivas validaciones y modificaciones necesarias para el buen funcionamiento del prototipo en el circuito de prueba, ahora se podrá realizar el diseño de la PCB, la cual se implementará y se realizarán los respectivos análisis para corroborar que todo haya quedado en perfecto estado.

Figura 81.
PCB sensorial Vista Bottom.



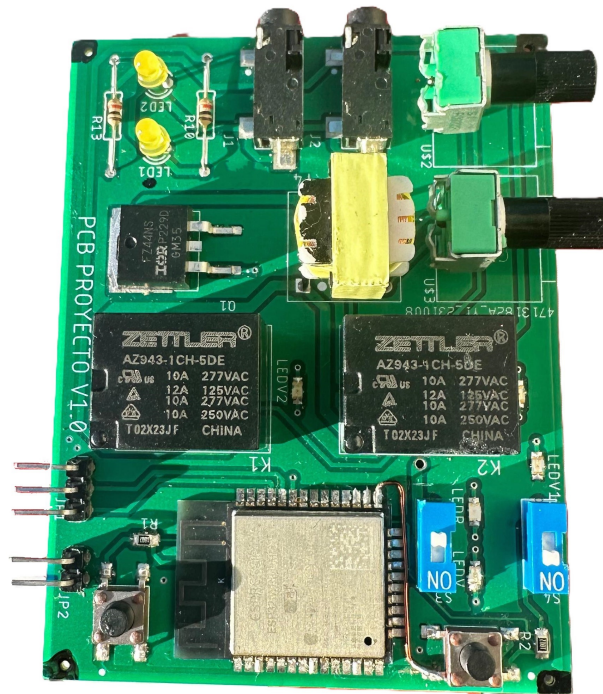
Nota. Diseño propio

Figura 82.
PCB sensorial vista Top.



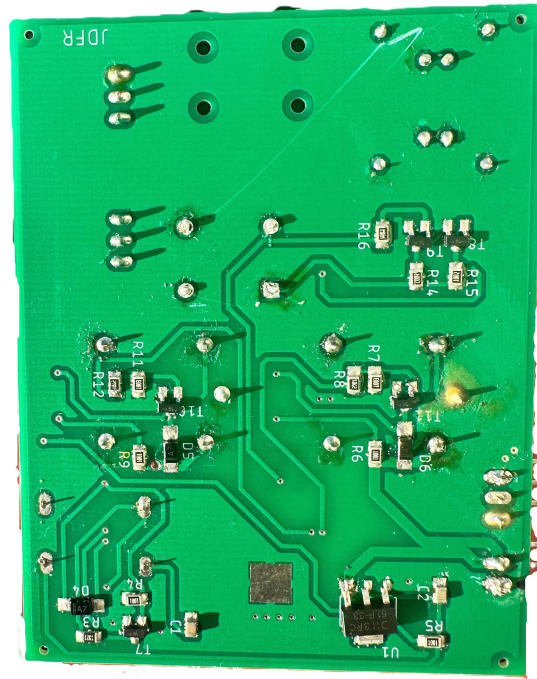
Nota. Diseño propio

Figura 83.
PCB maestra vista Top.



Nota. Diseño propio

Figura 84.
PCB maestra vista bottom.



Nota. Diseño propio

6. Validación

Para demostrar el funcionamiento del prototipo se usó el programa modulo serial del Arduino y osciloscopio digital. Se evidencia el desempeño de respuesta del sensor de acelerómetro en el módulo serial con respuestas como: aumentar, punto muerto y disminuir. En las siguientes fig.85, 86, 87, 88 se muestra la velocidad de respuesta del cambio entre cada fase.

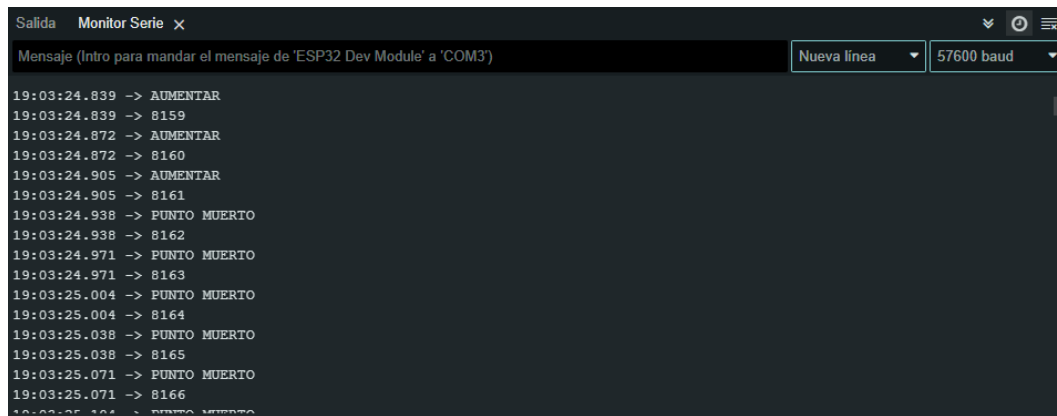
Figura 85.
Cambio del punto muerto a extensión.



```
Salida Monitor Serie x
Mensaje (Intro para mandar el mensaje de 'ESP32 Dev Module' a 'COM3') Nueva línea 57600 baud
19:03:23.715 -> 8125
19:03:23.748 -> PUNTO MUERTO
19:03:23.748 -> 8126
19:03:23.781 -> PUNTO MUERTO
19:03:23.781 -> 8127
19:03:23.814 -> PUNTO MUERTO
19:03:23.814 -> 8128
19:03:23.847 -> PUNTO MUERTO
19:03:23.847 -> 8129
19:03:23.881 -> PUNTO MUERTO
19:03:23.881 -> 8130
19:03:23.913 -> AUMENTAR
19:03:23.913 -> 8131
19:03:23.946 -> AUMENTAR
19:03:23.946 -> 8132
19:03:23.979 -> AUMENTAR
19:03:23.979 -> 8133
```

Nota. Diseño propio

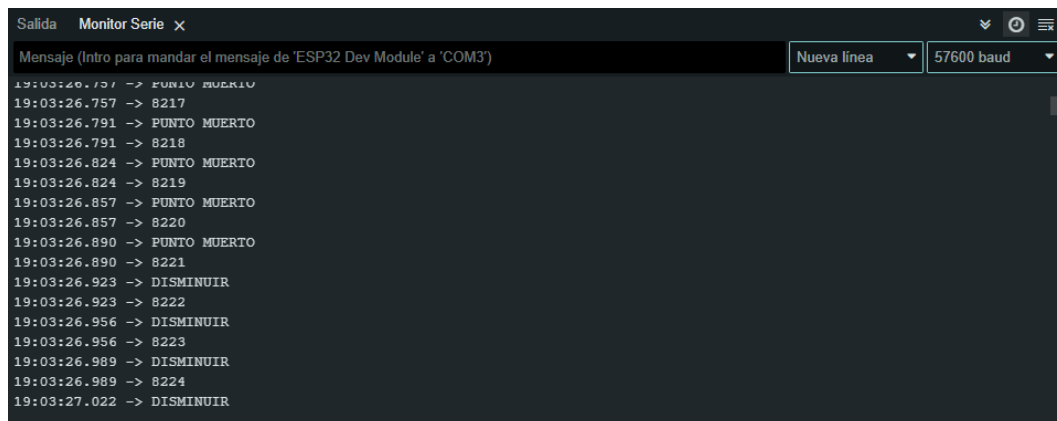
Figura 86.
Cambio de extensión a punto muerto.



```
Salida Monitor Serie x
Mensaje (Intro para mandar el mensaje de 'ESP32 Dev Module' a 'COM3') Nueva línea 57600 baud
19:03:24.839 -> AUMENTAR
19:03:24.839 -> 8159
19:03:24.872 -> AUMENTAR
19:03:24.872 -> 8160
19:03:24.905 -> AUMENTAR
19:03:24.905 -> 8161
19:03:24.938 -> PUNTO MUERTO
19:03:24.938 -> 8162
19:03:24.971 -> PUNTO MUERTO
19:03:24.971 -> 8163
19:03:25.004 -> PUNTO MUERTO
19:03:25.004 -> 8164
19:03:25.038 -> PUNTO MUERTO
19:03:25.038 -> 8165
19:03:25.071 -> PUNTO MUERTO
19:03:25.071 -> 8166
19:03:25.104 -> PUNTO MUERTO
```

Nota. Diseño propio

Figura 87.
Cambio del punto muerto a flexión.



```
Salida Monitor Serie x
Mensaje (Intro para mandar el mensaje de 'ESP32 Dev Module' a 'COM3') Nueva línea 57600 baud
19:03:26.757 -> PUNTO MUERTO
19:03:26.757 -> 8217
19:03:26.791 -> PUNTO MUERTO
19:03:26.791 -> 8218
19:03:26.824 -> PUNTO MUERTO
19:03:26.824 -> 8219
19:03:26.857 -> PUNTO MUERTO
19:03:26.857 -> 8220
19:03:26.890 -> PUNTO MUERTO
19:03:26.890 -> 8221
19:03:26.923 -> DISMINUIR
19:03:26.923 -> 8222
19:03:26.956 -> DISMINUIR
19:03:26.956 -> 8223
19:03:26.989 -> DISMINUIR
19:03:26.989 -> 8224
19:03:27.022 -> DISMINUIR
```

Nota. Diseño propio

Figura 88.
Cambio de flexión a punto muerto.

```

Salida  Monitor Serie x
Mensaje (Intro para mandar el mensaje de 'ESP32 Dev Module' a 'COM3') Nueva línea 57600 baud
19:03:26.757 -> PUNTO MUERTO
19:03:26.757 -> B217
19:03:26.791 -> PUNTO MUERTO
19:03:26.791 -> B218
19:03:26.824 -> PUNTO MUERTO
19:03:26.824 -> B219
19:03:26.857 -> PUNTO MUERTO
19:03:26.857 -> B220
19:03:26.890 -> PUNTO MUERTO
19:03:26.890 -> B221
19:03:26.923 -> DISMINUIR
19:03:26.923 -> B222
19:03:26.956 -> DISMINUIR
19:03:26.956 -> B223
19:03:26.989 -> DISMINUIR
19:03:26.989 -> B224
19:03:27.022 -> DISMINUIR
  
```

Nota. Diseño propio

Ahora bien, para la respuesta de los estímulos de salida hacía el paciente se utilizó una resistencia de carga $1k\Omega$ representando la resistencia que el cuerpo tiene hacia un pulso eléctrico y se usa el osciloscopio digital como visualización de la salida de las señales. A continuación, se evidencia las señales de salida del dispositivo.

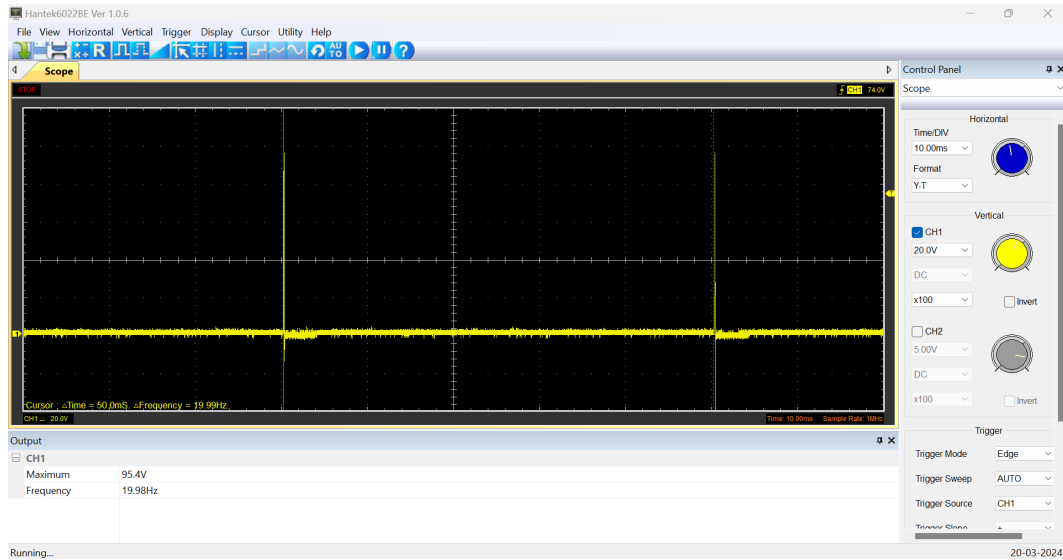
Para calcular el parámetro del error porcentual se debe tener en cuenta la siguiente ecuación:

$$E = \left[\frac{(V_{medido} - V_{app})}{V_{app}} \right] * 100 \quad (1)$$

Frecuencia mínima:

Figura 89.

Respuesta de estímulo con frecuencia mínima 20Hz.



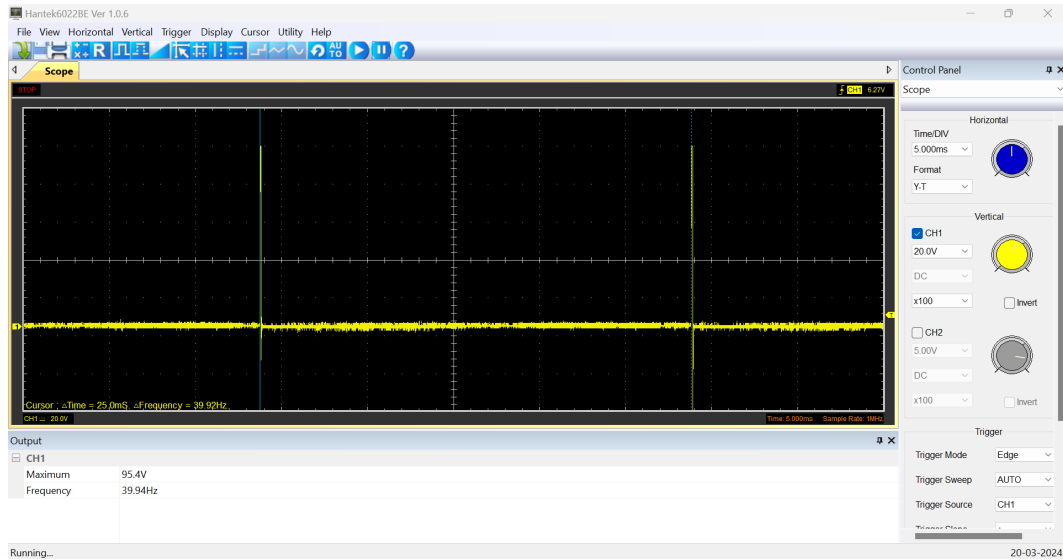
Nota. Diseño propio

$$E = \left[\frac{19.98 - 20}{20} \right] * 100 = 0.1\% \quad (2)$$

Frecuencia Máxima

Figura 90.

Respuesta de estímulo con frecuencia máxima 40Hz.



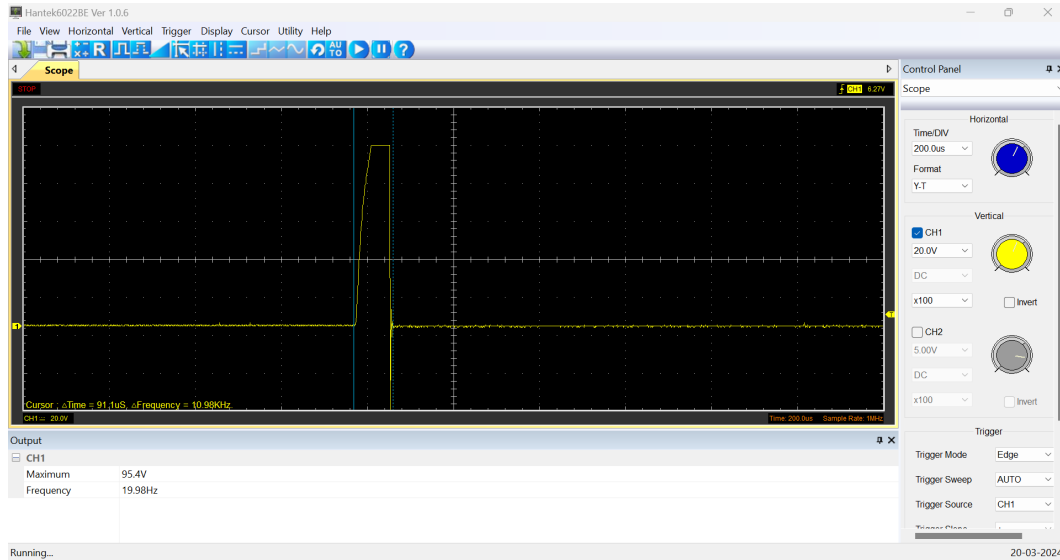
Nota. Diseño propio

$$E = \left[\frac{39.94 - 40}{40} \right] * 100 = 0.15\% \quad (3)$$

Amplitud mínima:

Figura 91.

Respuesta de estímulo con amplitud mínima $100\mu\text{s}$.



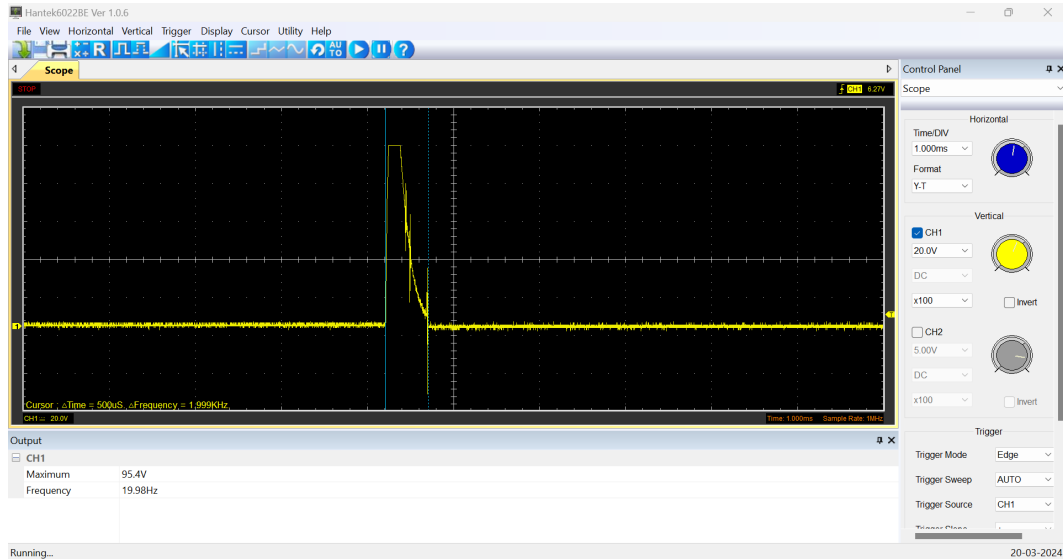
Nota. Diseño propio

$$E = \left[\frac{91.1 - 100}{100} \right] * 100 = 8.9\% \quad (4)$$

Amplitud máxima:

Figura 92.

Respuesta de estímulo con amplitud máxima $500\mu\text{s}$.



Nota. Diseño propio

$$E = \left[\frac{500 - 500}{500} \right] * 100 = 0\% \quad (5)$$

7. Conclusiones

Durante el desarrollo de este prototipo, se logró diseñar un dispositivo versátil y adaptable que cumple con los requisitos de funcionalidad y ajustabilidad requeridos para satisfacer las necesidades de los pacientes en rehabilitación de mano. Se ha demostrado que el prototipo es capaz de proporcionar estimulación eléctrica de manera efectiva con el fin de facilitar así la recuperación de la funcionalidad de la mano.

Así mismo, la interfaz de usuario ha sido diseñada en una aplicación para sistema operativo Android con un enfoque centrado en la usabilidad, garantizando una interacción intuitiva para los usuarios de diferentes niveles de experiencia. La aplicación proporciona una plataforma eficiente y conveniente para controlar el dispositivo de rehabilitación de manera inalámbrica. En conjunto, la aplicación móvil desarrollada no solo cumple con los requisitos establecidos, sino que también promueve la autonomía del paciente y facilita un seguimiento más efectivo del proceso de rehabilitación basada en FES.

Finalmente, el diseño del prototipo se centró en la accesibilidad y la facilidad de uso, lo que garantiza que los profesionales de rehabilitación física y los pacientes puedan utilizarlo fácilmente. Además se ha incorporado un enfoque ergonómico para asegurar que el dispositivo sea cómodo de sostener y manipular durante las sesiones de rehabilitación. Este prototipo ofrece una solución práctica y eficaz para la rehabilitación de mano, permitiendo a los pacientes integrar el dispositivo en su rutina diaria y mejorar su funcionalidad con el tiempo.

8. Recomendaciones

Si bien este prototipo de estimulación eléctrica funcional es el primer acercamiento a un dispositivo médico, para lograr un uso comercial hace falta de una aprobación médica, junto a testeos rigurosos. En consecuencia se dan las siguientes recomendaciones:

- Optimizar y mejorar el diseño de los circuitos impresos, mejorando las dimensiones de estos permitiendo así mayor comodidad y estética al paciente.
- Ampliar la funcionalidad del prototipo a otras partes del miembro superior dando así un mayor rango de uso del dispositivo.
- Implementar el uso de otros tipos de ondas relacionados a la electroterapia como ondas bifásicas balanceadas, pulsos triangulares y/o tren de impulsos.

Referencias Bibliográficas

- Aguar, M. M. (2016). Diseño de un sistema de rehabilitación activa con estimulación eléctrica. *Universidad politécnica de Valencia- Escuela tecnica superior ingenieros industriales Valencia*.
- Ambrosio, R., Borunda, J., Sifuentes, E., Heredia, A., Guerrero, F., and González, V. (2015). Prototipo de estimulación funcional eléctrica (fes) inalámbrico y de corriente constante para rehabilitación de extremidades. http://www-optica.inaoep.mx/~tecnologia_salud/2015/memorias/pdf/MyT2015_55_E.pdf.
- Bioness Inc. (2018). Bioness handmaster h200 wireless. <https://www.bionessuniversity.com/documents/MP612-00964-001>
- Castellano, M. (2017). Electroestimulador para miembro superior controlado por computadora. *Universidad Nacional de Córdoba, Córdoba, República Argentina*.
- Corredor Pastor, M. F. (2017). Instrumentación para dispositivo biomecánico corrector de pie caído tipo "afo". *Pontificia Universidad Javeriana, Bogotá D.C.*
- Cruz, L. A. (2018). Electroterapia para el fortalecimiento muscular. *Universidad Inca Garcilaso De La Vega*.
- de Salud y Protección Social, M. (2019). minsalud destaca las acciones en beneficio-de-la-poblacion-con-discapacidad. <https://www.minsalud.gov.co/Paginas/minsalud-destaca-las-acciones-en-beneficio-de-la-poblacion-con-discapacidad.aspx>.
- Flores Ramos, D. M. (1998). Actualizacion en electroterapia : comparacion de la efectividad de la estimulacion electrica funcional con el tratamiento rehabilitatorio tradicional en el paciente hemidejico secundario a evento vascular. *Universidad Nacional Autónoma de México*.
- Guerrero Martinez, J. F. (2011). Bioseñales. *Escuela Técnica Superior de Ingeniería, Universidad De Valencia*.

- Ortiz, J. M. and Ruiz, A. T. (2014). Diseño de un equipo de electroterapia para tratamiento de gonartrosis controlado des un smartphone. *CBEB 2014. Uberlândia, Brasil.*
- Portillo Martínez, E. O. (2010). Diseño conceptual órtesis mecatrónica para la rehabilitación de la parálisis parcial. *Universidad nacional autónoma de México.*
- Quintero Gonzalez, D. M. and Ferrer Betancur, C. (2015). Efectividad de la electroestimulación y la fisioterapia en el paciente critico. *Universidad CES, Medellin.*
- Saebo MyoTrac Infiniti (2021). Saebo myotrac infinity. <https://www.saebo.com/wp-content/uploads/2018/06/product-manual-smti.pdf>.
- Serna, A., Ros, F., and Rico, J. (2010). *Guía práctica de sensores*. Creaciones Copyright SL.
- Shenzhen XFT Medical Limited (2023). Xft-2003ea (shenzhen xft medical limited, china). <https://www.xft-china.com/products/xft-2003ea-h2-hand-rehab-system.html>.
- Zapata Vélez de Villa, M. A. (2010). Diseño de un estimulador eléctrico funcional para la rehabilitación física en miembros distales superiores con disfunción motriz. *Pontificia universidad católica del Perú.*

Apéndice A. Manual de usuario

Componentes del dispositivo

- Electrodo
- Dispositivo Sensorial
- Dispositivo Maestro
- Cargador de batería

Instrucciones

1. Colocar el dispositivo maestro en el brazo parético.

Figura A1.

Dispositivo maestro en el brazo parético.



Nota. Diseño propio

2. Posicionar 2 electrodos del mismo canal en la cara superior del antebrazo.

Figura A2.

Posicionamiento de 2 electrodos en la cara superior del antebrazo.



Nota. Diseño propio

3. Conectar los electrodos al dispositivo maestro.

Figura A3.

Conexión de los electrodos al dispositivo maestro.



Nota. Diseño propio

4. Posicionar 2 electrodos del mismo canal en la cara anterior del antebrazo.

Figura A4.

Posicionamiento de 2 electrodos en la cara anterior del antebrazo.

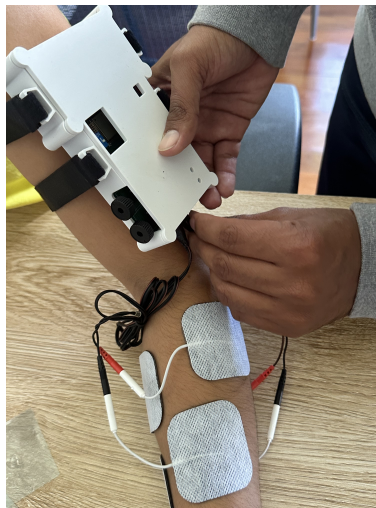


Nota. Diseño propio

5. Conectar los electrodos al dispositivo maestro.

Figura A5.

Conexión de los electrodos al dispositivo maestro.



Nota. Diseño propio

6. Colocar el dispositivo sensorial en la mano funcional.

Figura A6.
Dispositivo sensorial en la mano funcional.



Nota. Diseño propio

7. Encender los dispositivos maestro y sensorial.

Figura A7.
Botón de encendido del dispositivo maestro.



Nota. Diseño propio

8. Ingresar a la aplicación en el teléfono móvil.

Figura A8.
Aplicación en el teléfono móvil.



Nota. Diseño propio

9. Darle clic a siguiente.

Figura A9.
Aplicación en el teléfono móvil.



Nota. Diseño propio

10. Escoger el modo de terapia (FES-terapéutico o FES-actividades diarias).

Figura A10.

Modo de terapia (FES-terapéutico o FES-actividades diarias).

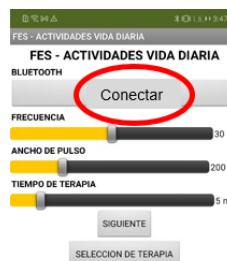


Nota. Diseño propio

11. Conectar a Bluetooth.

Figura A11.

Conexión a Bluetooth.



Nota. Diseño propio

12. Ajustar parámetros de la terapia (Frecuencia, ancho de pulso y duración de la terapia) y dar clic en siguiente.

Figura A12.

Ajuste parámetros y botón siguiente.



Nota. Diseño propio

13. Calibrar la intensidad de corriente aplicada.

Figura A13.

Calibración de la intensidad de corriente aplicada.



Nota. Diseño propio

14. Iniciar terapia.

Figura A14.
Iniciar terapia.



Nota. Diseño propio

Apéndice B. Códigos de programación

A continuación se muestran los códigos de programación usados en las ESP-32 maestra y sensorial realizados en la plataforma Arduino IDE.

■ Código ESP-32 Maestra

```
//LIBRERIAS
#include "BluetoothSerial.h"
#include <esp_now.h>
#include <WiFi.h>

#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run 'make menuconfig' to and
    enable it
#endif

typedef struct struct_message {
    float AnguloX;
    float AnguloY;
} struct_message;

struct_message Angulos;

int incoming[4];

int PinRele1 = 23;
int PinRele2 = 22;
int PinSenal = 21;

int Estado = 0;
int Frecuencia = 50;
int AnchoP = 20;
int Duracion = 5;

int j;
int CicloT;

double Duracionms;
double Tms = 1;
double veces;
double vecesP;
double Ton;
double Tonus;
double Toff;
double Toffus;
```

```

void OnDataRecv(const uint8_t * mac, const uint8_t
*incomingData, int len) { memcpy(&Angulos, incomingData, sizeof(Angulos))
;
}

BluetoothSerial BT;

void setup() {
  Serial.begin(57600);

  WiFi.mode(WIFI_STA);

  if (esp_now_init() != ESP_OK) {
    Serial.println("Error_initializing_ESP-NOW");
    return;
  }

  esp_now_register_recv_cb(OnDataRecv);

  BT.begin("ESP32_LED_Control");
  Serial.println("El_dispositivo_Bluetooth_esta_listo_para_emparejarse");

  pinMode(PinRele1, OUTPUT);
  pinMode(PinRele2, OUTPUT);
  pinMode(PinSenal, OUTPUT);
}

void loop() {
  if (BT.available()) {

    incoming[0] = BT.read();
    Estado = incoming[0];
    Serial.println(Estado);
    incoming[1] = BT.read();
    Frecuencia = incoming[1];
    incoming[2] = BT.read();
    AnchoP = incoming[2] * 10;
    incoming[3] = BT.read();
    Duracion = incoming[3];

  }

  Tms = 1000 / Frecuencia;
  Ton = AnchoP * 0.001;
  Toffus = (Tms - Ton) * 1000;

  Duracionms = Duracion * 60000;
}

```

```
veces = round(Duracionms / Tms);
vecesP = round(5000 / Tms);

CicloT = 10 * Frecuencia;

if (Estado == 1) {
    digitalWrite (PinRele1 , LOW);
    digitalWrite (PinRele2 , LOW);
    Serial.println ("TENS");
    for (int i = 0; i < veces; i++) {
        digitalWrite (PinSenal , HIGH);
        delayMicroseconds (AnchoP);
        digitalWrite (PinSenal , LOW);
        delayMicroseconds (Toffus);
        Serial.println (i);
        Serial.println (j);
        j = j + 1;

        if(j > 0 and j < CicloT){
            digitalWrite (PinRele1 , HIGH);
            digitalWrite (PinRele2 , LOW);
            Serial.println ("ABRIR_MANO");
        }
        else if(j > CicloT and j < 2*CicloT){
            digitalWrite (PinRele1 , LOW);
            digitalWrite (PinRele2 , LOW);
            Serial.println ("DESCANSO");
        }
        else if(j > 2*CicloT and j < 3*CicloT){
            digitalWrite (PinRele1 , LOW);
            digitalWrite (PinRele2 , HIGH);
            Serial.println ("CERRAR_MANO");
        }
        else if(j > 3*CicloT and j < 4*CicloT){
            digitalWrite (PinRele1 , LOW);
            digitalWrite (PinRele2 , LOW);
            Serial.println ("DESCANSO");
        }
        else if(j == 4*CicloT){
            j = 0;
        }

        if(BT.available()){
            incoming[0] = BT.read();
            Estado = incoming[0];
            if(Estado == 3){
                digitalWrite (PinRele1 , LOW);
                digitalWrite (PinRele2 , LOW);
                break;
            }
        }
    }
}
```

```

    }
  }
}
Estado = 3;
}

if (Estado == 2) {
  digitalWrite (PinRele1 , LOW);
  digitalWrite (PinRele2 , LOW);
  Serial.println ("FES");
  for (int i = 0; i < veces; i++) {

    digitalWrite (PinSenal , HIGH);
    delayMicroseconds (AnchoP);
    digitalWrite (PinSenal , LOW);
    delayMicroseconds (Tofus);

    if (Angulos.AnguloX > 10 and Angulos.AnguloX < 90) {
      digitalWrite (PinRele1 , HIGH);
      digitalWrite (PinRele2 , LOW);
      Serial.println ("AUMENTAR");
      if(BT.available()){
        incoming[0] = BT.read();
        Estado = incoming[0];
        if(Estado == 3){
          digitalWrite (PinRele1 , LOW);
          digitalWrite (PinRele2 , LOW);
          break;
        }
      }
    }
  }
  else if (Angulos.AnguloX < -10 and Angulos.AnguloX > -90) {
    digitalWrite (PinRele1 , LOW);
    digitalWrite (PinRele2 , HIGH);
    Serial.println ("DISMINUIR");
    if(BT.available()){
      incoming[0] = BT.read();
      Estado = incoming[0];
      if(Estado == 3){
        digitalWrite (PinRele1 , LOW);
        digitalWrite (PinRele2 , LOW);
        break;
      }
    }
  }
  else {
    digitalWrite (PinRele1 , LOW);
    digitalWrite (PinRele2 , LOW);
    Serial.println ("PUNTO_MUERTO");
  }
}

```

```
        if(BT.available()){
            incoming[0] = BT.read();
            Estado = incoming[0];
            if(Estado == 3){
                digitalWrite(PinRele1, LOW);
                digitalWrite(PinRele2, LOW);
                break;
            }
        }
        Serial.println(i);
    }
    Estado = 3;
}

if (Estado == 3) {
    digitalWrite(PinRele1, LOW);
    digitalWrite(PinRele2, LOW);
    Serial.println("STOP");
    Estado = 0;
}

if (Estado == 4){
    digitalWrite(PinRele1, HIGH);
    digitalWrite(PinRele2, LOW);
    Serial.println("ABRIR_MANO");
    for (int i = 0; i < vecesP; i++) {
        digitalWrite(PinSenal, HIGH);
        delayMicroseconds(AnchoP);
        digitalWrite(PinSenal, LOW);
        delayMicroseconds(Toffus);
        Serial.println(i);
        if(BT.available()){
            incoming[0] = BT.read();
            Estado = incoming[0];
            if(Estado == 3){
                digitalWrite(PinRele1, LOW);
                digitalWrite(PinRele2, LOW);
                break;
            }
        }
    }
    Estado = 3;
}

if (Estado == 5){
    digitalWrite(PinRele1, LOW);
    digitalWrite(PinRele2, HIGH);
    Serial.println("CERRAR_MANO");
}
```

```

for (int i = 0; i < vecesP; i++) {
    digitalWrite(PinSenal, HIGH);
    delayMicroseconds(AnchoP);
    digitalWrite(PinSenal, LOW);
    delayMicroseconds(Toffus);
    Serial.println(i);
    if(BT.available()){
        incoming[0] = BT.read();
        Estado = incoming[0];
        if(Estado == 3){
            digitalWrite(PinRele1, LOW);
            digitalWrite(PinRele2, LOW);
            break;
        }
    }
    Estado = 3;
}

```

- Código ESP-32 Sensorial

```

#include "MPU6050.h"
#include "Wire.h"

#include <esp_now.h>
#include <WiFi.h>

MPU6050 Sensor;

int ConnectedLed = 18;
int LedNotConnected = 19;

int16_t ax, ay, az;

uint8_t AddressESP[] = {0x7C, 0x87, 0xCE, 0x30, 0x94, 0xE8};

typedef struct struct_message {
    float AnguloX;
    float AnguloY;
} struct_message;

struct_message Angulos;

void OnDataSent(const uint8_t *mac_addr, esp_now_send_status_t status) {
    Serial.print("\r\nLast_Packet_Send_Status:\t");
    Serial.println(status == ESP_NOW_SEND_SUCCESS ? "Delivery
    _Success" : "Delivery_Fail");
}

```

```
    if(status == ESP_NOW_SEND_SUCCESS){
        digitalWrite(ConnectedLed, HIGH);
        digitalWrite(LedNotConnected, LOW);
    }
    else{
        digitalWrite(ConnectedLed, LOW);
        digitalWrite(LedNotConnected, HIGH);
    }
}

void setup(){

    Serial.begin(115200);
    Wire.begin();
    Sensor.initialize();

    pinMode(ConnectedLed, OUTPUT);
    pinMode(LedNotConnected, OUTPUT);

    if (Sensor.testConnection()) Serial.println("Sensor_iniciado
    _correctamente");
    else Serial.println("Error_al_iniciar_el_Sensor");

    WiFi.mode(WIFI_STA);

    if (esp_now_init() != ESP_OK) {
        Serial.println("Error_initializing_ESP-NOW");
        return;
    }

    esp_now_register_send_cb(OnDataSent);

    esp_now_peer_info_t peerInfo;
    memset(&peerInfo, 0, sizeof(peerInfo));
    memcpy(peerInfo.peer_addr, AddressESP, 6);
    peerInfo.channel = 0;
    peerInfo.encrypt = false;

    if (esp_now_add_peer(&peerInfo) != ESP_OK){
        Serial.println("Failed_to_add_peer");
        return;
    }
}

void loop(){

    Sensor.getAcceleration(&ax, &ay, &az);
```

```
float AnguloY = atan(ax/sqrt(pow(ay,2) + pow(az,2)))*(180.0/3.14);
float AnguloX = atan(ay/sqrt(pow(ax,2) + pow(az,2)))*(180.0/3.14);

Serial.print("Inclinacion_en_X:");
Serial.print(AnguloX + 31);
Serial.print("\tInclinacion_en_Y:");
Serial.println(AnguloY - 12);

Angulos.AnguloX = AnguloX + 31;
Angulos.AnguloY = AnguloY - 13;

esp_err_t result = esp_now_send(AddressESP, (uint8_t *)
&Angulos, sizeof(Angulos));

if (result == ESP_OK) {
    Serial.println("Sent_with_success");
}
else {
    Serial.println("Error_sending_the_data");
}
delay(500);
}
```